وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة عبد الحميد بن باديس مستغانم
Université Abdelhamid Ibn Badis Mostaganem

كلية العلوم والتكنولوجيا مستغانم
Faculté Des Sciences et de la Technologie Mostaganem

Mémoire De Fin D'Etude De
Master Académique

Filiere : Electronique

Spécialité : Systemès des télécommunications

Thème

MPLS (Multi prortocol Label Switching) Applications

Présenté par : DELMI Youssouf

Membres du jury :

Président : Mme BENCHELLAL Amel

Examinateur 1 : Mme BECHIRI Fatiha

Encadreur : Mr RESFA Abbes

Année Universitaire : 2019- 2020

Acknowledgment

Nous adressons en premier notre Seigneur ALLAH qui nous a donné la force et la détermination afin d'élaboré ce modeste travail.

Nous tenons tout d'abord à remercier ***Mr RASFA Abbes*** *notre encadreur de mémoire, pour tout le soutien, l'aide, l'orientation, la guidance qu'elle nous a apportés durant le mémoire, ainsi que pour ses précieux conseils et ses encouragements lors de la réalisation de notre mémoire.*

Nous remercions vivement le président de jury ***Mme BENCHELLAL Amel*** *ansi que les Membres du jury* ***Mme BECHIRI Fatiha*** *d'avoir accepté d'évaluer notre travail*.

Nous tenons ensuite à remercier nos chers parents qui nous ont enseigné la patience, la politesse, le sacrifice et qui nous ont supportés sans condition jusqu'à l'arrivée de ce jour longtemps attendu.

Dedication

J'ai l'honneur de dédier ce mémoire :

À l'homme, mon précieux offre du dieu, mon exemple éternel, mon soutien moral, celui qui s'est toujours sacrifié pour me voir réussir, à toi mon cher père.

À ma chéri, ma vie, ma source de joie et de force qui a toujours été à mes côtés pour me soutenir et m'encourager, maman que j'adore.

À mes chers frères ***Abd Aziz***, ***Amine***.

À ma chers Tante ***Rouba***.

À ma très chère grande mère pour son Douaa tout au long de ma vie, que dieu vous préserve la bonne santé et longue vie.

Abstract

This thesis will describe MPLS networks and explain the need for such technology as well as its contribution to networking in general.

We focused on its major applications, how it can provide WAN connectivity between remote sites in a scalable way by implementing MPLS VPN, how we can optimize network infrastructure usage and manually prioritize traffic based on its nature within the MPLS network via TE and QoS.

Finally, we deployed all the previous features separately in order to make our knowledge more tangible, for that we used GNS3 to get the closest results to real-world implementations, alongside WireShark to capture pertinent packets for demonstration purposes.

Keywords: IP, VPN, MPLS, TE and QoS.

Résumé

Cette mémoire explique la technologie MPLS et sa nécessité dans les réseaux à grandes dimension, ainsi que ses applications principales afin de fournir une connectivité WAN entre les sites distants grâce à l'implémentation d'une architecture MPLS VPN, la manière d'optimiser l'utilisation d'une infrastructure réseau et comment tracer manuellement le trafic au sein d'un réseau MPLS via TE et QoS.

Enfin, l'implémentation a été réalisée au moyen de simulateur GNS3 sur différentes topologie de réseau, et les tests en utilisant wireshark afin de capturer les paquets tout au long du chemin.

Les mots clés : IP, MPLS, VPN et QoS.

ملخص

سوف نعالج في هذه المذكرة تقنية تبديل الوسوم متعدد البروتوكولات ونشرح الحاجة إلى هذه التقنية وكذلك مساهمتها في الإتصالات بشكل عام . ركزنا على تطبيقاتها الرئيسية وكيف يمكن أن توفر اتصال الشبكات الخاصة الإفتراضية ، كيف يمكننا تحسين استخدام البنية التحتية للشبكة وتحديد أولويات حركة البيانات يدويا مرتكزين على طبيعتها داخل الشبكة عبر جودة الخدمات.أخيرا،قمنا بتطبيق جميع الميزات السابقة بشكل منفصل من أجل جعل معرفتنا أكثر تطبيقية ، لذلك استخدمنا محاكي(GNS3) للحصول على النتائج لتطبيقات العالم الواقعي ، جنبا إلى جنب مع محلل البيانات ذات الصلة لغرض البرهنة .

الكلمات المفتاحية : تبديل الوسوم متعدد البروتوكولات (MPLS) ،الشبكات الخاصة الافتراضية (VPN)،جودة الخدمات (QoS) .

Introduction

Telecommunications networks are continually expanding, and new technologies are being developed. The old packet management mechanisms are now redundant due to the latest service prerequisite

As the networks grows in size, the routers become very busy working with routing tables based on IP prefixes. In addition, routers decide the shortest path between the source and destination, and when all the traffic is sent through the shortest path, it can create congestion in the network. In order to deal with this, new mechanisms to improve the networks are needed.

