Chapter One

Introduction

1.1 Preface

Multiprotocol Label Switching (MPLS) is a mechanism in high performance telecommunications networks that routing data from one network node to the next based on short path labels rather than long network addresses, avoiding complex lookups in a routing table. The labels identify virtual links (paths) between distant nodes rather than endpoints. MPLS can encapsulate packets of various network protocols. MPLS can be used in different applications such as traffic engineering to provide better and more intelligent link use [21].

1.2 Problem Statement

Using standard IP routing all traffic between two points is sent over the shortest path even though multiple paths may exist. Especially during periods of high traffic volume, this can result in traffic congestion on certain routes while alternative routes are underused, even though traffic protection is not granted during link failure, the standard IP routing protocols don’t give a chance to mark some traffic as important than others.

1.3 Proposed solution

MPLS Traffic engineering (TE) will be used to control the path taken by traffic to improve the network recourses utilization and to avoid situation where parts of the network are congested, while others are underutilized. Label switched path (LSP) priorities and preemption will be used to confiscate resources from less important LSPs, and traffic will be rerouted using fast reroute (FRR) mechanism over the backup link in case of failure, to insure traffic protection as close to the point of failure.

1.4 Objectives

The aim of this research is to apply MPLS-TE application with its features in internet framework to:

1- Best improve the resources utilization which will achieve cost saving and avoid the congestion, that by distribute the users' traffic through different routes.

2- Maintain traffic protection during link failures for packets already passing through the LSP, that by fast reroute and failover concepts.

3- Grantee a certain amount of bandwidth is available for a particular customer's traffic, both in the steady state and under failure conditions, and this by applying customer priority by the pre-emption concept.

1.5 Methodology

In this research we will routing the users' traffic form point to point with two methods the first by using normal routing protocol users' traffic will take the best path that based on OSPF metric and the other by using MPLS-TE distributes the users' traffic through different routes and the users will share the same link when failover take a place and give user traffic priority more than the other when the active link it has not enough bandwidth capacity for all of them.

To apply the normal routing and MPLS-TE (GNS# version 1.2.1 will be used) network emulation with cisco 3725 routers series because it's one of the best types used for MPLS-TE technology.

1.6 Thesis Outlines

The thesis will be organized as follows:

Chapter one: Introduction: a general introduction to what the thesis is all about, problem statement, what will be the cover of results. Gives the answers for what are our goals, what we are going to do to achieve those goals and why?, **Chapter two:** Background: a brief section that gives necessary background information about our research area; especially what have been done before MPLS, MPLS-based VPNs and QoS and to show why MPLS is a better technique for TE **Chapter three**: MPLS-TE: This section provides the detailed experimental work, network design, implementing network design in GNS3 emulator and explains the protocols, used by the MPLS network. **Chapter four:** Result and Discussion of simulation results, **Chapter five:** Conclusion and Recommendations: provides what we have learned, did we meet our goals, what are the suggestions about the research area, what we have untouched in the research area?

Chapter Two

MPLS Basic Principles

MPLS is a new forwarding mechanism in which packets are forwarded based on labels; Labels usually correspond to Layer 3 destination addresses (equal to destination-based routing). Labels can also correspond to other parameters, such as quality of service (QoS), source address, or a Layer 2 circuit. MPLS was designed to support forwarding of other protocols as well. Label switching is performed regardless of the Layer 3 protocol and it is Layer 2.5 protocol explains at figure 2.1. [5]

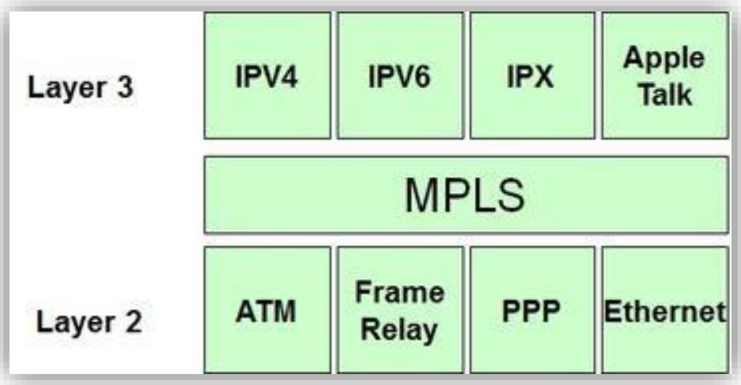


Figure2.1 MPLS at protocols stack [24]

2.1 MPLS Mechanism

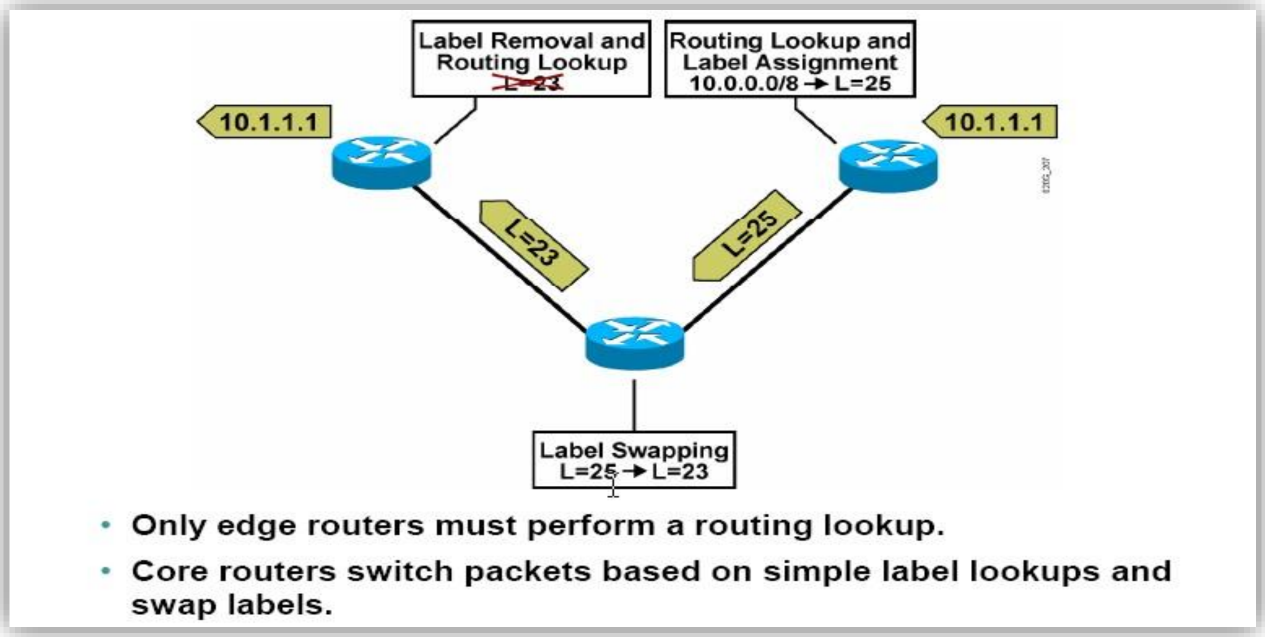
A fundamental property of an MPLS network is that it can be used to tunnel multiple traffic types through the core of the network. Tunneling is a powerful tool because only the routers at the ingress and the egress of the tunnel need to understand the ‘context’ of the underlying traffic carried over the tunnel (e.g. the protocol that the traffic belongs to and the reachability information required to route and forward it in its native form). This detail is hidden from routers in the core of the network. As a consequence, core devices only need to carry sufficient state to enable them to switch MPLS-encapsulated packets without regard to their underlying content. Besides these aggregation properties, which apply to tunnels in general, MPLS tunnels have the following particular properties [1]:

1. Traffic can be explicitly routed, depending on which signaling protocol is used.
2. Recursion is provided for; hence tunnels can exist within tunnels.
3. There is protection against data spoofing, as the only place where data

Can be injected into an MPLS tunnel is at the head end of that tunnel.

In contrast, data can be injected into an IP tunnel from any source that has connectivity to the network that carries the tunnel.

4. The encapsulation overhead is relatively low (4 bytes per MPLS header).



- Only edge routers must perform a routing lookup.
- Core routers switch packets based on simple label lookups and swap labels.

Figure2.2 Basic MPLS Concepts Example [1]

The figure2.2 illustrates a situation in which the intermediary router does not have to perform a time-consuming routing lookup. Instead, this router simply swaps a label with another label (25 is replaced by 23) and forwards the packet based on the received label (23).Usually replaces labels in packets before forwarding Called “**Label swapping**”, Forwarding decision based on Exact Match. Label Switching Router use standard IP protocols to determine where to forward the packet [5].

2.2 MPLS Components

An MPLS network consists of edge devices known as **Label Edge Routers** (**LERs**) or **Provider Edge** (**PE**) routers and core routers known as **Label Switching Routers** (**LSRs**) or **Provider** (**P**) routers.

A mesh of unidirectional tunnels, known as Label Switched Paths (LSPs) is built between the LERs in order that a packet entering the network at the ingress LER can be transported to the appropriate egress LER [1].

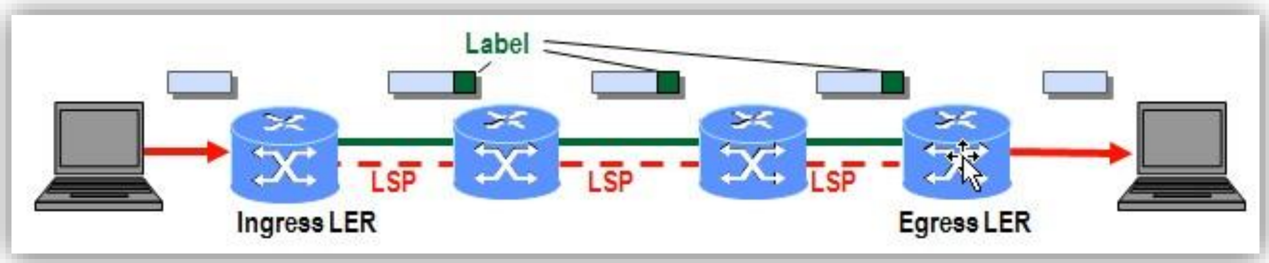


Figure2.3 Label and Label Switched Path (LSP)

When packets enter a network, the ingress router determines which Forwarding Equivalence Class (FEC) the packets belong to. Packets that are to be forwarded to the same egress point in the network along the same path and with the same forwarding treatment along that path are said to belong to the same FEC. Packets belonging to the same FEC are forwarded with the same MPLS label. In a simple case, packets whose destination addresses correspond to the same Border Gateway Protocol (BGP) next-hop are regarded by the ingress router as belonging to the same FEC. In other cases, there may be a more granular assignment of packets to FECs. For example, in DiffServ Aware TE, each egress point in the network may have multiple FECs, each belonging to a different traffic class. It is the role of the ingress LER to determine the appropriate egress LER and LSP to that egress LER associated with the FEC.MPLS has the property that multiple traffic types can be multiplexed on to a single LSP. Therefore, if desired by the network operator, a single LSP can be used to carry all the traffic (e.g. L3VPN, public IP and Layer 2) between a particular ingress LER and a particular egress LER. Transit routers along the path of the LSP make their forwarding decision on the basis of a fixed-format MPLS header, and hence do not need to store 'routes' (L3VPN routes, external IP routes, Layer 2 forwarding information) pertaining to the underlying tunneled packets. This is an important scaling property, as otherwise each of the core routers would have to carry routing information equivalent to the sum of the routing information carried by all the edge routers in the network [1].

MPLS Labels is a short fixed length physically contiguous identifier which is used to identify a FEC, usually of local significance shown at figure2.4.

- **Label Format**

MPLS uses a 32-bit label field that contains the following information [5]:

- 20-bit label
- 3-bit experimental field
- 1-bit bottom-of-stack indicator
- 8-bit TTL field

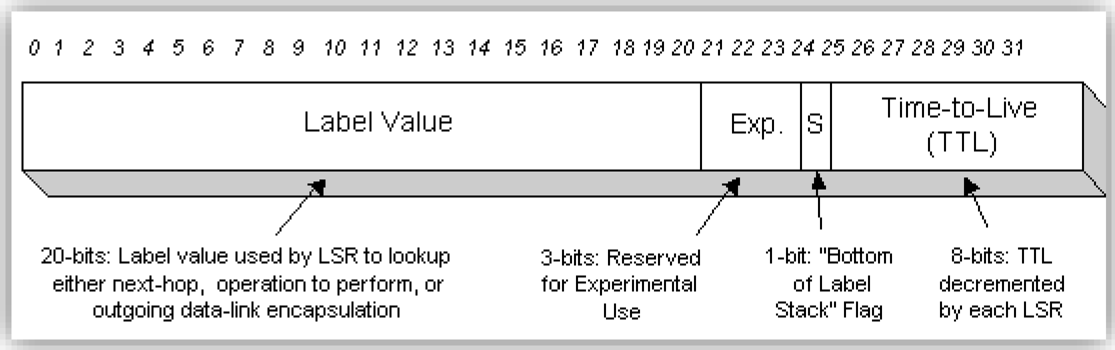

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Label Value
Exp.
S
Time-to-Live (TTL)
20-bits: Label value used by LSR to lookup either next-hop, operation to perform, or outgoing data-link encapsulation
3-bits: Reserved for Experimental Use
1-bit: "Bottom of Label Stack" Flag
8-bits: TTL decremented by each LSR

Figure 2.4 MPLS label format [25]

2.3 MPLS Architecture

MPLS has two major planes:

- **Control plane**: Exchanges Layer 3 routing information and labels; contains complex mechanisms to exchange routing information, such as OSPF, EIGRP, IS-IS, and BGP, and to exchange labels; such as TDP, LDP, BGP, and RSVP. The control plane takes care of the routing information exchange and the label exchange between adjacent devices.
- **Data plane**: The data plane takes care of forwarding based on either destination addresses or labels; this is also known as the forwarding plane. [5]

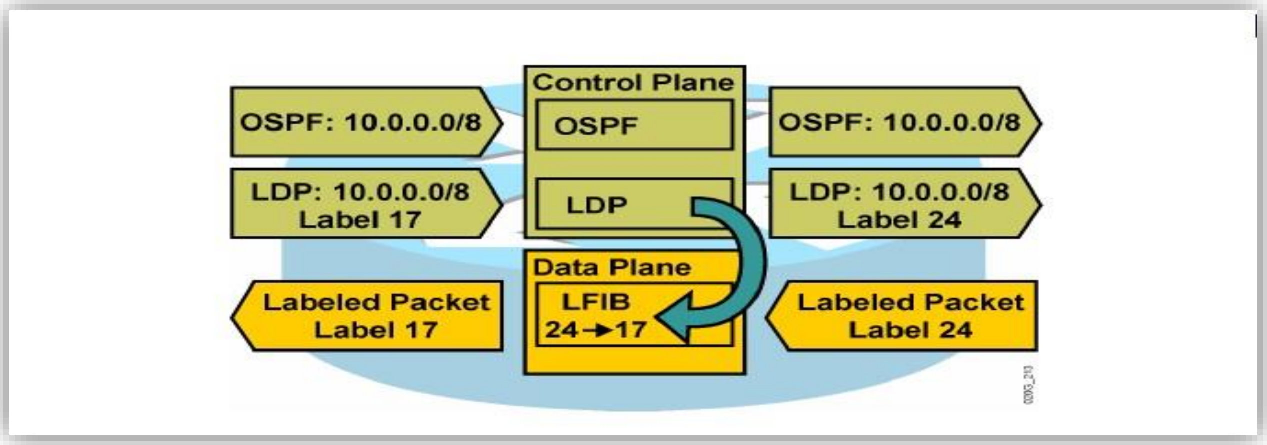


Figure2.5 MPLS Router functionality [5]

At figure 2.5 router functionality is divided into two major parts: the control plane and the data plane.

A large number of different routing protocols, such as Open Shortest Path First (OSPF),Interior Gateway Routing Protocol (IGRP), Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), Routing Information Protocol (RIP), and Border Gateway Protocol (BGP), can be used in the control plane. The control plane also requires protocols such as the label exchange protocols, Tag Distribution Protocol (TDP), MPLS Label Distribution Protocol (LDP), BGP (used by MPLS VPN), to exchange labels. Resource Reservation Protocol (RSVP) is used by MPLS TE to accomplish this exchange .The data plane, however, is a simple label-based forwarding engine that is independent of the type of routing protocol or label exchange protocol. The label forwarding information base (LFIB) table is used to forward packets based on labels. The LFIB table is populated by the label exchange protocols (TDP or LDP, or both) used [5].

2.4 Setting up Traffic-Engineered Paths Using MPLS-TE

Traffic engineering is accomplished in two steps: computing a path that satisfies a set of constraints and forwarding traffic along this path, it is necessary to introduce the concept of LSP priorities.

2.4.1 LSP priorities and preemption

MPLS-TE uses LSP priorities to mark some LSPs as more important than others and to allow them to confiscate resources from less important LSPs (preempt the less important LSPs). Doing this guarantees that:

1. In the absence of important LSPs, resources can be reserved by less important LSPs.

2. An important LSP is always established along the most optimal (shortest) path that fits the constraints, regardless of existing reservations.
3. When LSPs need to reroute (e.g. after a link failure), important LSPs have a better chance of finding an alternate path.