On the other hand, real time traffic requires certain guarantees. When real time traffic is sent through a network, it has to share the resources with other traffic types, and does not get enough resources to be routed without delay, jitter and congestion problems.

Mpls network can provide an interconnected environment to create a converging network capable of delivering QoS and traffic engineering services. Most companies are developing their newest networks, built with mpls, and moving the older ones to mpls. also, it is considered that MPLS can provide a better support to the QoS. The network configured with MPLS can be used to handle performance factors of the network in a better way as compared to just IP routing.

Content

List of Figures

List of Abbreviations

ATM	Asynchronous transfer mode
ACL	Access List
BGP	Border gateway Protocol
CE	Customer edge
CEF	Cisco Express Forwarding
CSPF	Constrained Shortest Path First
DSCP	Differentiated Services Code Point
EGP	Exterior getway protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
ERO	Explicit Route Object
ERP	Entreprise resource planning
EXP	Experimental
FIB	Forwarding information base
HDLC	Level Data Link Control
IGP	Interior getway protocol
LFIB	Label Forwarding Info Base
LDP	Label Distribution Protocol
LSR	Label Switch Router
LSA	Link-state advertisement
MPBGP	Multi-Protocol Border gateway Protocol
MPLS	Multi protocol label switching
NLRI	Network Layer Reachability Information
OSPF	Open shortest path Firset

PE	Provider edge
QoS	Quality of services
RIP	Routing Information Protocol
RSVP	Resource Reservation Protocol
SLA	Service-level agreement
TE	Traffic Engineering
TED	Traffic Engineering Database
TLV	Type Length value
ToS	Type of service
VoIP	Voice over IP
VPN	Virtual Protocol Network
WAN	Wide area network

Chapter 1

MPLS

In this opening chapter, we will dive into the MPLS technology, from its evolution to its main operations; we will discuss it from both a control and a data plane standpoint.

1.1 Introduction

1.2 MPLS Technology

MPLS is a set of protocols that create a new paradigm of how routers forward packets. Instead of forwarding packets based on the IP addresses, MPLS introduces a new way to define packet routes, which is the MPLS label, which changes on a per-hop basis, which allows MPLS to have various factors to influence the packet forwarding mechanism and such as: QoS requirements, the ability to provide privacy for multiple customers connected to the same MPLS network and traffic engineering, which will be discussed further in the upcoming chapters [7] .

1.3 Label Forwarding Logic

The MPLS forwarding model uses labels in order to transmit packets in the MPLS network. Once the packet enters the MPLS network, an LSR (Label Switch Router) which represents any router that is MPLS aware, imposes a label corresponding to the destination IP address after a FIB (Forwarding Information Base) lookup and then forwards it as a labeled packet through the MPLS network until it reaches the egress router in the MPLS network, which will pop the label and forward the unlabeled packet, thus the host and the recipient of the packet do not necessarily need to have any awareness about MPLS whatsoever, the example below demonstrates the process:

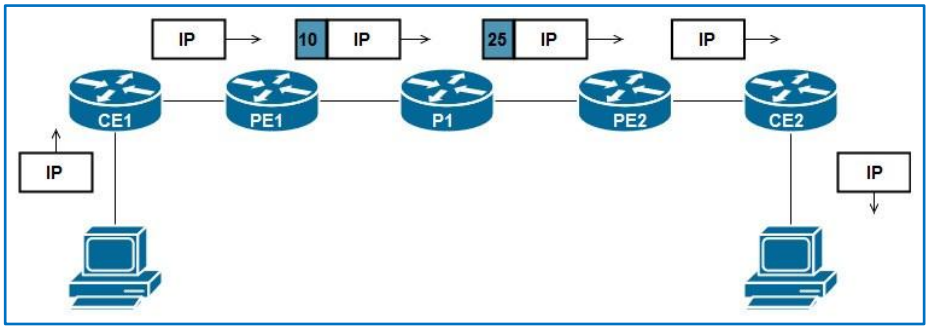

Figure 1.1: MPLS Forwarding Model

In a typical MPLS network, we can discern three types of LSRs, namely:

- Ingress LSR (PE1): It is an LSR that receives an unlabeled packet, then pushes a corresponding label and forwards it.

- Intermediate LSR (P): It is a router that is found within the MPLS network and it processes labeled packets only.

- Egress LSR (PE2): It is an LSR that receives a labeled packet, then pops all the labels and forward it as an unlabeled packet [1].

1.4 Label Distribution Protocol

In order for IP routing to work, we need a control plane protocol to populate the routing table and allow the packets to be forwarded, which is one of the Interior Gateway Protocols (IGP) or Border Gateway Protocol (BGP), however in MPLS, we need an IGP and another control plane protocol in order to populate the Label Forwarding Info Base (LFIB) which is the Label Distribution Protocol (LDP), the next point explain the process of MPLS unicast IP forwarding:

1. The router creates local labels for each IP routing table (FIB) entry from its label range once MPLS is enabled.

2. LDP is enabled and starts multicasting UDP Hello packets to 224.0.0.2 in order to discover and form a neighborship with connected LSRs, note that a router needs a route to the IP address used as an LDP router-id in the remote router.

3. Once Hellos are mutually exchanged, a TCP 3-way handshake is established and the neighborship is up.

4. The neighbors LSRs start exchanging their IP Address-to-Label mappings found in their respective LFIB.

5. Once an eLSR learns a new IP route through its IGP, it allocates a new local label and triggers an LDP update to its neighbors [11].

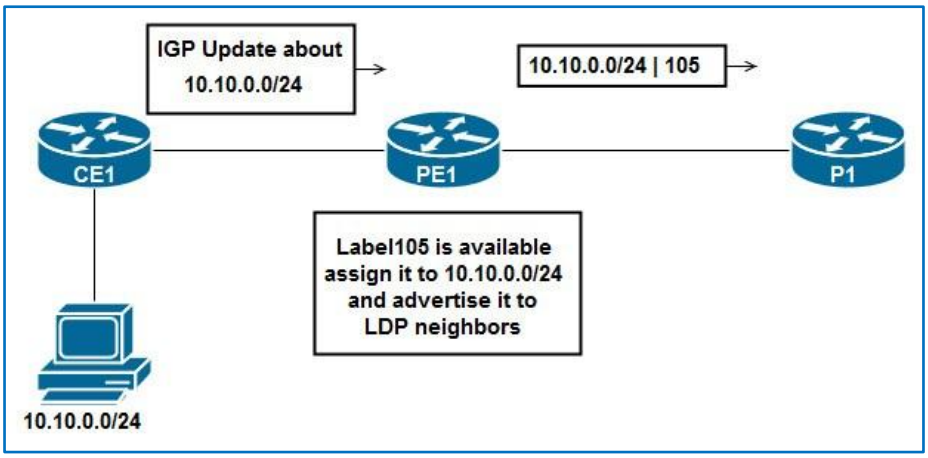

Figure 1.2 : LDP Labels Creation and Distribution

Now, once P1 receives the (10.10.0.0/24|105) binding, it will store it in its LFIB and will create a local binding for 10.10.0.0/24 then advertise it to its neighbors and so on and so forth until every LSR has a local binding for 10.10.0.0/24 and remote bindings advertised by neighbors [4].

This following table covers all LDP features:

LDP Feature	LDP Implementation
Transport protocols	UDP (Hellos), TCP (updates)
Port numbers	646 (LDP), 711 (TDP)
Hello destination address	224.0.0.2
Who initiates TCP connection	Highest LDP ID
TCP connection uses this address	Transport IP address (if configured), or LDP ID if no transport address is configured
LDP ID determined by these rules, in order of precedence	Configuration Highest IP address of an up/up loopback when LDP comes up Highest IP address of an up/up nonloopback when LDP comes up

Figure 1.3: LDP Features

However, we must note that the hello packets are sourced from the physical ports, carrying a transport address and the LSR ID, as the figure 1.4 demon-states:

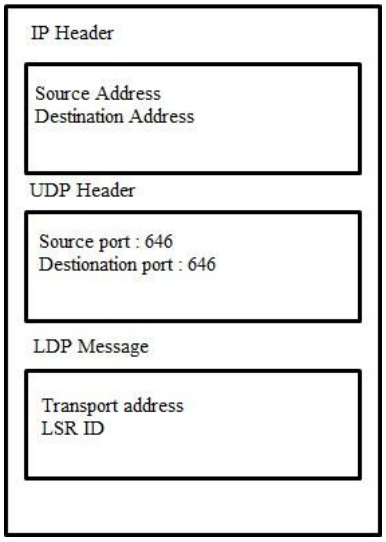

Figure 1.4: LDP Hello Message

These transport addresses are used to form the TCP connection between the two LSRs, instead of using the physical addresses, note that a mutual connectivity between these two transport addresses is mandatory to instruct the TCP session, it is also configurable through the CLI, The LDP identifier (LDP ID) is used to uniquely identify the neighbor Now that the entire MPLS network is up to date, an MPLS Label Switched Path (LSP) is the sequence of labels that must be used to forward the packets correctly to the destination. For exemple, Figure 1.5 demonstrates the (10-25) LSP [4].

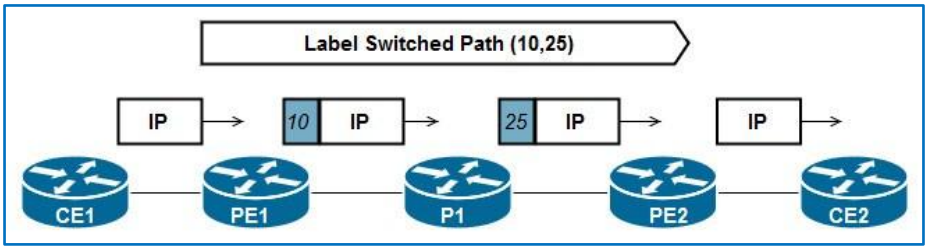

Figure 1.5: Labels 10-25 LSP

We know that in order to populate the routing table in an IP network, we use an IGP such as OSPF, EIGRP Etc. In case of multiple routes to the same destination, we resort to either the Metric or the Administrative Distance in case those routes are provided by the same IGP or multiple IGPs respectively, in order to pick the best routes and add them to the FIB, however, there's no such feature in MPLS to define the best path, that is why MPLS depends on the

existing IP network in order to make forwarding decisions, by choosing the LDP peer that has the destination network next hop address bound to it.