MPLS-TE defines eight priority levels, with 0 as the best and 7 as the worst priority. An LSP has two priorities associated with it: a setup priority and a hold priority. The setup priority controls access to the resources when the LSP is established and the hold priority controls access to the resources for an LSP that is already established. When an LSP is set up, if not enough resources are available, the setup priority of the new LSP is compared to the hold priority of the LSPs using the resources in order to determine whether the new LSP can preempt any of the existing LSPs and take over their resources. If so, the other LSP(s) are torn down. So far so good, but is it ever necessary to assign distinct setup and hold priorities to an LSP? The answer is 'yes', and doing so is the default for many implementations. Assigning an important hold priority (say 0) and a less important setup priority (say 7) to an LSP creates a stable network environment. Using these priorities, a new LSP can never preempt an existing LSP and in turn can never be preempted. Conversely, assigning an unimportant hold priority (say 7) and an important setup priority (say 0) is a recipe for disaster, because it guarantees constant churn if two LSPs compete for the same resource. Imagine that LSP1 has been established over a particular path and that LSP2 wants to use the same links. LSP2's setup priority is better than LSP1's hold priority; thus LSP2 can preempt LSP1. When LSP1 attempts to reestablish, it notices that it can preempt LSP2, and so the cycle of preemption continues indefinitely. For this reason, most implementations disallow the configuration of a hold priority that is worse than the setup priority. Priorities determine the treatment of an LSP in cases of resource contention in the network. They are essential for ensuring that 'important' traffic obtains the necessary resources at a time of shortage (e.g. after a link failure). However, this is not their only application. In a network where large LSPs and small LSPs exist, large LSPs are usually given better priorities to prevent setup failures. The reasoning is that smaller LSPs have a better chance of finding the necessary resources over an alternate path. Having introduced the concept of priorities, we are now ready to start the discussion of path computation [1].

2.4.2 Information distribution – IGP extensions

As seen in the example scenarios, the requirement is to find a path in the network that meets a series of constraints. Therefore, the constraints must be taken into account when calculating feasible paths to a destination. Some of the constraints are [1]:

1. The bandwidth requested for a particular LSP (such as 10 Mbps from Source x to destination y).
2. The administrative attributes of the links that the traffic is allowed to cross. An example of a constraint expressed in terms of link colors is to avoid high-latency links, where these links are marked with a particular administrative attribute.
3. The metric that is assigned to a link for the purpose of traffic engineering.
4. The number of hops that the traffic is allowed to transit.
5. The setup priority of the LSP.

Other constraints are also possible, such as the inclusion or exclusion of a particular hop in the path or the requirement to place two related LSPs on different links, to ensure that failure of a single link does not affect both LSPs. Note that the constraints fall into two categories:

(a) Link properties such as available bandwidth, link color and traffic engineering metric; and

(b) LSP properties such as number of hops or priority.

2.4.3 Path calculation – CSPF

Like conventional SPF, constrained SPF (CSPF) computes a shortest path with regard to some administrative metric. CSPF takes into account only paths that satisfy one or more user-defined constraints (such as available bandwidth) by pruning out of the network topology links that do not satisfy the constraints. For example, if the constraint is bandwidth, CSPF prunes from the topology links that do not have enough bandwidth. In the below scenario:

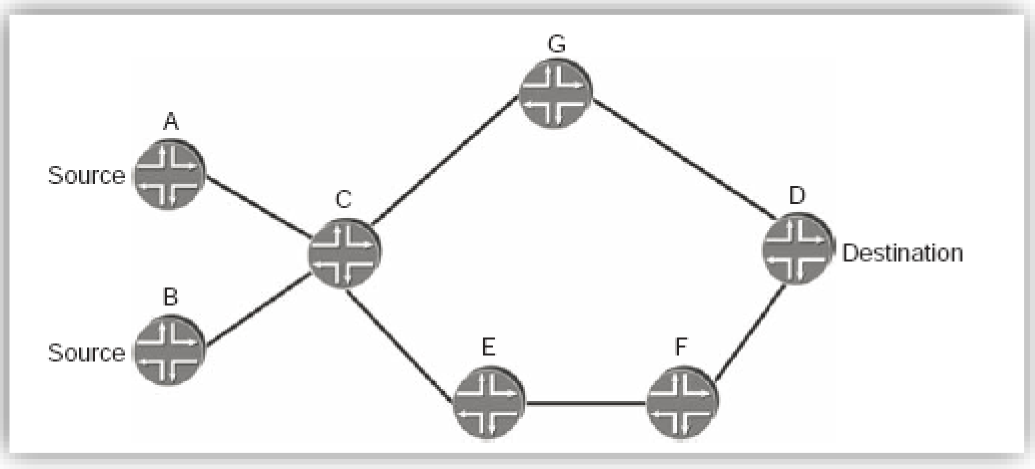


Figure 2.6 a network with two sources, A and B, and two unequal cost Paths to destination D

Once the LSP A–D is set up for 120 Mbps, only 30 Mbps are available along the path A–C–G–D. Thus, when computing the path for LSP B–D, with a requirement of 40 Mbps, the links C–G and G–D are removed from the topology and CSPF picks the alternate path as the best available. Another frequently used constraint is link coloring (also called administrative attributes). The concept of link colors is very intuitive. Links are marked with different colors through configuration and a link can be marked with multiple colors if desired or no colors at all. Up to 32 different colors are available. Figure 2.7 shows an example network where links E–F and F–D are colored ‘red’, link C–D is colored ‘blue’, link C–G is not Colored at all while link C–E is colored both ‘red’ and ‘green’. There is no Restriction on how link colors are assigned, but they typically correspond to link properties such as latency, loss, operational cost or geographic location. The colors are used to express the desire to include or exclude a link or set of links from a particular path. For example, if the operator marks all high latency links with the color ‘blue’, he or she can then compute a path that does not cross high-latency links by excluding links marked with the color ‘blue’ from the path. For example in figure 2.7.

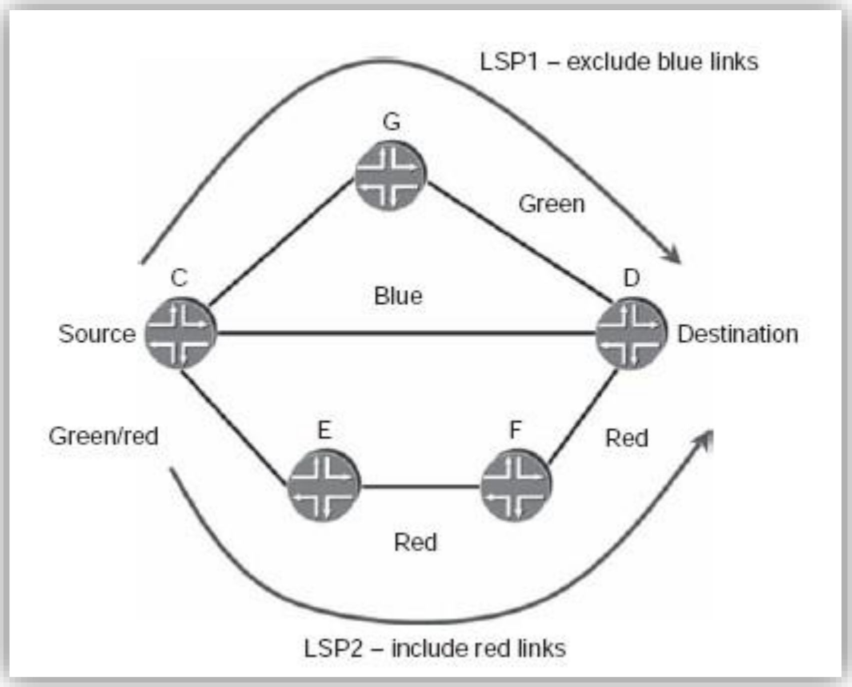


Figure2.7 using color link coloring [1]

Assume link C–D is a high-latency link. LSP1 is set up between C and D with the Constraint ‘exclude blue links’. This means that none of the links in the path can be marked ‘blue’. Thus, although the shortest path is through link C–D, this link is excluded from the computation due to its coloring and the LSP must establish the best path in a topology that does not include link C–D, yielding path

C–G–D. Similarly, LSP2 is set up between C and D with a constraint of ‘include red links’. Thus, all links in the path must be marked red. Note that for this purpose link C–E, which is marked with two colors (red and green), is acceptable. Although the example shown includes and excludes constraints separately, they can be used together for the same LSP. In effect, link coloring creates several TE topologies in the network, where some of the links belong to several topologies. The reason administrative attributes (colors) are sometimes perceived as intimidating is because of how this feature is implemented by some vendors. Administrative attributes are encoded in a bit field in the IGP link advertisement. Froman implementation point of view, the inclusion or exclusion of a link from the computation is accomplished by encoding the user-defined constraints in a similar bit-field format and then performing the necessary bit arithmetic between the link bit field and the constraint bit field. Some implementations force the user to express the constraints in bit format, something non-intuitive for most users. Other implementations offer friendlier interfaces, where instead of bits the user deals with attribute names expressed as strings (words). In both cases, however, the concept is equally simple: tag a link to be able to reference it in the computation.
From the CSPF point of view, the links whose colors do not match the constraints are pruned from the topology [1].

2.4.4 Path setup – RSVP extensions and admission control

After a path has been successfully calculated, it is set up using RSVP-TE; the path is specified at the LSP head end in the Explicit Route Object (ERO). However, the ERO is not the only TE-related information that must be carried in the RSVP messages. RSVP must also carry: the TE information that intermediate nodes must keep track of, such as the bandwidth requested by the LSP, and the information that is relevant in the path setup, such as the setup and hold priorities of the LSP.As the RESV messages travel from the LSP tail end towards the LSP head end, admission control is performed at each node. Admission control during RSVP signaling is required for the following reasons:

1. The LSP may not have necessarily been computed with CSPF.
2. Even if it was computed with CSPF, the state of the available resources

Between the time the computation was performed and the path was signaled may have changed (e.g. because another LSP was set up, sourced at a different node).

3. The result of CSPF is only as accurate as the information in the TED

(Which may not always be up to date because of link advertisement throttling)?

If enough resources are available at a particular node, admission control is successful, the path is set up through the node and the available resources are updated. This information is fed back into the IGP so that other nodes in the network become aware of the new state of the available resources. The information may not be immediately distributed.

It is important to understand that the bandwidth reservations are in the control plane only and that there is no enforcement of the reservations in the data plane. This means that the data plane usage may be higher than the control plane reservation. When it is important to keep the two equal, policing must to be enforced at the ingress of the LSP to ensure that traffic stays within the bounds of the reservation.

If not enough resources are available, it may be necessary to preempt other LSPs passing through the node. This is where the setup and hold priorities of the LSPs come into play. If preemption cannot solve the resource problem, the reservation fails and an error message is sent to the head end. On receipt of the admission control error message, the head end of the LSP recomputed the path. However, if the TED at the head end was not updated in the meantime, it is very likely that the same path is recomputed and the path setup fails again [1].

2.5 Using the Traffic-Engineered Paths

The simplest, most basic way to map traffic to LSPs is through static routing. The LSR can be configured to send traffic to a destination by sending it over the LSP. However, the fact that the route must be manually configured to use the LSP is both restrictive and unscalable from an operational point of view, thus limiting widespread use. To reap the benefits of the traffic-engineered paths, it is necessary for the routing protocols to become aware of the LSPs. From the routing protocol's point of view, an LSP is treated as an interface (a tunnel) and has a metric associated with it. The metric can be the same as that of the underlying IP path or it can be configured to a different value to influence the routing decision. Different routing protocols have different properties and therefore their use of the LSP is different [1].

The rule for LSP usage in BGP is that when an LSP is available to the BGP next-hop of a route, the LSP can be used to forward traffic to that destination. This property is crucial for the implementation of Layer 3 BGP/MPLS VPNs. In a plain IP/MPLS network (non-VPN), this means that if an LSP is set up between the AS a border router (ASBRs), all traffic transiting the AS uses the LSP, with the following consequences:

1. Forwarding for transit traffic is done based on MPLS labels. Thus, none of the routers except the ASBRs need to have knowledge of the destinations outside the AS, and the routers in the core of the network are not required to run BGP. By using an LSP to carry traffic inside the domain it is thus possible to achieve a 'BGP-free core'.
2. The use of an LSP allows tight control over the path that transit traffic takes inside the domain. For example, it is possible to ensure that transit traffic is

forwarded over dedicated links, making it easier to enforce Service-level agreements (SLAs) between providers.

The use of LSPs by the IGPs makes it possible to mix paths determined by constraint-based routing with paths determined by IP routing.

Therefore, even when traffic engineering is applied to only a portion of the network, label-switched paths are taken into account when computing paths across the entire network. This is a very important property from a scalability point of view.

In the context of IGPs, there are two distinct behaviors:

1. Allow the IGP on the LSP head end to use the LSP in the SPF computation.
2. Advertise the LSP in the link-state advertisements so that other routers can also take it into account in their SPF (shortest path first).

There is often a lot of confusion about why two different behaviors are needed and how they differ. This confusion is not helped by the fact that the two behaviors are individually configurable and that vendors use nonintuitive names for the two features. To illustrate the difference between the two, refer to Figure 2.8, which shows a simple network topology, with a single LSP set up between E and D, along the path E–F–D, with a metric of 15. Note that the LSP metric in this case is smaller and therefore better than the IGP metric of the path E–F–D, which is 50.

Traffic is forwarded towards destination W from two sources, E and A. The goal is to forward the traffic along the shortest path. For source E, this means taking the LSP E–D and then the link D–W, yielding a metric of 25 (15 + 10). When the SPF algorithm runs at node E, in order to find this path E has to be able to take the LSP E–D into account in the SPF computation. This is the first behavior described above, called AutoRoute [1].

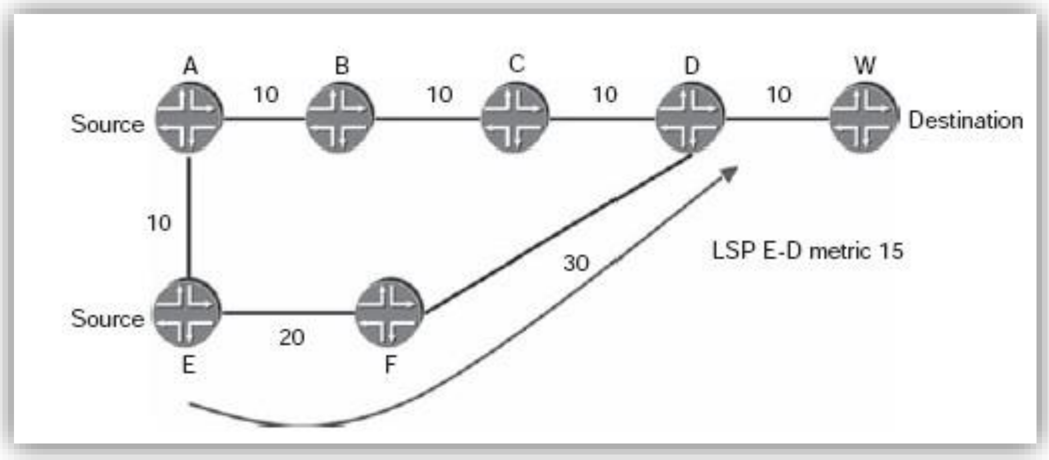


Figure2.8 IGP use LSPs [1]

Concept, however, is very simple: use LSPs originating at a particular node in its SPF computation. When source A sends traffic to destination W, the path with the smallest Metric is through E and the LSP E–D, with a metric of 35 (10 + 15 + 10).However, A is oblivious of the existence of the LSP E–D, because the LSP Originates at node E. For A to be able to take the LSP into account when computing its SPF, it is necessary for node E to advertise the LSP as a link in the link-state advertisements. This is the second behavior described above, called forwarding adjacency or advertise LSP in different vendors' implementations. The concept is simple: distribute the knowledge about the existence of the LSP to other nodes in the network so they can use it in their SPF computation.

Relying on LSP information distributed by other nodes can sometimes cause surprising behavior. This is because the routing decision is made based on a different router's judgment on what the shortest path should be. Let us continue the example above with a slight modification: the metric of the link E–F is 10 instead of 20, as illustrated in Figure 2.9. Because E advertises the LSP in its link-state advertisements, the node F also receives this advertisement. Consequently, F concludes that the shortest path to destination W is through E along the path F–E–

LSP–D–W with a metric of 35 (10 + 15 + 10), rather than through the path F–D–W, with a metric of 40. What happens is that the traffic from F is forwarded to E and then right back to F, only to follow the same links as the pure IGP path. This Happens because F has no insight into the LSP's path and relies on E's advertisement that traffic to W should be forwarded through it. Regardless of whether the protocol used is BGP or one of the IGPs, when several LSPs are available to the same destination, most vendors allow the user the flexibility to pick one out of several LSPs for forwarding, based on various local policies. One such policy can use the class-of-service [1].