1.5 Information Base Structures

In order to forward packets the LSRs either use the Forwarding Info Base (FIB) or the Label Forwarding Info Base (LFIB) depending on the incoming packet nature, meaning whether it is labeled or unlabeled, both of these tables take their information from the Routing Information Base (RIB) and the Label Information Base (LIB) respectively, which are populated by control plane protocols such as the IGPs for the RIB and the LDP for the LIB, Cisco devices also have the Cisco Express Forwarding (CEF) feature that is a base composed of the FIB and an adjacency table that holds the Layer 2 encapsulation information for each entry in the FIB, resulting in a very quick forwarding decision making, the figure below shows the use of the CEF FIB and LFIB in forwarding packets in an MPLS network.

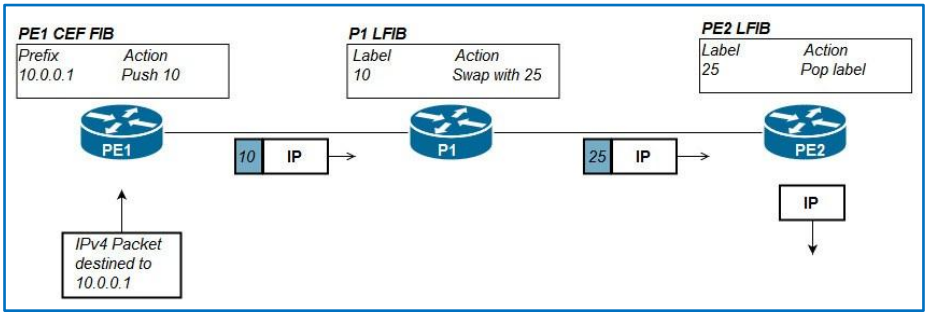

Figure 1.6: Different Information Base Structures

1. PE1 Router receives an unlabeled packet destined to 10.0.0.1, it runs this address against the entries found in the CEF FIB, then finds an entry for the 10.0.0.0/24 network that lists pushing label 10 and forward it through the listed outgoing interface.

2. P1 Router received a labeled packet and thus uses its LFIB to find an entry in order to forward it, it finds an entry that lists swapping label 10 with 25 and then forward it through the listed outgoing interface.

3. PE2 Router receives the packet labeled 25 and runs a lookup against its LFIB table, to find an entry that lists the label should be popped and forwarded unlabeled through the listed outgoing interface [1].

1.6 MPLS Labels

The MPLS header is a 4-byte header composed of four fields, the Label, EXP, BoS and the TTL as the figure below shows:

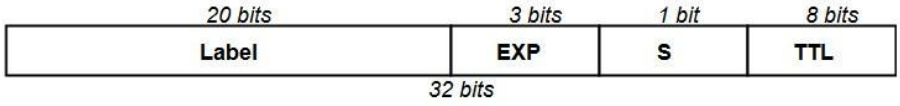

Figure 1.7: MPLS Header

- Label: 20 bits in length hold the label integer value.

- Experimental (EXP): 3 bits field used for QoS marking.
- Bottom of Stack (S): a one-bit flag, when set to 1, it means that the header directly precedes the IP header while the rest are set to zero.

- Time to Live (TTL): 8 bits in length, serves as the number of hops left for the packet before it is discarded by the router, with a maximum of 256 hops.

When an unlabeled packet enters an MPLS network, the LSR pushes a label and copies the IP TTL to the MPLS TTL field, then it will be decreased as it goes through the MPLS network while the IP TTL will stay intact, once the last label is popped the egress LSR copies the MPLS TTL value to the IP TTL field.

1.6.1 Label stacking

MPLS applications need several labels on each packet, this is done by stacking MPLS headers on top of each other, where all the headers have a Bottom of Stack equal to zero except the bottom label that precedes the IP header directly.

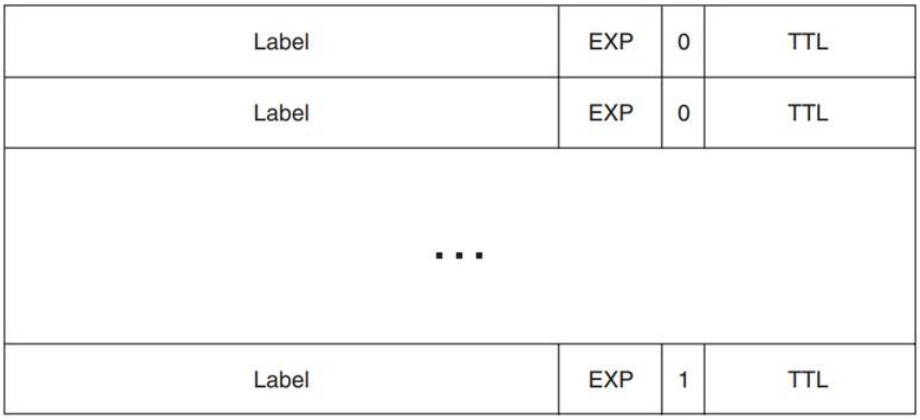

Figure 1.8: Label stack

1.6.2 MPLS Encapsulation

MPLS forwards traffic based on labels without performing any IP lookup thanks to its header placement, right after the layer 2 protocol header (Ethernet, PPP or HDLC) and right before the transported network protocol, some say MPLS header operates at layer 2.5, the following figure shows where it is placed:

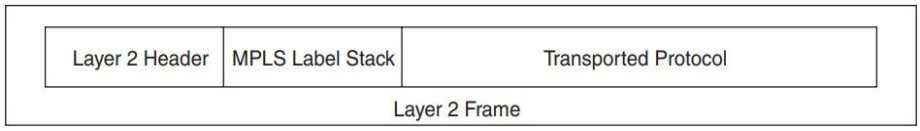

Figure 1.9: MPLS Header Placement

1.7 Conclusion

In this chapter, we have seen how label forwarding functions and how LDP and all the information base structures contribute to it, the MPLS encapsulation and how MPLS logic could be used to introduce several applications for different purposes.

Chapter 2

2 MPLS VPN

In this part of thesis, we will get to discover one of MPLS major applications, MPLS VPN; we will have a brief walkthrough in its history followed by a deeper discussion of its main operations.

2.1 Introduction

As seen in the previous section, MPLS unicast IP forwarding doesn't have that much added value on its own with today's technology and CEF implementation, however, its applications are what makes it so powerful, and one of the most popular applications is the MPLS VPN, which allows service providers to provide some important services such as Layer-3 VPNs that allow WAN connectivity between customers' remote sites, replacing the traditional Layer-2 WAN services and providing the customers with the same privacy they had with WAN services while ensuring that the overlapping prefixes of the private addresses in each site do not cause any problems.

MPLS VPN uses the MPLS unicast IP forwarding inside the SP MPLS network, with some added features on the edge LSRs, alongside Multi-Protocol BGP (MPBGP) to handle the huge number of customers' internetworks. The following figure shows the typical MPLS VPN architecture connecting four remote sites that belong to two different customers.

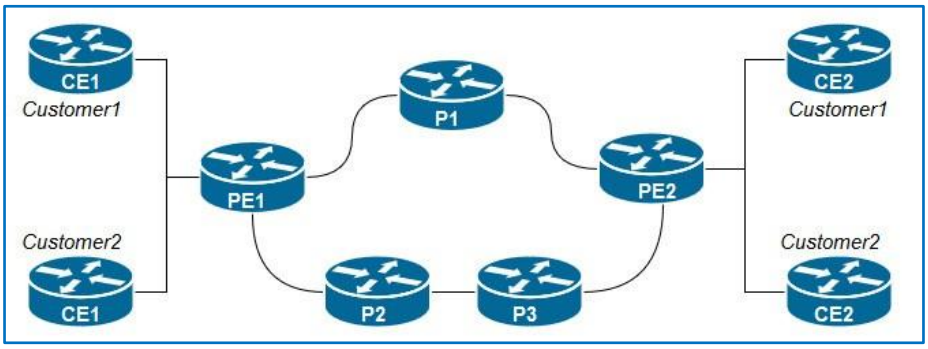

Figure 2.1: MPLS VPN Architecture

We can distinguish three types of LSRs, as follows:

- Customer Edge Router (CE): it has no MPLS awareness and it is directly connected to the PE Router.
- Provider Edge Router (PE): shares at least one link with a CE, contains the customer-tailored VRFs and has multiple control plane protocols enabled.
- Provider Router (P) found in the SP MPLS network, forwards labeled traffic only, between the PEs.

2.2 Control Plane

In order to see the bigger picture of the MPLS VPN, we can focus on the control plane protocols included, P and PE routers run LDP alongside one of the IGPs in order to map the whole MPLS network and execute MPLS unicast IP forwarding, meaning the P and PE can forward packets from the ingress PE to the egress PE to provide the WAN connectivity.

PEs has a range of other tasks, which are oriented towards learning customer routes and tracking of which routes belong to which customers.