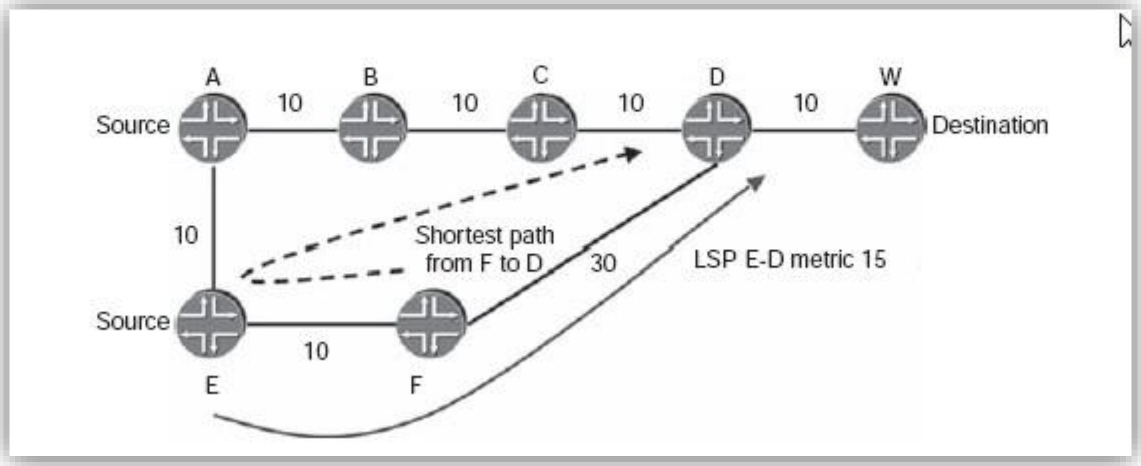


Figure2.9 Surprising behavior when using LSPs in the shortest path computation [1]

Classification of the incoming IP traffic for picking the LSP. For example, Best-effort traffic is mapped to one LSP, while expedited forwarding traffic is mapped to another. By manipulating the properties of these LSPs, the operator can provide more guarantees to the more important traffic.

Mapping traffic to different LSPs in this way is particularly useful in the context of MPLS DiffServ-TE.

To summarize, the ability of the routing protocols to make use of the Traffic engineered paths set up in the network enables control over the path that transit traffic takes in a domain and allows deployment of MPLS-TE in just parts of the network. After seeing how traffic-engineered paths are computed and used, the next things to look at are some of the considerations for deploying a traffic engineering solution [1].

2.6 MPLS VPN Architecture

The MPLS VPN architecture offers service providers a peer-to-peer VPN architecture that combines the best features of overlay VPNs (support for overlapping customer address spaces) with the best features of peer-to-peer VPNs. The following describes these characteristics [5]:

- PE routers participate in customer routing, guaranteeing optimum routing between customer sites.
- PE routers carry a separate set of routes for each customer, resulting in perfect isolation between customers.
- Customers can use overlapping addresses

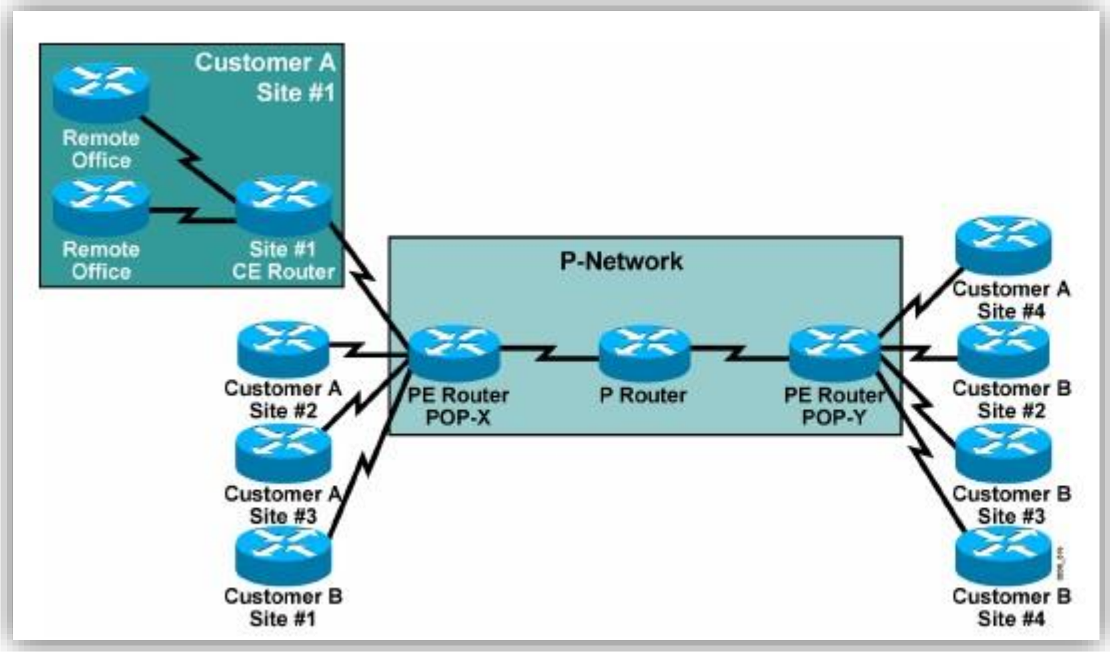


Figure2.10 MPLS VPN Architecture Terminology [5]

Figure 2.10 show MPLS VPN terminology divides the overall network into a customer-controlled part (the C network) and a provider-controlled part (the P-network). Contiguous portions of the C-network are called sites and are linked with the P-network via CE routers. The CE routers are connected to the PE routers, which serve as the edge devices of the P-network. The core devices in the P network, the P routers, provide transit transport across the provider backbone and do not carry customer routes.

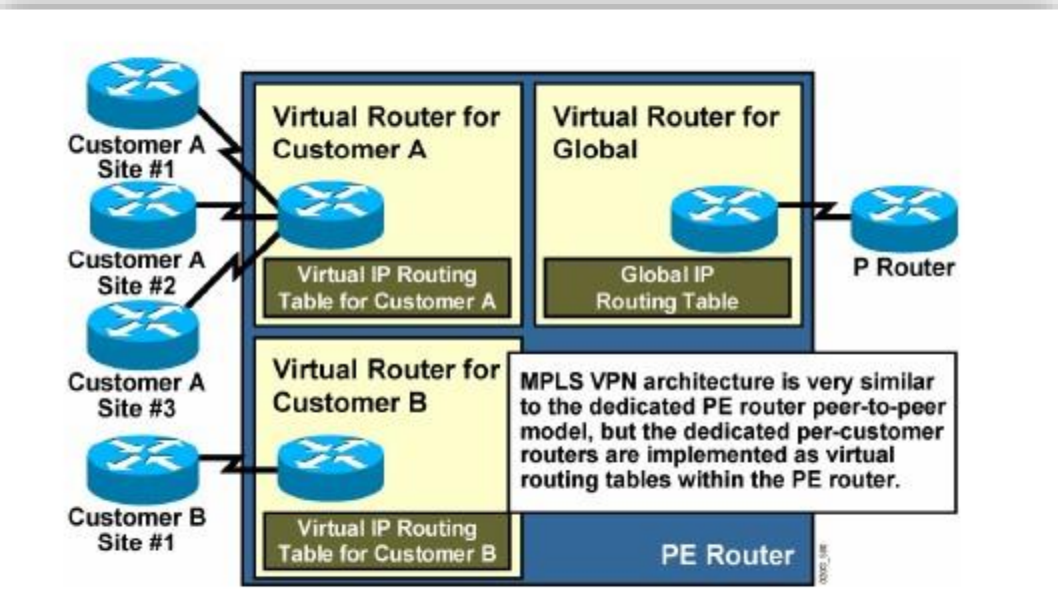

Customer A Site #1
Customer A Site #2
Customer A Site #3
Customer B Site #1
Virtual Router for Customer A
Virtual IP Routing Table for Customer A
Virtual Router for Global
Global IP Routing Table
P Router
Virtual Router for Customer B
Virtual IP Routing Table for Customer B
MPLS VPN architecture is very similar to the dedicated PE router peer-to-peer model, but the dedicated per-customer routers are implemented as virtual routing tables within the PE router.
PE Router

Figure2.11 PE Router Architecture [5]

Figure 2.11 show the architecture of a PE router in an MPLS VPN is very similar to the architecture of a POP in the dedicated PE router peer-to-peer model. The only difference is that the whole architecture is condensed into one physical device. Each customer is assigned an independent routing table (virtual routing table) that corresponds to the dedicated PE router in the traditional peer-to-peer model. Routing across the provider backbone is performed by another routing process that uses a global IP routing table corresponding to the intra-POP P router in the traditional peer-to-peer Model [5].

Although virtual routing tables provide isolation between customers, the data from these routing tables still needs to be exchanged between PE routers to enable data transfer between sites attached to different PE routers. Therefore, a routing protocol is needed that will transport all customer routes across the P-network, while maintaining the independence of individual customer address spaces.

An obvious solution, implemented by various VPN vendors, is to run a separate routing protocol for each customer. There are two common implementations. Both require a per customer routing protocol be run between PE routers. In one implementation, the P routers participate in customer routing and pass the customer routing information between PE routers.

In the other implementation, the PE routers are connected via point-to-point tunnels, for example IPSEC, thereby hiding the customer routing from the P routers [5].

This solution, although very simple to implement (and often used by some customers), is not appropriate in service provider environments because it simply does not scale. The specific problems are as follows and see the figure 2.12:

- The PE routers have to run a large number of routing protocols.
- The P routers have to carry all customer routes.

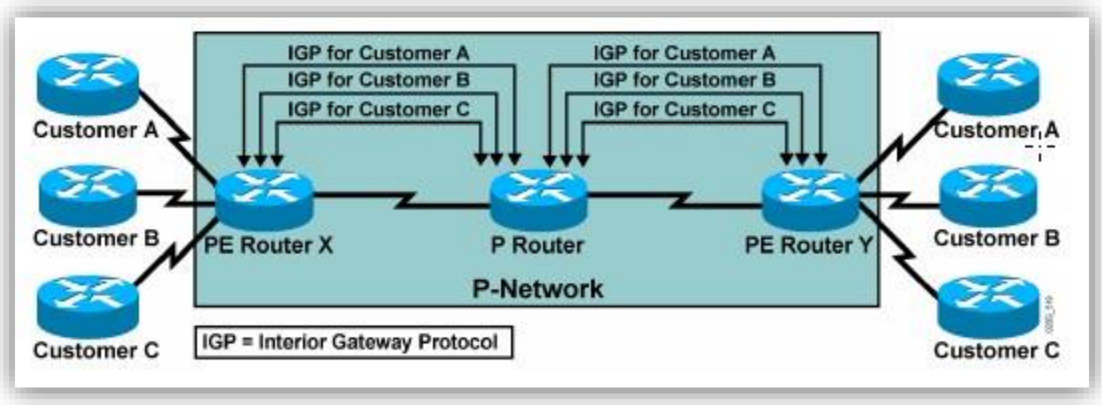


Figure 2.12 Propagation of Routing Information across the P-Network [5]

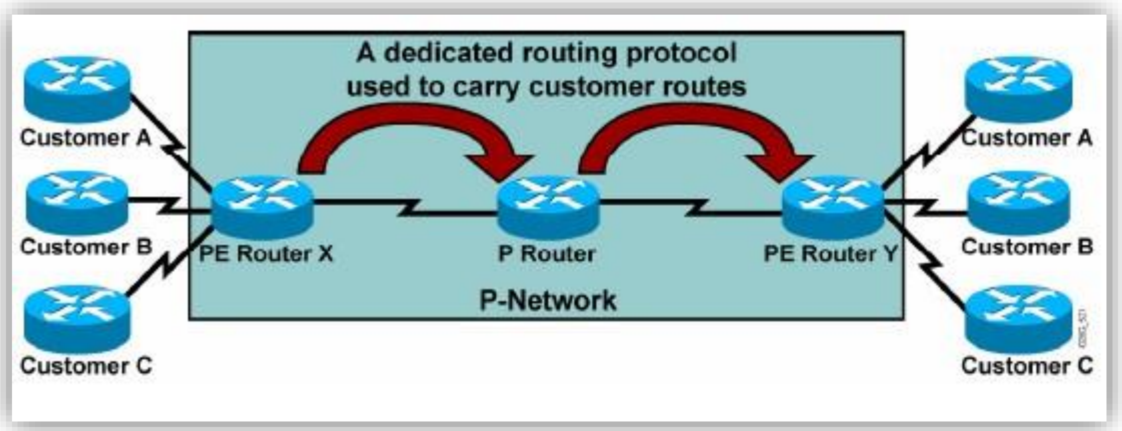


Figure 2.13 a single routing protocol that will carry all customer routes [5]

A better approach to the route propagation problem is to deploy a single routing protocol that can exchange all customer routes across the P-network show figure 2.13. Although this approach is better than the previous one, the P routers are still involved in customer routing; therefore, the proposal retains some of the same scalability issues of the previous one [1].

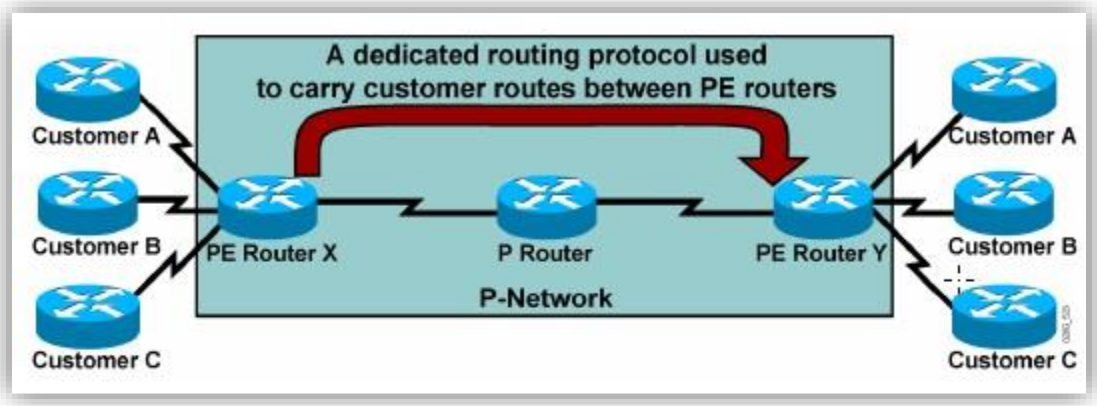


Figure2.14 a single routing protocol between PE routers that without the involvement of the P [5]

The best solution to the customer route propagation issue is to run a single routing protocol between PE routers that will exchange all customer routes without the involvement of the P routers show at figure 2.14. This solution is scalable. Some of the benefits of this approach are as follows:

- The number of routing protocols running between PE routers does not increase with an increasing number of customers.
- The P routers do not carry customer routes.

The next design decision to be made is the choice of the routing protocol running between PE routers. Given that the total number of customer routes is expected to be very large, the only well-known protocol with the required scalability is BGP. In fact, BGP is used in MPLS VPN architecture to transport customer routes directly between PE routers.

MPLS VPN architecture differs in an important way from traditional peer-to-peer VPN solutions—the supports of overlapping customer address spaces.

With the deployment of a single routing protocol (BGP) exchanging all customer routes between PE routers, an important issue arises: how can BGP propagate several identical prefixes, belonging to different customers, between PE routers?

The only solution to this dilemma is the expansion of customer IP prefixes with a unique prefix that makes them unique even if they had previously overlapped. A 64-bit prefix called the RD is used in MPLS VPNs to convert nonunique 32-bit customer addresses into 96-bit unique addresses that can be transported between PE routers [5].

2.7 Route Distinguishers

The RD is used only to transform non unique 32-bit customer IP version 4 (IPv4) addresses into unique 96-bit VPNv4 addresses (also called VPN IPv4 addresses).VPNv4 addresses are exchanged only between PE routers; they are never used between CE routers. BGP between PE routers must therefore support the exchange of traditional IPv4 prefixes and the exchange of VPNv4 prefixes. A BGP session between PE routers is consequently called a Multiprotocol BGP (MP-BGP) session [5].

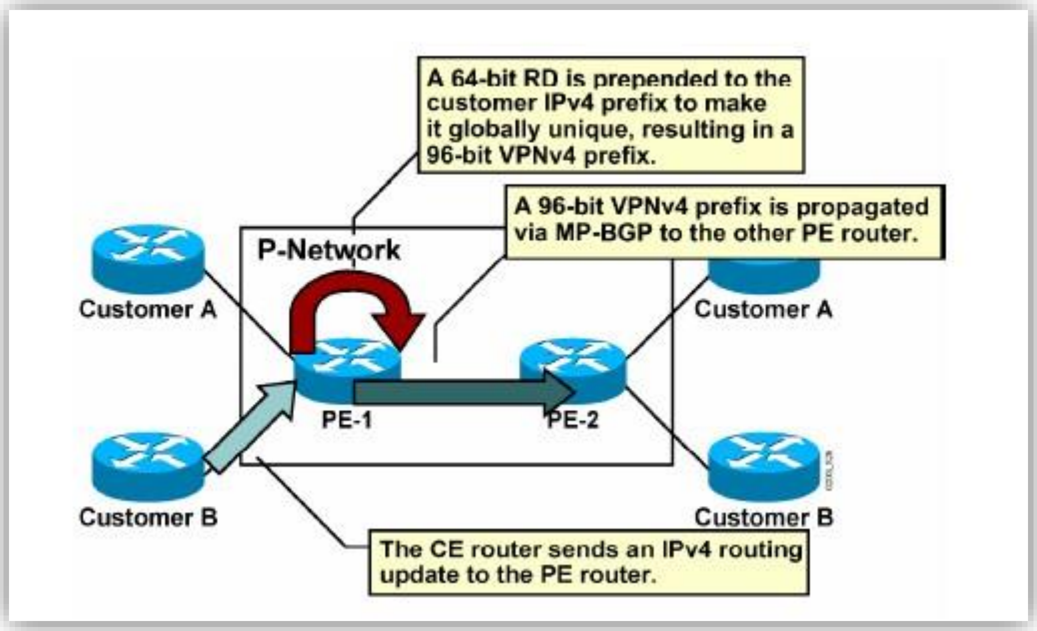

A 64-bit RD is prepended to the customer IPv4 prefix to make it globally unique, resulting in a 96-bit VPNv4 prefix.
A 96-bit VPNv4 prefix is propagated via MP-BGP to the other PE router.
P-Network
Customer A
Customer A
PE-1
PE-2
Customer B
Customer B
The CE router sends an IPv4 routing update to the PE router.