PEs has a range of other tasks, which are oriented towards learning customer routes and tracking of which routes belong to which customers. PEs exchange routes with the connected customer edge (CE) routers from various customers, using either eBGP, RIPv2, OSPF, or EIGRP, noting which routes are learned from which customers to keep track of the possibly overlapping prefixes, PE routers do not put the routes in the normal IP routing table—instead, PEs store those routes in separate per-customer routing tables, called VRFs. Then the PEs use Multi-Protocol BGP to exchange these customer routes with other PEs never advertising the routes to the P routers, the figure below shows these control plane concepts [1]:

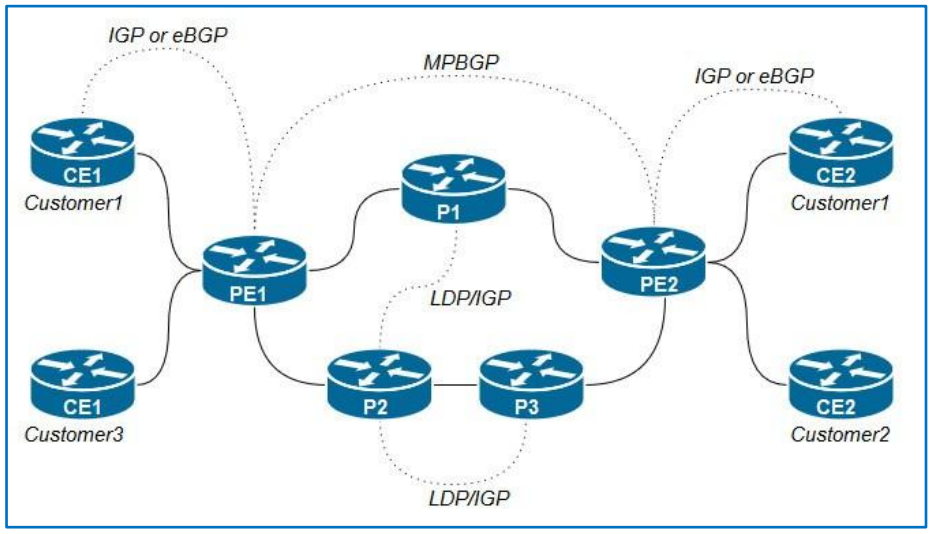

Figure 2.2: MPLS VPN Control Plane Protocols

2.3 MPBGP Instances

MPBGP plays a crucial role in MPLS VPN control plane by advertising customers' routes while keeping them separate between PE routers, to achieve this, three important instances has been introduced:

- Virtual Routing Forwarding (VRFs).
- Route Distinguishers (RDs).
- Route Targets (RTs).

2.3.1 Virtual Routing Forwarding (VRF)

In order to support multiple customers with overlapping addresses schemes, we resort to VRFs, VRFs are basically virtual routers existing in MPLS aware routers only, while typically enabled on PE routers, allowing us to store routes separately for each customer and consequently preventing interferences between customers' overlapping addresses [1]. Typically, a router needs a VRF for each customer connected to the PE in question, could be used to separate traffic between sites of the same customer, where we can use different VRFs for the same customer [4].

The following figure puts the VRF concept more in-depth:

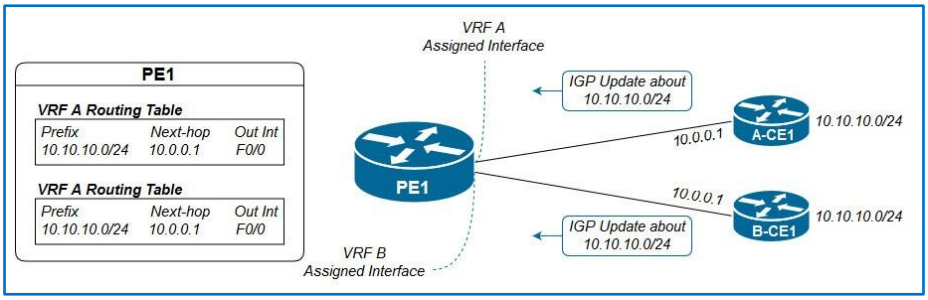

Figure 2.3: VRF Process

We can break this process to these two steps:

1. A-CE1 and B-CE1 send IGP updates to PE1 about the 10.10.10.0/24 subnet.

2. PE1 receives an IGP update about 10.10.10.0/24 on its VRF-A assigned interface, so it places it in the VRF-A routing table.

3. PE1 receives an IGP update about 10.10.10.0/24 on its VRF-B assigned interface, so it places it on the VRF-B routing table.

VRFs, being virtual routers, have these major instances as well:

- Routing Information Base (RIB).
- Forwarding Information Base (FIB).
- A separate instance or process of the IGP used to exchange routes with the CEs.

2.3.2 Route Distinguisher (RD)

While VRFs are used to separately store routes on PEs, they cannot be used to advertise the routes between the customers sites separately, that is why there was an added "number" before the network layer reachability information (NLRI) prefix in the BGP update, that was destined to make each route unique to a customer by assigning a unique number for each customer, all thanks to the MPBGP RFC 4760 that allows the redefinition of an additional number, conventionally called an Address Family to be added before the NLRI prefix, in order to support the MPLS VPN, namely, the Route Distinguisher (RD) [8].