Figure2.15 adding RD to customer IPv4 prefix [5]

Customer route propagation across an MPLS VPN network is done using the following process, see figure 2.15:

Step 1 The CE router sends an IPv4 routing update to the PE router.

Step 2 The PE router prepends a 64-bit RD to the IPv4 routing update, resulting in a globally unique 96-bit VPNv4 prefix.

Step 3 The VPNv4 prefix is propagated via a Multiprotocol Internal Border Gateway Protocol (MP-IBGP) session to other PE routers.

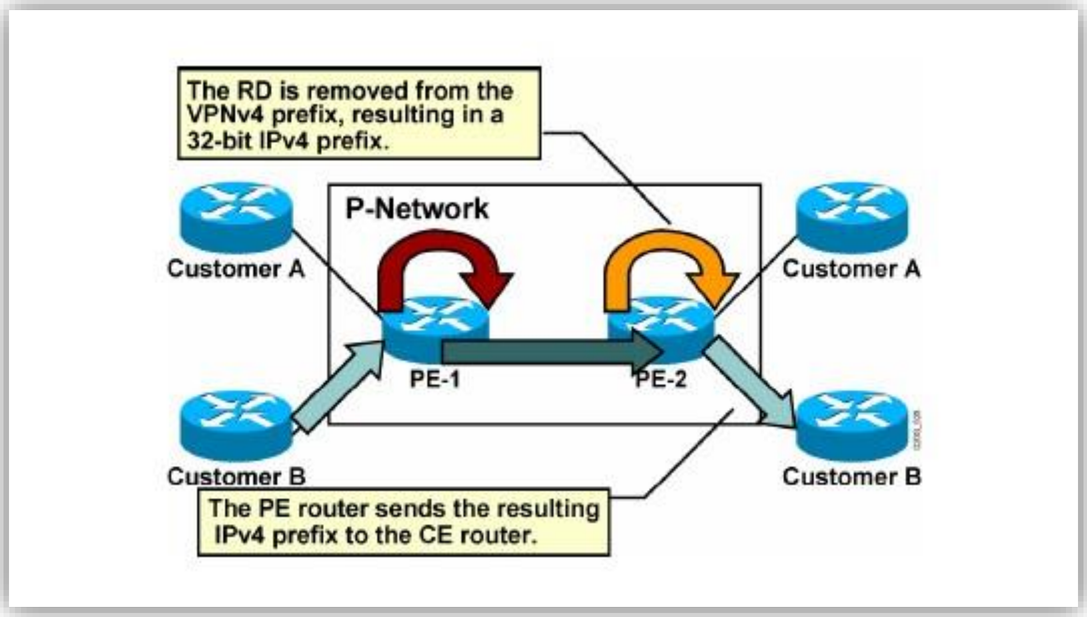

The RD is removed from the VPNv4 prefix, resulting in a 32-bit IPv4 prefix.
P-Network
Customer A
Customer A
PE-1
PE-2
Customer B
Customer B
The PE router sends the resulting IPv4 prefix to the CE router.

Figure 2.15 RD removing from VPNv4 prefix [6]

Step 4 the receiving PE routers strip the RD from the VPNv4 prefix, resulting in an IPv4 prefix.

Step 5 The IPv4 prefix is forwarded to other CE routers within an IPv4 routing update.

The RD has no special meaning or role in MPLS VPN architecture; its only function is to make overlapping IPv4 addresses globally unique.

Configured on the CPE and is not visible to the customer.

Simple VPN topologies require only one RD per customer, raising the possibility that the RD could serve as a VPN identifier. This design, however, would not allow implementation of more complex VPN topologies, such as when a customer site belongs to multiple VPNs.

2.7.1 Route Target

The RD (again, a single entity prepended to an IPv4 route) cannot indicate that a site participates in more than one VPN. A method is needed in which a set of VPN identifiers can be attached to a route to indicate its membership in several VPNs. RTs were introduced into the MPLS VPN architecture to support this requirement. RTs are attributes that are attached to a VPNv4 BGP route to indicate its VPN membership.

The extended BGP communities of routing updates are used to carry the RT of that update, thus identifying to which VPN the update belongs.

As with standard BGP communities, a set of extended communities can be attached to a single BGP route, satisfying the requirements of complex VPN topologies. Extended BGP communities are 64-bit values. The semantics of the extended BGP community are encoded in the high-order 16 bits of the value, making those bits useful for a number of different applications, such as MPLS VPN RTs [6].

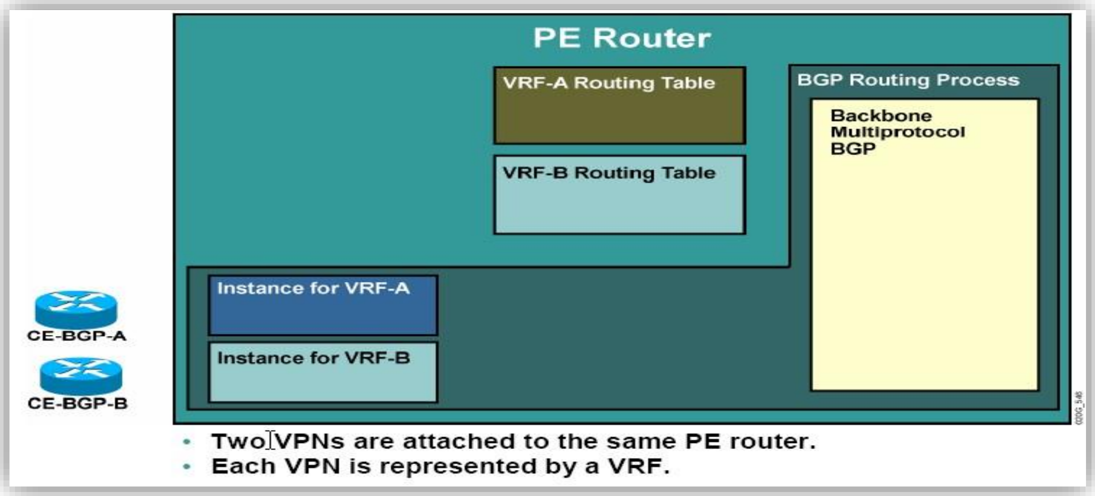
2.7.2 VPN-Aware Routing Protocols

"Routing contexts" were introduced in Cisco IOS software to support the need for separate isolated copies of VPN routing protocols. Routing contexts can be implemented as separate routing processes (OSPF), similar to traditional Cisco IOS software implementation, or as separate isolated "instances" of the same routing protocol.

If the routing contexts are implemented as instances of the same routing protocol, each instance contains its own independent routing protocol parameters. Examples would include networks over which the routing protocol is run, timers, authentication parameters, passive interfaces, and neighbors. This independence allows the network designer maximum flexibility in implementing routing protocols between PE and CE routers.

The routes received from **VRF** routing protocol instances or from dedicated VRF routing processes are inserted into the IP routing table contained within the VRF. This IP routing table supports exactly the same set of mechanisms as the standard Cisco IOS software routing table.

These mechanisms include filter mechanisms (distribute lists or prefix lists) and inter protocol route selection mechanisms (administrative distances).The per-VRF forwarding table (FIB) is built from the per-VRF routing table. This table is used to forward all the packets received through the interfaces associated with the VRF. Any interface can be associated with a VRF, be it a physical interface, sub interface, or a logical interface, as long as it supports CEF switching. There is no limit to the number of interfaces associated with one VRF (other than the number of interfaces supported by the router). However, in practice, each interface can be assigned to only one VRF because the router needs to uniquely identify the forwarding table to be used for packets received over an interface [6].



- Two VPNs are attached to the same PE router.
- Each VPN is represented by a VRF.

Figure2.17 the outbound BGP route propagation process in an MPLS VPN [6]

The figure 2.17 illustrates the interactions between VRF instances of routing processes, VRF routing tables, and the global VPNv4 BGP routing process.

The conclusion a VRF table is a routing and forwarding instance that associates additional attributes such as RD, import RT, and export RT to routing entries.

- Routing contexts allow multiple copies of routing protocols to run concurrently as separate VRF instances to prevent undesired route leakage between VPNs.
- VPN-aware routing protocols allow separation of routing tables either as separate routing processes (OSPF) or separate isolated instances of the same protocol (BGP, EIGRP, RIPv2).
- A VRF table is used to logically separate routing information from different VPNs.

Chapter Three

Network Design and Modeling

GNS3 is a graphical network simulator that allows designing complex network topologies. You may run simulations or configure devices ranging from simple workstations to powerful Cisco routers. It is based on Dynamips, Pemu/Qemu and Dynagen.

Using GNS3 MPLS-TE Tunnels per VRF technique shown in figure 3.1 will be modeled and the flow chart of routing process show at figure3.2 for normal routing protocol (OSPF) the all users' traffic will take the shortest path to reach their destinations and figure 3.3 for MPLS-TE routing process the user' traffic will distribute through specific routes assign by MPSL-TE.

This chapter covers MPLS-TE Tunnels per VRF this is a solution to route traffic from different VRFs to different MPLS-TE tunnels with different routes rather than making all VRFs traffic to go through a single MPLS-TE tunnel and route.

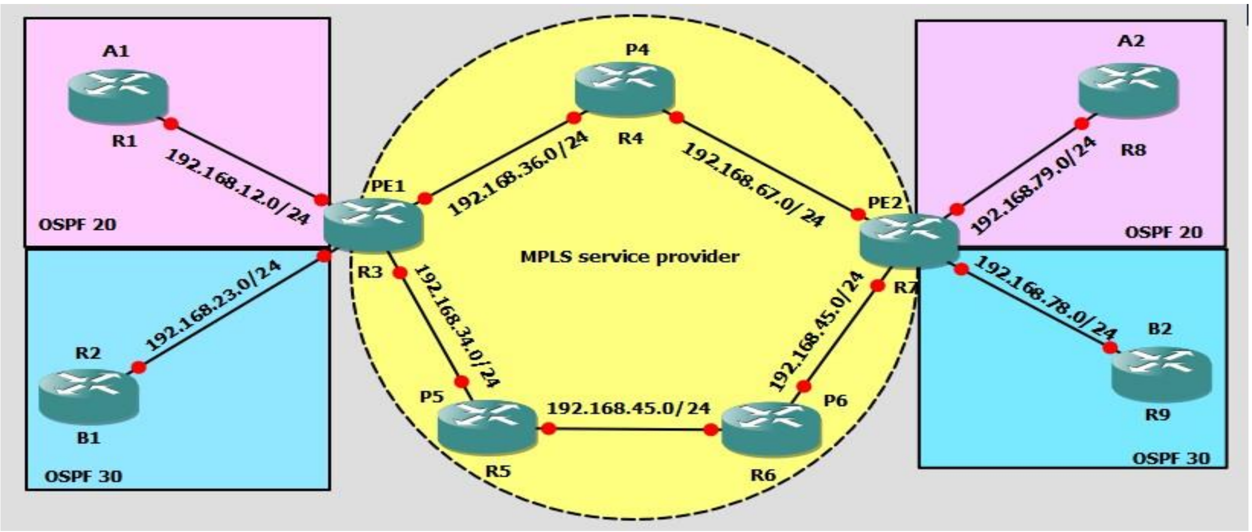


Figure 3.1 MPLS-TE Simulation model

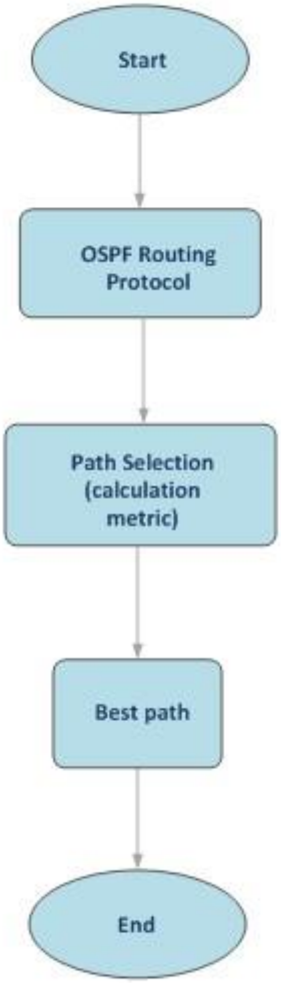


Figure 3.2 normal routing processes (OSPF)

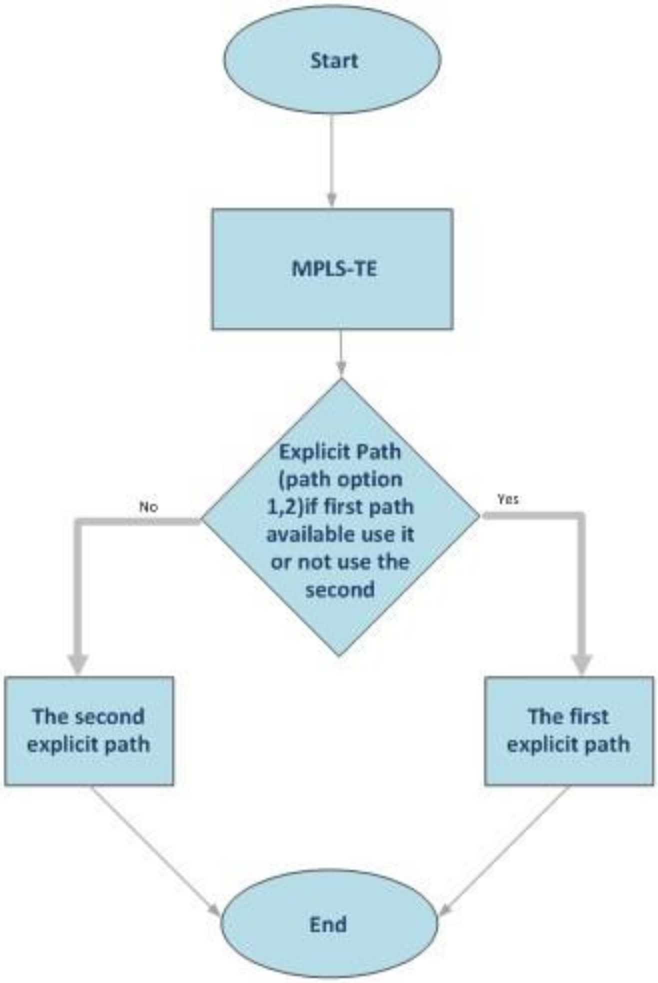

Start
MPLS-TE
Explicit Path (path option 1,2)if first path available use it or not use the second
No
Yes
The second explicit path
The first explicit path
End

Figure 3.3 the MPLS-TE process

3.1 Simulation Assumption and Parameters

For a large MPLS provider two of customers called “A” and “B” would like to use MPLS services to connect their head quarter and branch offices. MPLS backbone has multiple routers and one of the problems there is no load-sharing within the MPLS cloud. Need to make sure offer customers L3 services and that MPLS backbone has traffic engineering so can share the load on all routers based on tagging & tunneling.

- ❖ There are two companies A and B.

In (the) simulation (there will be) two companies (A and B); each company has two users one at Head quarter the other at the branch.

Here (four scenario will be applied) to communicate these users (A1 with A2 and B1 with B2) between each other's, taking on our consideration that the connectivity termination of the two companies is on the same routing equipment.

1. Without using MPLS-TE the routers routed the traffic based on the normal routing protocol (OSPF) by selecting the best path.

   The figure 3.4 simulate the normal routing between two end points; routing to specific destination will always be through one path for all companies.

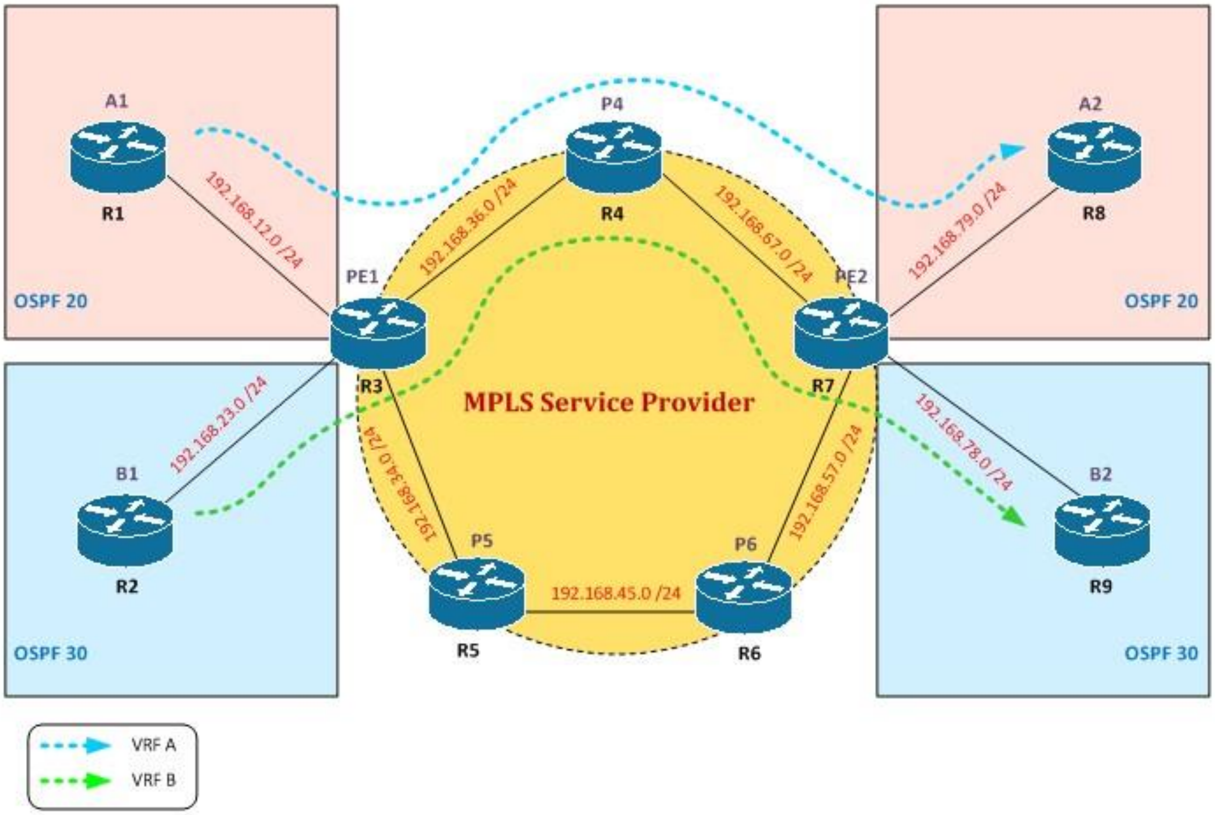


Figure3.4 Routing based on OSPF (normal routing)

2. All traffic from A1 router to A2 router (these routers are part of VRF CUST1) should go through MPLS-TE Tunnel11 while traffic from B1 router to B2 router (part of VRF CUST2) must go through Tunnel10.

   The figure 3.5 each company will take different path to reach the end point using the TE mechanism.

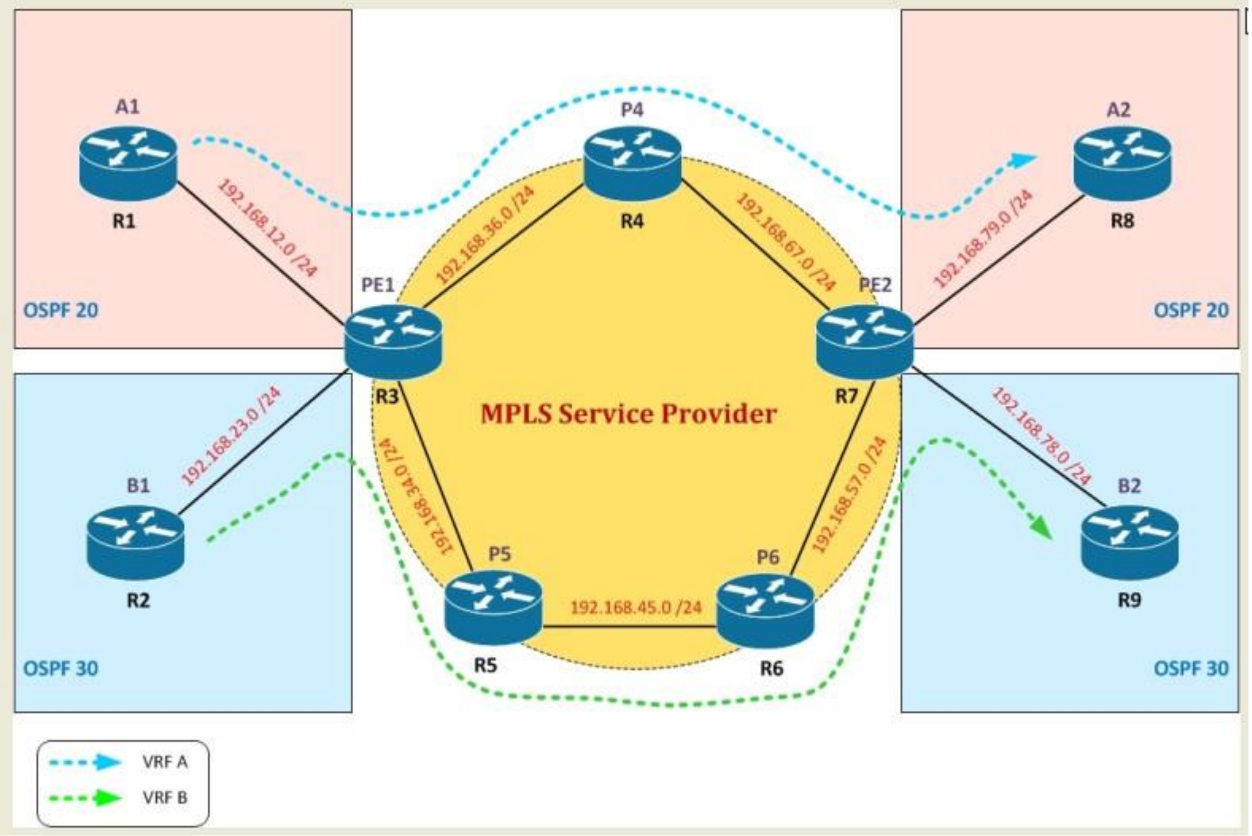


Figure3.5 Routing based on MPLS-TE per VRF

3. When the tunnel down the traffic will go through the other backup route.

   The figure 3.6 Show if there is a link failure the companies will share the active link because the other link will be out of service by concept called Fast reroute.

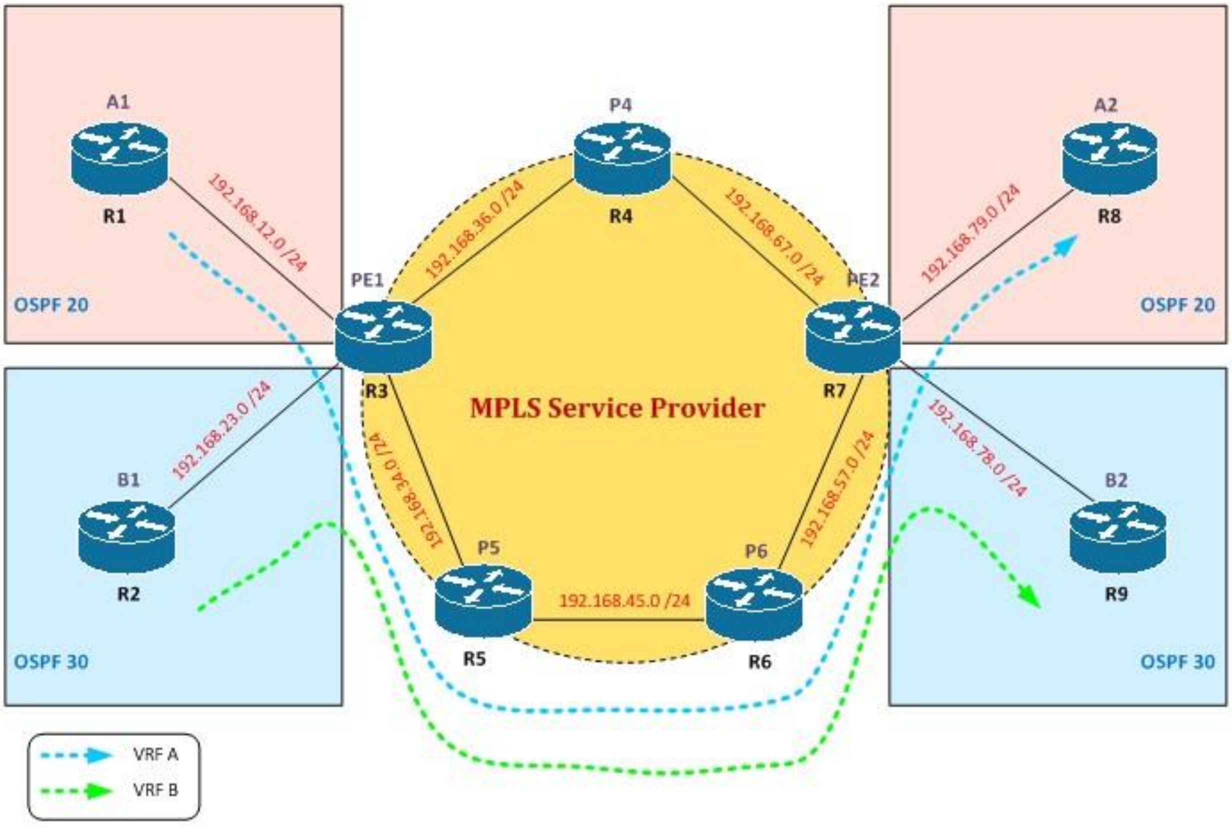


Figure3.6 Fast reroute when failure link take place

4. And when using preemption the customer with high priority will take over all routes if needed.

   The figure 3.7 Explain if there is a link failure the VIP Company “A” will use the overall link bandwidth and eliminate the other company (B1), by preemption concept.

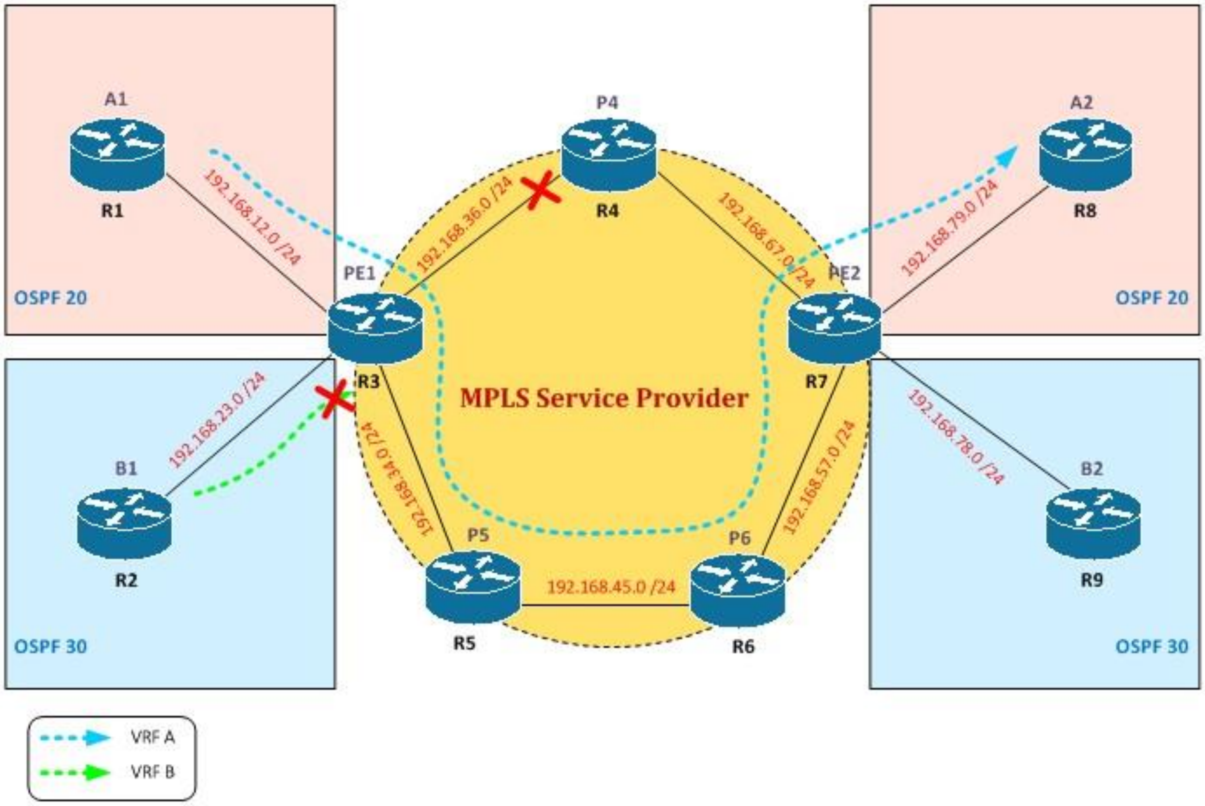


Figure3.7 Preemption based on priority

5. The main parameter is the bandwidth reserve.

3.2 Routers Configuration steps:

- The TCP/IP configure at all interfaces.
- Every router has a loopback0 interfaced configured.
- Configure OSPF Area 0 at the provider side (Router PE1, PE2, P4, P5 and P6).
- Advertise the loopback interfaces as well in OSPF.
- Advertise the loopback0 interfaces as /32.
- Configure MPLS on all physical interfaces in the service provider domain.
- Configure VRF “**A**” on PE1 and PE2 as following:

  - RD 1:100
  - Route-target both 1:100
- Configure VRF “**B**” on PE1 and PE2 as following:
  - RD 1:200
  - Route-target both 1:200
- On router PE1 and PE2 add the interfaces pointing towards the customers to the VRFs.
- Configure OSPF Area 20 on router “**A**” HQ and “**A**” Branch. Advertise the loopbacks as well.
- Configure OSPF Area 30 on router “**B**” HQ and “**B**” Branch. Advertise the loopbacks as well.
- Configure OSPF on router PE1 and PE2 for the correct VRFs.
- Configure BGP AS 1 between Router PE1 and PE2.
- Configure the correct BGP address families and make sure communities are sent between neighbors.
- Redistribute OSPF into BGP; use the correct address-family for the VRFs.
- At this moment should have a working MPLS network but all traffic is being sent through P4. We are going to use MPLS traffic engineering to use P5 and P6 as well.
- Configure the loopback0 interfaces on router PE1 and PE2 as the BGP neighbor next-hop for VRF “**B** and **A**”.
- Configure a tunnel10 interface on router PE1 and PE2 for VRF “**B**”. Make sure the tunnel is in MPLS traffic engineer mode.
- Configure the hold and setup priority to 1 for the tunnel 10 interface, set the bandwidth to 2000.

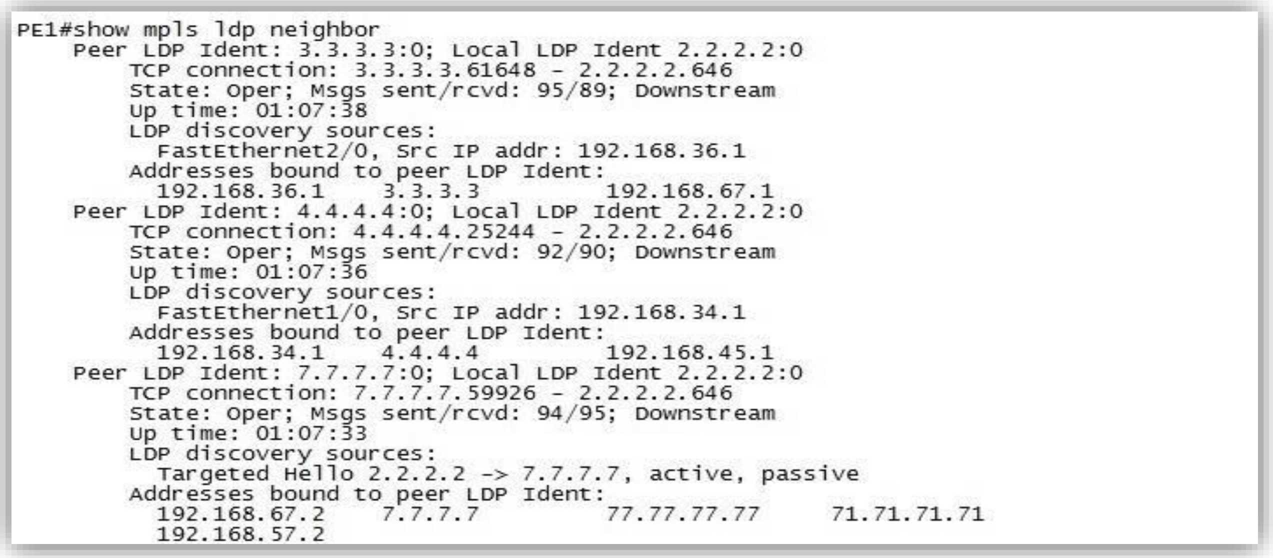
- Configure a tunnel11 interface on router PE1 and PE2 for VRF "**A"**. Make sure the tunnel is in MPLS traffic engineer mode.
- Configure the hold and setup priority to 1 for the tunnel 11 interface, set the bandwidth to 2000.
- Configure the RSVP bandwidth to 4000 for all links interconnecting the P and PE routers.
- Configure MPLS traffic engineering tunnel support for all links interconnecting the P and PE routers.
- Configure the loopback1 interfaces on router PE1 and PE2 as the BGP next-hop for VRF "**A"** (binding VRF to tunnel).
- Configure static routes at PE1 to reach loopback1 at PE2 and vice Versa.
- Configure the loopback2 interfaces on router PE1 and PE2 as the BGP next-hop for VRF "**B"** (binding VRF to tunnel).
- Configure static routes at PE1 to reach loopback2 at PE2 and vice Versa.
- Now traffic for customer "**B"** is sent from PE1 through P5 and P6.
- Now traffic for customer "**A"** is sent from PE1 through P4.
- R1- R9 have been configured see Appendix 1 – 9.