Network Layer Reachability Information (NLRI) is exchanged among BGP routers using UPDATE messages. An NLRI is composed of a LENGTH and a PREFIX. The length is a network mask in CIDR notation (e.g. /25) specifying the number of network bits, and the prefix is the Network address for that subnet [1].

Route distinguishers, which are an MPBGP address family (VPNv4), are used to distinguish between duplicate addresses while advertising for them with MPBGP by adding them before the prefix in the NLRI which will make them unique, since every customer has his own RD, as a result the NLRI will become a 96 bits address composed of:

- A 64-bit Route distinguisher

- A 32-bit IPv4 Address

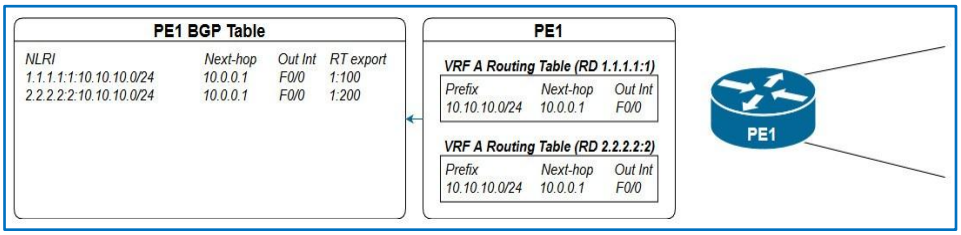

Figure 2.4: IGP to BGP VPNv4 Routes Exportation

What happens is that the PE router redistributes the routes found in the VRFs learned by OSPF into BGP, resulting in extracting an RD from each VRF, and store them in its BGP table so it can advertise them to the other Pes, and since the BGP table has all redistributed routes, the appended RD value will make storing duplicate addresses possible.

RD formatting

RD is eight bytes long, while the first two bytes are reserved for the type, we can configure the rest of the six bytes under three formats:

- 4 bytes integer:2 bytes integer.
- 2 bytes integer:4 bytes integer.
- 4 bytes in DDN: 2 bytes integer.

We must note that the first half of the RD has to be either an ASN number or an IPv4 address.

2.3.3 Route Target (RT)

We have seen how MPLS VPN solved the duplicate addressing scheme, through RDs and VRFs respectively, the only remaining problem, is how do egress PEs know in which VRFs should they place the advertisements?

Route Targets (RT), MPLS uses RTs to determine in which VRF the advertisements should be placed, RTs are also a redefinition of a BGP advertisement update field, called the extended

community, an 8-byte field, while advertisements are bound to only one RD, they can be marked by several RT values, we can discern two types of RTs [1]:

- Export RT: Configured on the ingress PE.
- Import RT: Configured on the egress PE.

When the advertisement arrives, the egress PE runs the export RTs against its import RTs and places the information into the respective VRF when a match occurs, as the following figure shows:

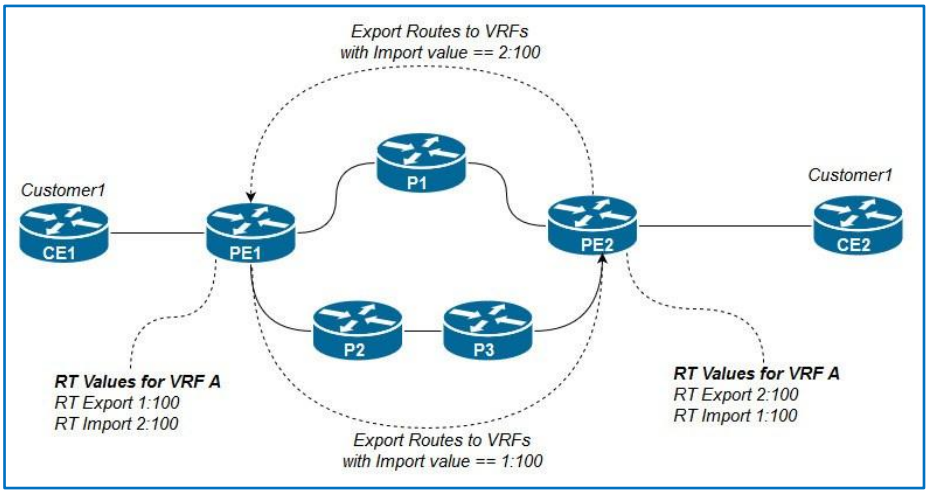

Figure 2.5: Routes Exportation and Importation

It is important to note that this process is unidirectional, if the egress PE in this case, advertises routes from the CEs connected to it, it should have its own Export RT and the ingress PE should have its own Import RT.

2.4 PE-CE Routing

After seeing the bigger picture of how MPLS VPN functions from a control plane standpoint, we need to break its process into two important parts, the PE-CE routing and the PE-PE routing.