3.2.1Verification:

3.2.1.1 LDP adjacency over MPLS TE:

Since LDP was enabled on MPLS TE tunnels, LDP forms adjacency over TE tunnels. This is important for label exchange for Loopback prefixes.

Figure 3.8 show the LDP neighbor, the LDP was implemented at the global mode and under interfaces also

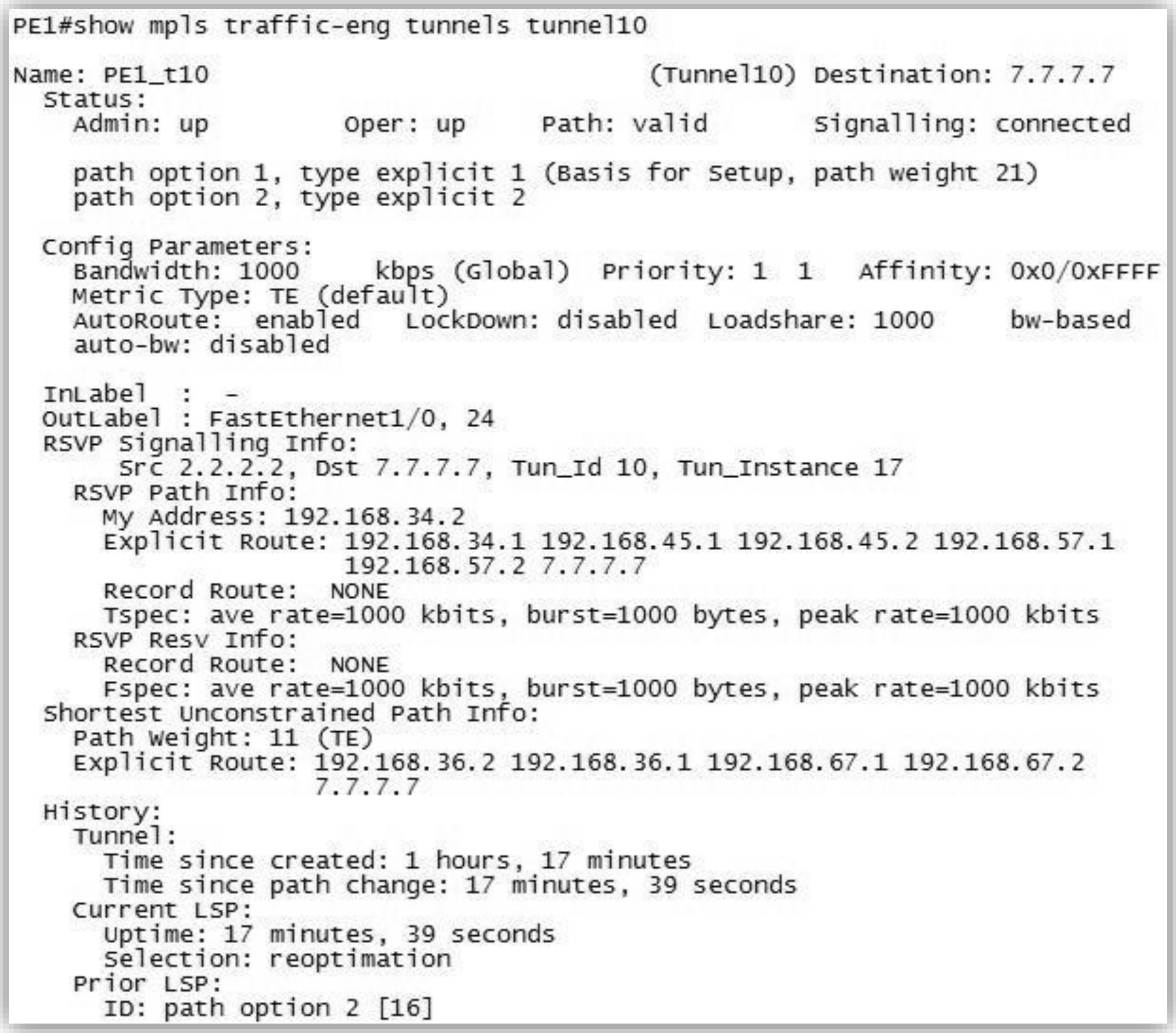


```
PE1#show mpls ldp neighbor
    Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 3.3.3.3.61648 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 95/89; Downstream
        Up time: 01:07:38
        LDP discovery sources:
          FastEthernet2/0, Src IP addr: 192.168.36.1
        Addresses bound to peer LDP Ident:
          192.168.36.1    3.3.3.3         192.168.67.1
    Peer LDP Ident: 4.4.4.4:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 4.4.4.4.25244 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 92/90; Downstream
        Up time: 01:07:36
        LDP discovery sources:
          FastEthernet1/0, Src IP addr: 192.168.34.1
        Addresses bound to peer LDP Ident:
          192.168.34.1    4.4.4.4         192.168.45.1
    Peer LDP Ident: 7.7.7.7:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 7.7.7.7.59926 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 94/95; Downstream
        Up time: 01:07:33
        LDP discovery sources:
          Targeted Hello 2.2.2.2 -> 7.7.7.7, active, passive
        Addresses bound to peer LDP Ident:
          192.168.67.2    7.7.7.7         77.77.77.77     71.71.71.71
          192.168.57.2
```

Figure 3.8 MPLS LDP neighbor

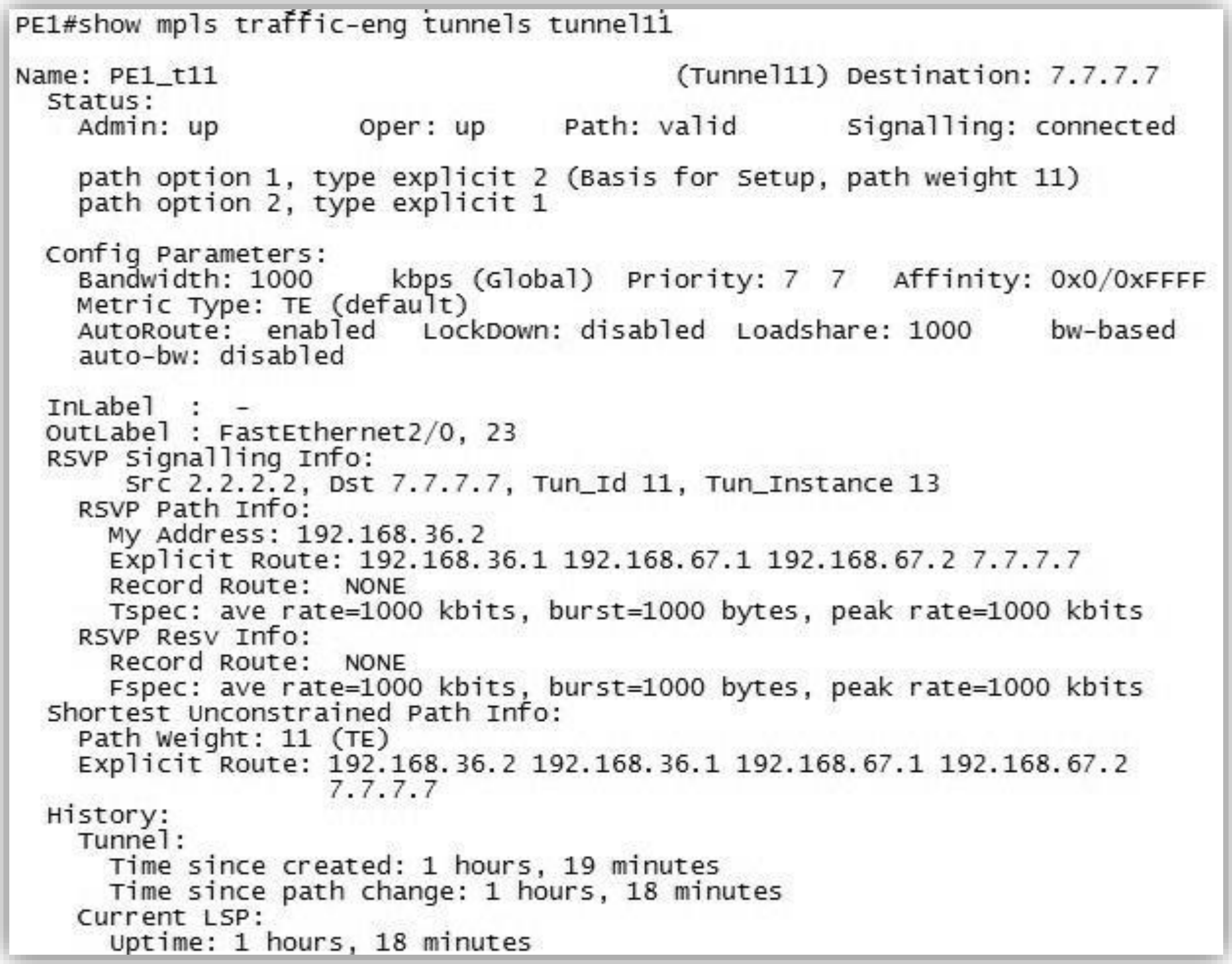
3.2.1.2 Label exchange using RSVP-TE:

RSVP is used to exchange labels for MPLS TE tunnels. In this case, PE1 and PE2 are connected and Tunnel10 and tunnel11 is explicitly configured, the explicit routes shown at the figure 3.9 (tunnel10) and figure 3.10 (tunnel11) also the other information like the tunnel status, the tunnel name also the head and the tail to the tunnel.



```
PE1#show mpls traffic-eng tunnels tunnel10

Name: PE1_t10                             (Tunnel10) Destination: 7.7.7.7
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 1, type explicit 1 (Basis for Setup, path weight 21)
    path option 2, type explicit 2

  Config Parameters:
    Bandwidth: 1000     kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 1000     bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : FastEthernet1/0, 24
  RSVP Signalling Info:
       Src 2.2.2.2, Dst 7.7.7.7, Tun_Id 10, Tun_Instance 17
    RSVP Path Info:
      My Address: 192.168.34.2
      Explicit Route: 192.168.34.1 192.168.45.1 192.168.45.2 192.168.57.1
                      192.168.57.2 7.7.7.7
      Record Route:  NONE
      Tspec: ave rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
    RSVP Resv Info:
      Record Route:  NONE
      Fspec: ave rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 11 (TE)
    Explicit Route: 192.168.36.2 192.168.36.1 192.168.67.1 192.168.67.2
                    7.7.7.7
  History:
    Tunnel:
      Time since created: 1 hours, 17 minutes
      Time since path change: 17 minutes, 39 seconds
    Current LSP:
      Uptime: 17 minutes, 39 seconds
      Selection: reoptimation
    Prior LSP:
      ID: path option 2 [16]
```

Figure 3.9 the status of MPLS traffic tunnel10



```
PE1#show mpls traffic-eng tunnels tunnel11

Name: PE1_t11                               (Tunnel11) Destination: 7.7.7.7
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 1, type explicit 2 (Basis for Setup, path weight 11)
    path option 2, type explicit 1

  Config Parameters:
    Bandwidth: 1000     kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 1000     bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : FastEthernet2/0, 23
  RSVP Signalling Info:
       Src 2.2.2.2, Dst 7.7.7.7, Tun_Id 11, Tun_Instance 13
    RSVP Path Info:
      My Address: 192.168.36.2
      Explicit Route: 192.168.36.1 192.168.67.1 192.168.67.2 7.7.7.7
      Record Route:  NONE
      Tspec: ave rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
    RSVP Resv Info:
      Record Route:  NONE
      Fspec: ave rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 11 (TE)
    Explicit Route: 192.168.36.2 192.168.36.1 192.168.67.1 192.168.67.2
                    7.7.7.7
  History:
    Tunnel:
      Time since created: 1 hours, 19 minutes
      Time since path change: 1 hours, 18 minutes
    Current LSP:
      Uptime: 1 hours, 18 minutes
```

Figure 3.10 the status of MPLS traffic tunnel11

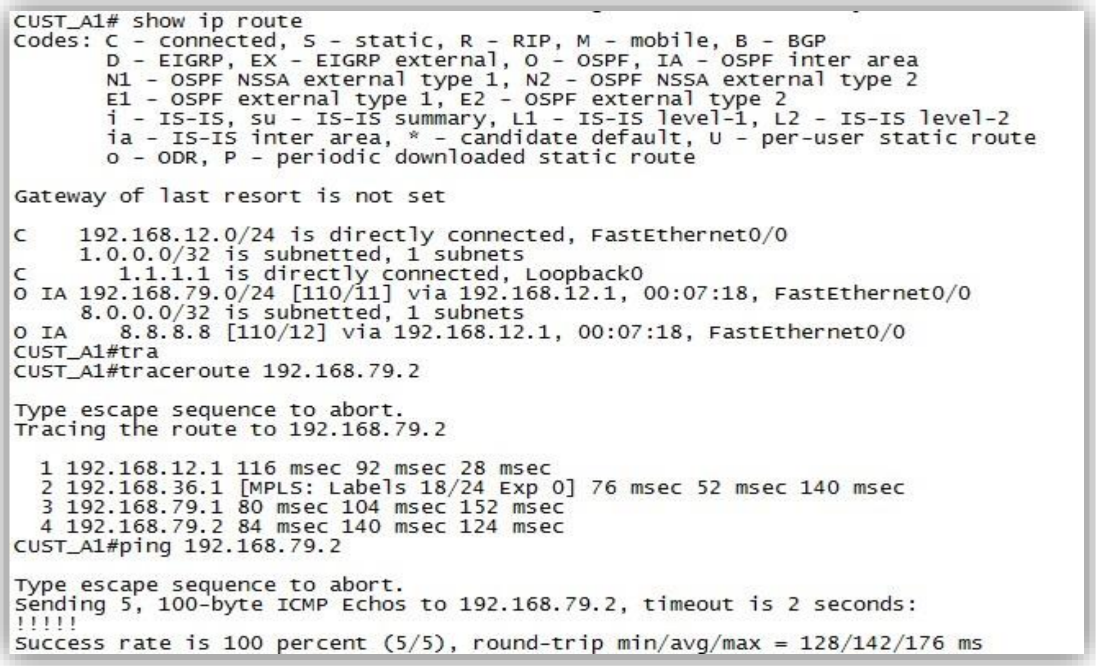
Chapter Four

Results and Discussion

In this chapter the normal routing based on OSPF protocol and MPLS-TE will be analyzed to deliver customer traffic in different scenarios with different techniques such as fast reroute, load share and preemption, MPLS-TE account for link bandwidth and for the size of the traffic flow when determining routes for LSPs across the backbone, Has a dynamic adaptation mechanism that enables the backbone to be resilient to failures, even if several primary paths are pre calculated off-line Enhancements to the IGP (OSPF) calculations to automatically calculate what traffic should be sent over what LSPs.

4.1 Routing based on OSPF protocol

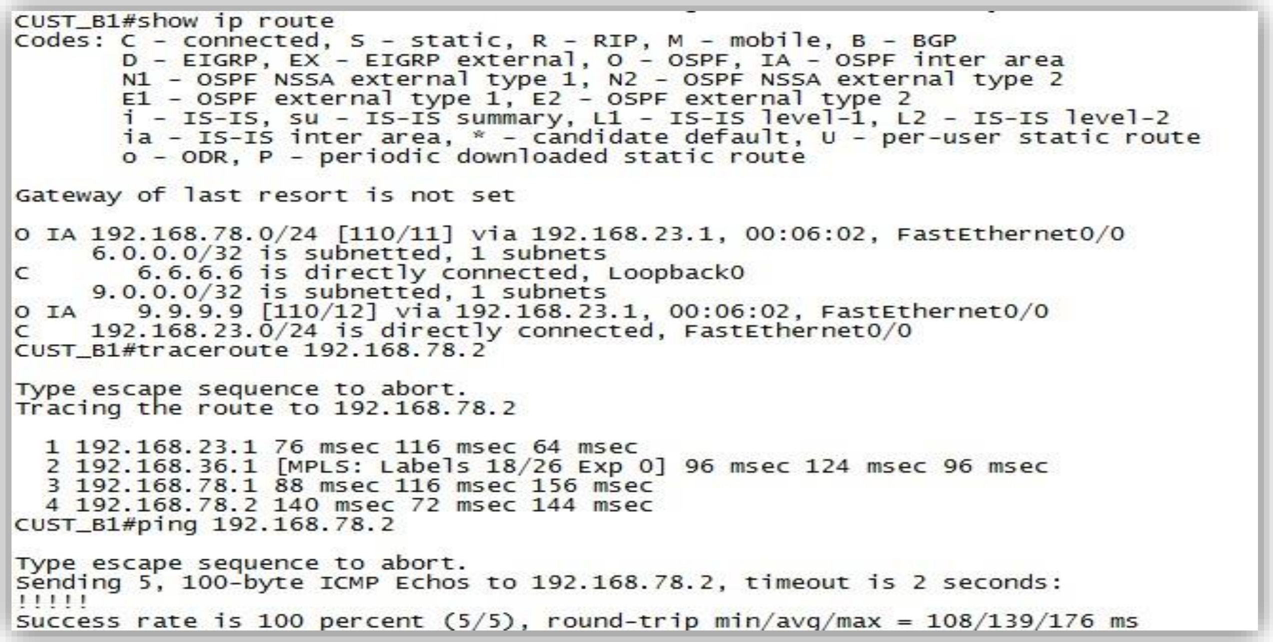
The OSPF protocol select the best path to forward CUST_A1 traffic figure 4.1 show the best path to destination and all the customers at this router used the same path. OSPF uses a link state routing algorithm, It computes the shortest path tree for each route using a method based on Dijkstra's algorithm, a shortest path first algorithm.



```
CUST_A1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
O IA 192.168.79.0/24 [110/11] via 192.168.12.1, 00:07:18, FastEthernet0/0
     8.0.0.0/32 is subnetted, 1 subnets
O IA    8.8.8.8 [110/12] via 192.168.12.1, 00:07:18, FastEthernet0/0
CUST_A1#tra
CUST_A1#traceroute 192.168.79.2

Type escape sequence to abort.
Tracing the route to 192.168.79.2

  1 192.168.12.1 116 msec 92 msec 28 msec
  2 192.168.36.1 [MPLS: Labels 18/24 Exp 0] 76 msec 52 msec 140 msec
  3 192.168.79.1 80 msec 104 msec 152 msec
  4 192.168.79.2 84 msec 140 msec 124 msec
CUST_A1#ping 192.168.79.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.79.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 128/142/176 ms
```

Figure 4.1 Normal routing (OSPF) for CUST_A1

The OSPF protocol select the best path to forward CUST_B1 traffic figure 4.2 show the best path to destination and all the customers at this router used the same path.

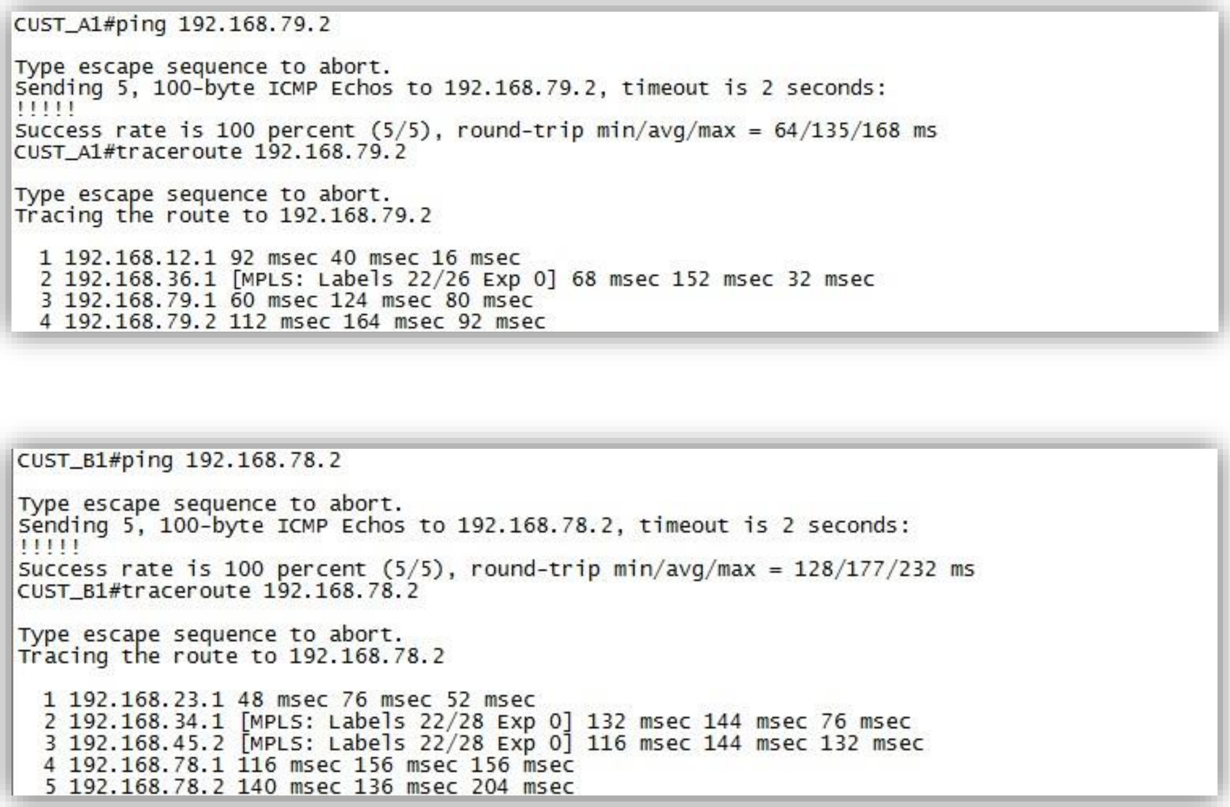


```
CUST_B1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O IA 192.168.78.0/24 [110/11] via 192.168.23.1, 00:06:02, FastEthernet0/0
     6.0.0.0/32 is subnetted, 1 subnets
C       6.6.6.6 is directly connected, Loopback0
     9.0.0.0/32 is subnetted, 1 subnets
O IA    9.9.9.9 [110/12] via 192.168.23.1, 00:06:02, FastEthernet0/0
C    192.168.23.0/24 is directly connected, FastEthernet0/0
CUST_B1#traceroute 192.168.78.2

Type escape sequence to abort.
Tracing the route to 192.168.78.2

  1 192.168.23.1 76 msec 116 msec 64 msec
  2 192.168.36.1 [MPLS: Labels 18/26 Exp 0] 96 msec 124 msec 96 msec
  3 192.168.78.1 88 msec 116 msec 156 msec
  4 192.168.78.2 140 msec 72 msec 144 msec
CUST_B1#ping 192.168.78.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.78.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 108/139/176 ms
```

Figure 4.2 Normal routing (OSPF) for CUST_B1

4.2 Routing based on MPLS-TE per VRF

At figure 4.3 the traffic from different VRFs were routed to different MPLS-TE tunnels rather than making all VRFs traffic to go through a single MPLS-TE tunnel and this prevent congestion and for cost issue. All traffic from A1 router to A2 router these routers are part of VRF CUST_A1 go through MPLS-TE Tunnel11 while traffic from router B1 to B1 router (part of VRF CUST_B1) go through Tunnel10.with VRF many routing table were virtually created on PE1 and PE2 and these routing table completely isolated from the global routing table

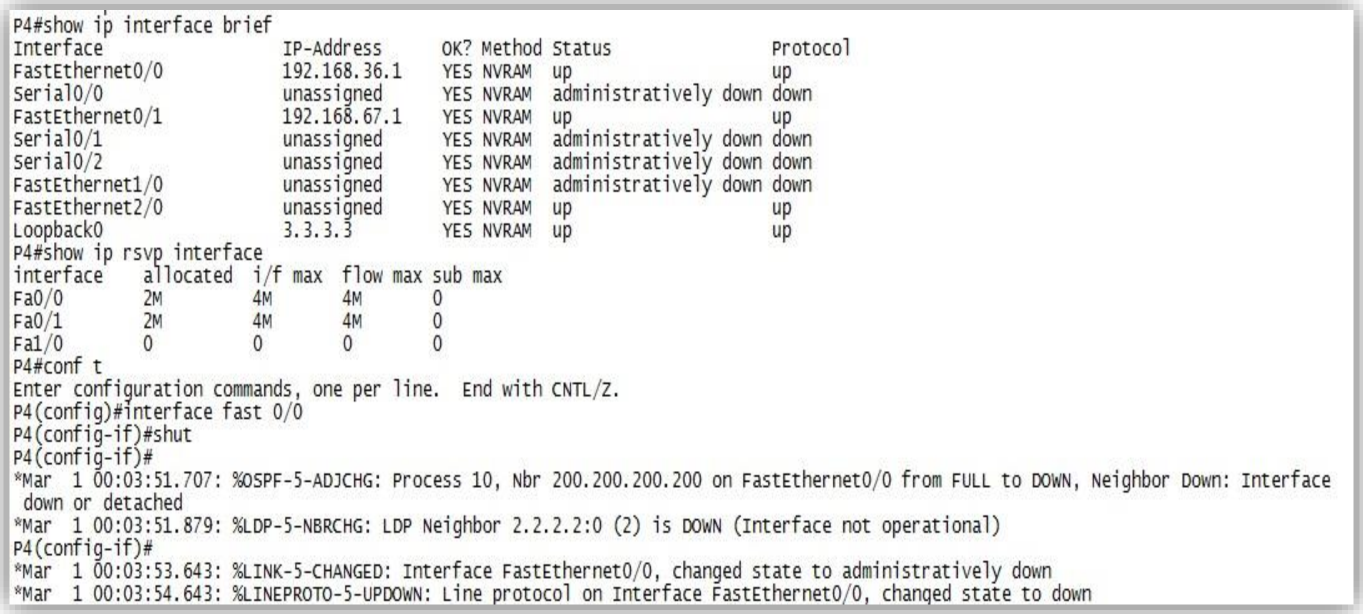


```
CUST_A1#ping 192.168.79.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.79.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/135/168 ms
CUST_A1#traceroute 192.168.79.2

Type escape sequence to abort.
Tracing the route to 192.168.79.2

  1 192.168.12.1 92 msec 40 msec 16 msec
  2 192.168.36.1 [MPLS: Labels 22/26 Exp 0] 68 msec 152 msec 32 msec
  3 192.168.79.1 60 msec 124 msec 80 msec
  4 192.168.79.2 112 msec 164 msec 92 msec
```

```
CUST_B1#ping 192.168.78.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.78.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 128/177/232 ms
CUST_B1#traceroute 192.168.78.2

Type escape sequence to abort.
Tracing the route to 192.168.78.2

  1 192.168.23.1 48 msec 76 msec 52 msec
  2 192.168.34.1 [MPLS: Labels 22/28 Exp 0] 132 msec 144 msec 76 msec
  3 192.168.45.2 [MPLS: Labels 22/28 Exp 0] 116 msec 144 msec 132 msec
  4 192.168.78.1 116 msec 156 msec 156 msec
  5 192.168.78.2 140 msec 136 msec 204 msec
```

Figure 4.3 each customer takes different route tunnel to destination

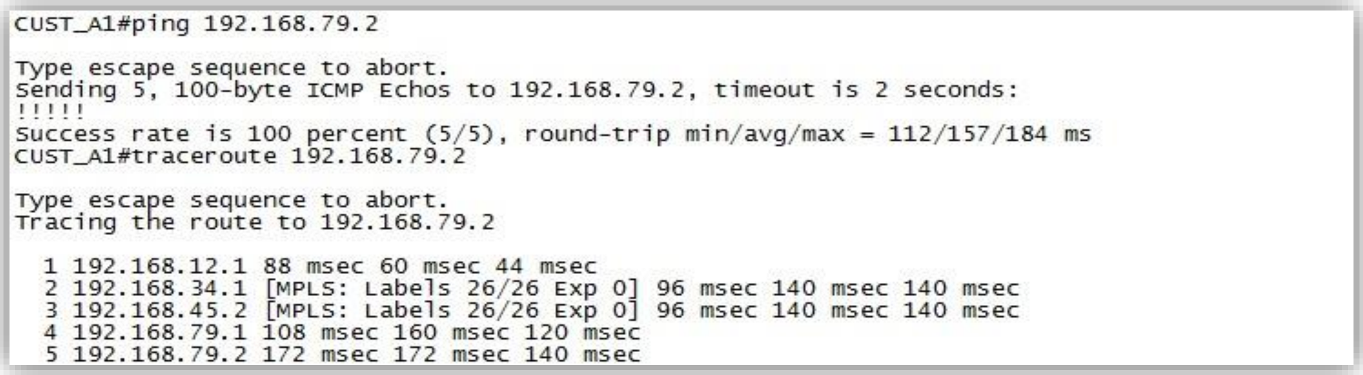
The main tunnel of customer A1 down at figure 4.4 that by shutdown the interface between PE1 and P4 here the metric recalculate and assign the backup route



```
P4#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.36.1    YES NVRAM  up                    up
Serial0/0                  unassigned      YES NVRAM  administratively down down
FastEthernet0/1            192.168.67.1    YES NVRAM  up                    up
Serial0/1                  unassigned      YES NVRAM  administratively down down
Serial0/2                  unassigned      YES NVRAM  administratively down down
FastEthernet1/0            unassigned      YES NVRAM  administratively down down
FastEthernet2/0            unassigned      YES NVRAM  up                    up
Loopback0                  3.3.3.3         YES NVRAM  up                    up
P4#show ip rsvp interface
interface    allocated  i/f max  flow max sub max
Fa0/0        2M         4M       4M       0
Fa0/1        2M         4M       4M       0
Fa1/0        0          0        0        0
P4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
P4(config)#interface fast 0/0
P4(config-if)#shut
P4(config-if)#
*Mar  1 00:03:51.707: %OSPF-5-ADJCHG: Process 10, Nbr 200.200.200.200 on FastEthernet0/0 from FULL to DOWN, Neighbor Down: Interface
 down or detached
*Mar  1 00:03:51.879: %LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (2) is DOWN (Interface not operational)
P4(config-if)#
*Mar  1 00:03:53.643: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
*Mar  1 00:03:54.643: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to down
```

Figure 4.4 the customer A1 tunnel down

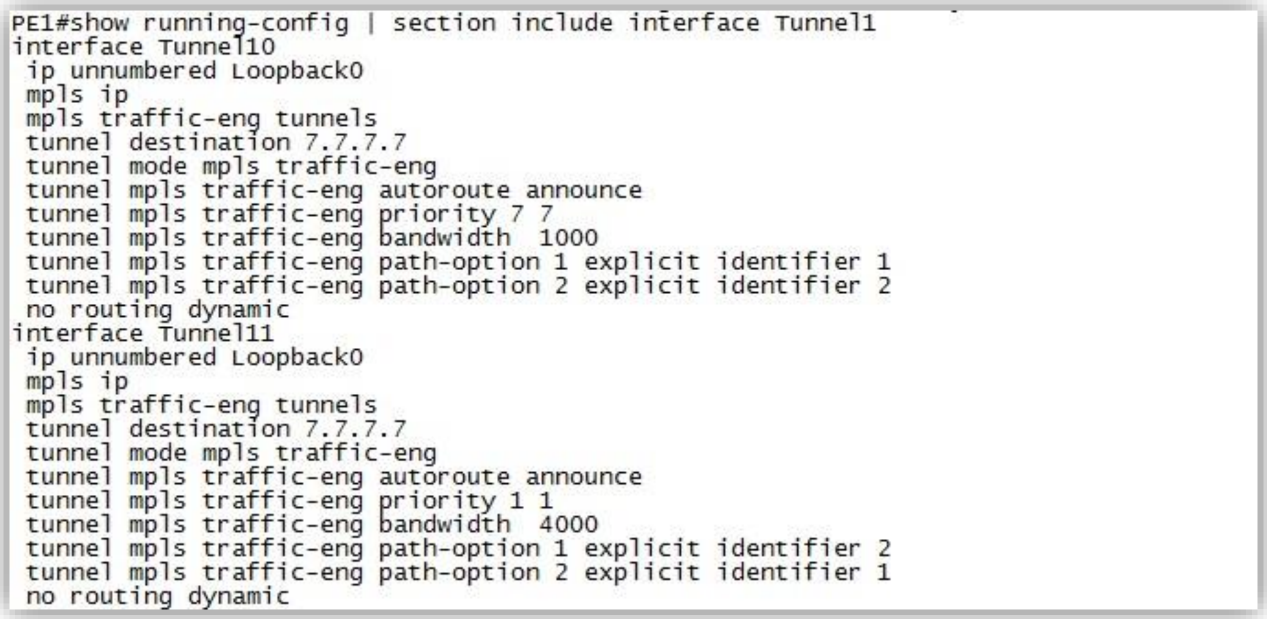
The MPLS-TE assign Figure 4.5 shows that the traffic of CUST_A1 reroutes to the backup route because the CUST_A1 tunnel down here the bandwidth and priority are the same for the two users. The main tunnel of customer's A1 explicated configure also the backup route by path option concept.



```
CUST_A1#ping 192.168.79.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.79.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/157/184 ms
CUST_A1#traceroute 192.168.79.2

Type escape sequence to abort.
Tracing the route to 192.168.79.2

  1 192.168.12.1 88 msec 60 msec 44 msec
  2 192.168.34.1 [MPLS: Labels 26/26 Exp 0] 96 msec 140 msec 140 msec
  3 192.168.45.2 [MPLS: Labels 26/26 Exp 0] 96 msec 140 msec 140 msec
  4 192.168.79.1 108 msec 160 msec 120 msec
  5 192.168.79.2 172 msec 172 msec 140 msec
```

Figure 4.5 when the main tunnel down the traffic reroute to the backup route

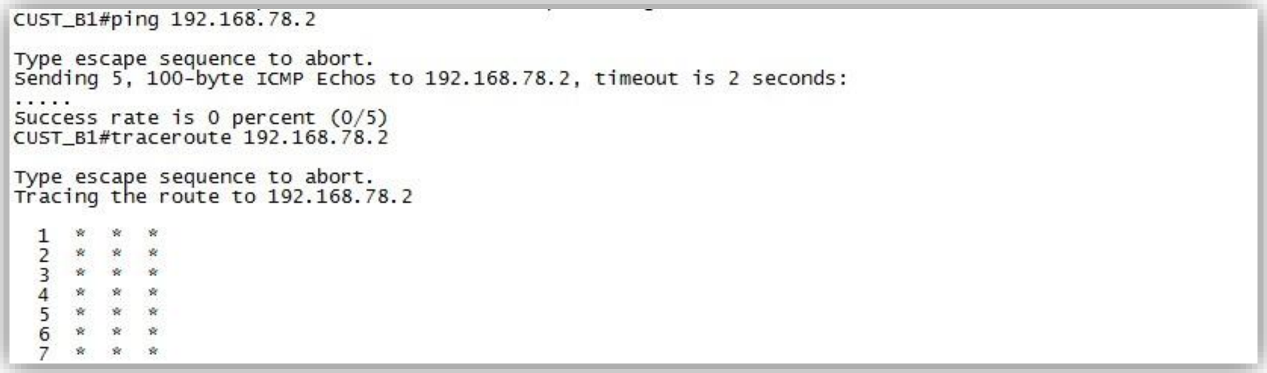
Each tunnel has parameters figure 4.6 show the tunnels parameters such a bandwidth, priorities and the tunnels paths, here CUST_A1 has high priority (1) and bandwidth (4000 K) and CUST_B1 has low priority (7) and bandwidth (1000 K), the all interfaces were configure for 4000 K and the RSVP support the reservation across the IP network



```
PE1#show running-config | section include interface Tunnel1
interface Tunnel10
 ip unnumbered Loopback0
 mpls ip
 mpls traffic-eng tunnels
 tunnel destination 7.7.7.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth  1000
 tunnel mpls traffic-eng path-option 1 explicit identifier 1
 tunnel mpls traffic-eng path-option 2 explicit identifier 2
 no routing dynamic
interface Tunnel11
 ip unnumbered Loopback0
 mpls ip
 mpls traffic-eng tunnels
 tunnel destination 7.7.7.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth  4000
 tunnel mpls traffic-eng path-option 1 explicit identifier 2
 tunnel mpls traffic-eng path-option 2 explicit identifier 1
 no routing dynamic
```

Figure 4.6 the bandwidth, priorities and tunnel path

The CUST_A1 has high priority and bandwidth and the RSVP will reserve 4000 k to customer CUST_A1 because he has high priority and disconnect CUST_B1 tunnel because the link bandwidth is 4000k and the total traffic of CUST_A1 and CUST_B1 equal 5000k. Figure 4.7 show that CUST_B1 can’t reach the branch and this is preemption concept.



```
CUST_B1#ping 192.168.78.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.78.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CUST_B1#traceroute 192.168.78.2

Type escape sequence to abort.
Tracing the route to 192.168.78.2

  1  *  *  *
  2  *  *  *
  3  *  *  *
  4  *  *  *
  5  *  *  *
  6  *  *  *
  7  *  *  *
```

Figure 4.7 the CUST_B1 tunnel disconnected

Chapter Five

Conclusion and recommendation

5.1. Conclusion

The MPLS-TE per VRF is very good way for distribution the traffic at the core networks; it prevents the congestion and increase the performance, stability and availability of services.

According to results shown in chapter four using normal routing protocol not good enough to distribute the traffic because it select the best path and all traffic at the same routing equipment use the same path which cause congestion.

Using MPLS –TE per VRF each user's traffic passed through different path and that by create VRF and bind it to MPLS traffic tunnel with cooperate of MP-BGP and OSPF protocols. Also from result of chapter four the traffic tunnel of CUST_A1 down and traffic rerouted to the backup route there the bandwidth of the backup route it is enough to passing the two customers at the time also the priorities equal for users there is no preemption but at the other scenario there is VIP customer with high priority and the route can't passing all of them because the bandwidth it not enough the preemption process take place and passing CUST_A1 traffic and disconnected CUST_B1 traffic because it has low priority than CUST_A1.

5.2. Recommendation

MPLS-TE controlling the path taken by traffic through a network there are many reasons for network operators want to influence the path traffic is taking in their networks. The most popular reason is improving utilization of network resources. The goal is simple: avoid a situation where parts of the network are congested while others are underutilized. Other reasons for using traffic engineering include ensuring that the path has certain characteristics (e.g. it does not use high-latency links), ensuring that transmission resources are available along a particular path, and determining which traffic gets priority at a time of resource crunch (e.g. following a link cut) so I recommended to do many research about MPLS QoS because it is very important to deliver stable service.

Reference

1. Daniel O.Awduche, (2011)," WILEY SERIES IN COMMUNICATIONS NETWORKING& DISTRIBUTED SYSTEMS", THIRD ADITION.
2. Jong-Moon Chung, Aug 8-11, 2000,"Analysis of MPLS Traffic Engineering paper QoS ".
3. Alcatel Lucent, (2009), "7750 SR OS MPLS Guide"7750 SR OS MPLS Configuration Guide, 2010.
4. Adami, D., Callegari, C., Giordano, S., Mustacchio, F., Pagano, M., & Vitucci, F. (2006)." Signalling protocols in DiffServ-aware MPLS networks", Design and implementation RSVP-TE network simulator, IEEE Global Telecommunications Conference.
5. Implementing Cisco MPLS volume1 Student Guide 2004, Cisco Systems
6. Implementing Cisco MPLS volume2 Student Guide 2004, Cisco Systems
7. Apostolopoulos, G. (2007). "Bringing traffic engineering and resiliency to LDP provisioned MPLS forwarding planes". Proceedings of HPSR '07: IEEE Workshop on High Performance Switching and Routing, 41-47. Piscataway, NJ: IEEE. doi:10.1109/HPSR.2007.4281251
8. Capello, A., Milani, S., Moriondo, C., Rossi, G., Salamandra, P., Perrone, M., & Barone, M. (2005)." Non-stop forwarding behavior and performance in high-end IP routers for ISP's backbone networks". 5th International Workshop on Design of Reliable Communication Networks 2005, October 16, 2005 - October 19, , 2005 279-285. Accession number: 8706709
9. De Ghein, L. (2007). "MPLS Fundamentals. Indianapolis, IN: Cisco Press Juniper Networks (2010). "Network scaling with BGP labeled unicast," Configuration guide, 2010. Retrieved from http://www.juniper.net/us/en/local/pdf/design-guides/8020013-en.pdf .
10. Osborne, E., Simha, A., (2002). "Traffic Engineering with MPLS". Indianapolis, IN: Cisco Press.
11. Soricelli, M. J, Hammond, J.L., Pildush, G. D., Van Meter, E. T., Warble, M.T. (2003). "Juniper Networks Certified Internet Associate": Study Guide. Retrieved from http://www.juniper.net/us/en/training/certification/JNCIA_studyguide.pdf

12. Chuntung Chou , 2002,"Traffic engineering for MPLS-based virtual private networks", paper writtin by *University of Wollongong*, ctchou@uow.edu.au.

13. Yasukawa, S., Farrel, A., & Komolafe, O. (2009). "An analysis of scaling issues in MPLS-TE-core networks". Internet RFC 5439 (Informational), IETF, February 2009. Retrieved from http://www.ietf.org/rfc/rfc5439.txt

14. Shahid Ali,Bilal Zahid Rana,(2011)," Analysis of VoIP over MPLS VPN with IP QoS"

15. Evaluating performance on an ISP MPLS network paper written by Dilmohan Narula, Mauricio Rojasmartinez, Venkatachalapati Rayipati 6 December 2010. Project directed by Professor Jose Santos.

16. Cisco CPT Command Reference Guide–CTC and Documentation Release 9.3 and Cisco IOS Release 15.1(01)SA

17. Hon-Wai Chu, Chi-Chung Cheung, Kin-Hon Ho, and Ning Wang ,"Green MPLS Traffic Engineering", paper

18. Keping Long, Zhongshan Zhang, Shiduan Cheng, 2001," Load Balancing Algorithms in MPLS Traffic Engineering", paper.National Laboratory of Switching Technology & Telecommunication Networks IEEE.

19. Muhammed Naeen,Aslam Yassar,(2008),"Traffic Engineering with Multi-Protocol Label Switching-Performance Comparison with IP networks", Master Thesis

20. Chetan Kumar Ress,"MPLS Traffic Engineering Per VRF/VPN"

21. Luc De Ghein,(2006), "MPLS Fundamentals".

22. Pan, P., Atlas, A., Swallow, G. (2005). Fast Reroute Extensions to RSVP-TE for LSP Tunnels. Internet RFC 4090 (Informational), IETF, May 2005. Retrieved from http://www.rfcarchive.org/getrfc.php?rfc=4090

23. Soricelli, M. J., (2004). Juniper Networks Certified Internet Specialist: Study Guide. Retrieved from http://www.juniper.net/us/en/training/certification/JNCIS_studyguide.pdf

24. http://i4frfewriting.wikispaces.com/R5+-+IP-MPLS+technology

25. https://www.hep.ucl.ac.uk/~ytl/qos/mpls_01.html

APPENDIX 1

Router1:

```
hostname CUST_A1
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet1/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 20
 log-adjacency-changes
 network 1.1.1.0 0.0.0.255 area 0
 network 192.168.12.0 0.0.0.255 area 0
```

APPENDIX 2

Router2:

```
hostname CUST_B1
!
interface Loopback0
 ip address 6.6.6.6 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.23.2 255.255.255.0
 duplex auto
 speed auto
!
router ospf 30
 log-adjacency-changes
 network 6.6.6.0 0.0.0.255 area 0
 network 192.168.23.0 0.0.0.255 area 0
```

APPENDIX 3

Router3:

```
hostname PE1
!
ip vrf A
 rd 1:100
 route-target export 1:100
 route-target import 1:100
 bgp next-hop Loopback1
!
ip vrf B
 rd 1:200
 route-target export 1:200
 route-target import 1:200
 bgp next-hop Loopback2
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface Loopback1
 ip address 22.22.22.22 255.255.255.255
!
interface Loopback2
```

```
 ip address 200.200.200.200 255.255.255.255
!
interface Tunnel10
 ip unnumbered Loopback0
 mpls ip
 mpls traffic-eng tunnels
 tunnel destination 7.7.7.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth  1000
 tunnel mpls traffic-eng path-option 1 explicit identifier 1
 tunnel mpls traffic-eng path-option 2 explicit identifier 2
 no routing dynamic
!
interface Tunnel11
 ip unnumbered Loopback0
 mpls ip
 mpls traffic-eng tunnels
 tunnel destination 7.7.7.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth  1000
 tunnel mpls traffic-eng path-option 1 explicit identifier 2
 tunnel mpls traffic-eng path-option 2 explicit identifier 1
 no routing dynamic
```

```
!
interface FastEthernet0/0
 ip vrf forwarding A
 ip address 192.168.12.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip vrf forwarding B
 ip address 192.168.23.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet1/0
 ip address 192.168.34.2 255.255.255.0
 ip ospf 10 area 0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
 ip rsvp resource-provider none
!
interface FastEthernet2/0
 ip address 192.168.36.2 255.255.255.0
 ip ospf 10 area 0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
 ip rsvp resource-provider none
!
router ospf 30 vrf B
 log-adjacency-changes
 redistribute bgp 1 subnets
 network 192.168.23.0 0.0.0.255 area 0
!
router ospf 20 vrf A
 log-adjacency-changes
 redistribute bgp 1 subnets
 network 192.168.12.0 0.0.0.255 area 0
!
router ospf 10
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 2.2.2.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 7.7.7.7 remote-as 1
 neighbor 7.7.7.7 update-source Loopback0
 neighbor 7.7.7.7 send-community extended
 no auto-summary
 !
 address-family vpnv4
  neighbor 7.7.7.7 activate
  neighbor 7.7.7.7 send-community both
 exit-address-family
 !
 address-family ipv4 vrf B
  redistribute ospf 30 vrf B
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf A
  redistribute ospf 20 vrf A
  no synchronization
 exit-address-family
!
ip route 71.71.71.71 255.255.255.255 Tunnel10
ip route 77.77.77.77 255.255.255.255 Tunnel11
!
ip explicit-path identifier 1 enable
 next-address 192.168.34.1
 next-address 192.168.45.2
 next-address 192.168.57.2
!
ip explicit-path identifier 2 enable
 next-address 192.168.36.1
 next-address 192.168.67.2
```

APPENDIX 4

Router4:

```
hostname P4
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.36.1 255.255.255.0
 ip ospf 10 area 0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
!
interface FastEthernet0/1
 ip address 192.168.67.1 255.255.255.0
 ip ospf 10 area 0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
!
router ospf 10
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 3.3.3.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.255.255 area 0
!
mpls ldp router-id Loopback0
```

APPENDIX 5

Router5:

```
hostname P5
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.34.1 255.255.255.0
 ip ospf 10 area 0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
!
interface FastEthernet0/1
 ip address 192.168.45.1 255.255.255.0
 ip ospf 10 area 0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
!
router ospf 10
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 4.4.4.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.255.255 area 0
!
mpls ldp router-id Loopback0
```

APPENDIX 6

Router6:

```
hostname P6
!
mpls label protocol ldp
mpls traffic-eng tunnels
interface Loopback0
 ip address 5.5.5.5 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.45.2 255.255.255.0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
!
interface FastEthernet0/1
 ip address 192.168.57.1 255.255.255.0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
```

```
!
router ospf 10
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 5.5.5.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.255.255 area 0
!
mpls ldp router-id Loopback0
```

APPENDIX 7

Router7:

```
hostname PE2
!

ip vrf A
 rd 1:100
 route-target export 1:100
 route-target import 1:100
 bgp next-hop Loopback1
!
ip vrf B
 rd 1:200
 route-target export 1:200
 route-target import 1:200
 bgp next-hop Loopback2
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 7.7.7.7 255.255.255.255
!
interface Loopback1
 ip address 77.77.77.77 255.255.255.255
!
```

```
interface Loopback2
 ip address 71.71.71.71 255.255.255.255
!
interface Tunnel10
 ip unnumbered Loopback0
 mpls ip
 mpls traffic-eng tunnels
 tunnel destination 2.2.2.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth  1000
 tunnel mpls traffic-eng path-option 1 explicit identifier 1
 tunnel mpls traffic-eng path-option 2 explicit identifier 2
 no routing dynamic
!
interface Tunnel11
 ip unnumbered Loopback0
 mpls ip
 mpls traffic-eng tunnels
 tunnel destination 2.2.2.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth  1000
 tunnel mpls traffic-eng path-option 1 explicit identifier 2
 tunnel mpls traffic-eng path-option 2 explicit identifier 1
 no routing dynamic
!
interface FastEthernet0/0
 ip address 192.168.67.2 255.255.255.0
 ip ospf 10 area 0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
!
interface FastEthernet0/1
 ip address 192.168.57.2 255.255.255.0
 ip ospf 10 area 0
 duplex auto
 speed auto
 mpls label protocol ldp
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 4000
!
interface FastEthernet1/0
 ip vrf forwarding A
 ip address 192.168.79.1 255.255.255.0
 duplex auto
 speed auto
```

```
!
interface FastEthernet2/0
 ip vrf forwarding B
 ip address 192.168.78.1 255.255.255.0
 duplex auto
 speed auto
!
router ospf 30 vrf B
 log-adjacency-changes
 redistribute bgp 1 subnets
 network 192.168.78.0 0.0.0.255 area 0
!
router ospf 20 vrf A
 log-adjacency-changes
 redistribute bgp 1 subnets
 network 192.168.79.0 0.0.0.255 area 0
!
router ospf 10
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 7.7.7.0 0.0.0.255 area 0
 network 192.168.78.0 0.0.0.255 area 0
 network 192.168.79.0 0.0.0.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 2.2.2.2 send-community both
 no auto-summary
 !
 address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf B
  redistribute ospf 30 vrf B
!
 address-family ipv4 vrf A
  redistribute ospf 20 vrf A
!
ip forward-protocol nd
ip route 22.22.22.22 255.255.255.255 Tunnel11
ip route 200.200.200.200 255.255.255.255 Tunnel10
!
ip explicit-path identifier 1 enable
 next-address 192.168.57.1
 next-address 192.168.45.1
 next-address 192.168.34.2
!
ip explicit-path identifier 2 enable
```

```
 next-address 192.168.67.1
 next-address 192.168.36.2
!
no cdp log mismatch duplex
!
mpls ldp router-id Loopback0
```

APPENDIX 8

Router8:

```
hostname CUST_A2
!
interface Loopback0
 ip address 8.8.8.8 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.79.2 255.255.255.0
 duplex auto
 speed auto
!
router ospf 20
 log-adjacency-changes
 network 8.8.8.0 0.0.0.255 area 0
 network 192.168.79.0 0.0.0.255 area 0
```

APPENDIX 9

Router9:

```
hostname CUST_B2
!
interface Loopback0
 ip address 9.9.9.9 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.78.2 255.255.255.0
 duplex auto
 speed auto
!
router ospf 30
 log-adjacency-changes
 network 9.9.9.0 0.0.0.255 area 0
 network 192.168.78.0 0.0.0.255 area 0
```