2.4.1 Ingress PE-CE Routing

First, we will discuss the ingress PE-CE routing, we need to implement an IGP or even an eBGP instance between the CE and the ingress PE routers VRF in order to advertise routes in the CE site to the PE, more specifically the PE VRF routing table, that is reserved for that customer [8].

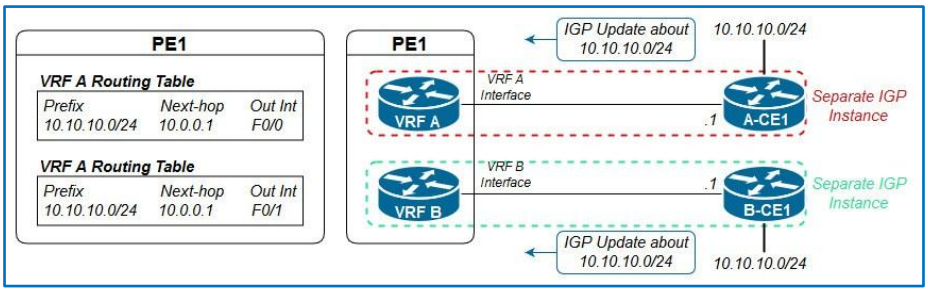

Figure 2.6: Ingress PE-CE Routing Process Overview

As this figure 2.6 shows, We can see that A-CE1 and B-CE1 are connected to PE11 via the interfaces assigned to VRF-A and VRF-B (F0/0 and F0/1 respectively), now we need to run two different IGP instances on PE1 to form an adjacency with each site, in order to advertise routes on the sites, as a result, we can see that each VRF routing table has an entry for 10.10.10.0/24 with A-CE1 and B-CE1 as next-hops learned by the said IGP.

2.4.2 Egress PE-CE Routing

Now, we will see the egress PE-CE routing. After forwarding the MPBGP updates between the PE routers, the routes are placed into the BGP table as VPNv4 routes, then the egress PE router will add the VPN label to its LFIB, then strip the RD and RT values and place the routes as IPv4 routes in the appropriate VRFs, as BGP-learned routes with the ingress PE loopback address as next-hop, afterwards, the egress PE will perform route redistribution on the learned routes, from BGP to the IGP used on the VRF and then advertise them to the CE sites, which will receive these routes as IGP-learned routes with the egress PE2 VRF assigned interface as next-hop, the following figure sums up the whole process.

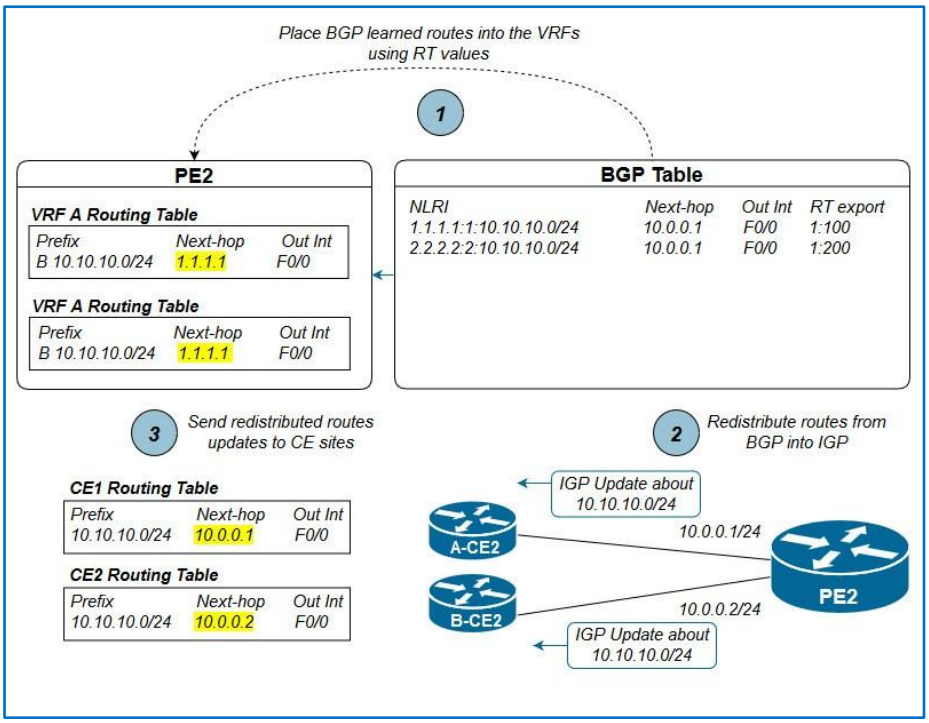

Figure 2.7: Egress PE-CE Process Overview

2.5 PE-PE Routing

After injecting the IGP-learned IPv4 routes in the PE router VRFs as shown in figure 2.6, we need to forward them to the other PE routers in the MPLS network. MPBGP supports several address families alongside IPv4 such as VPNv4, which are IPv4 routes prepended with RDs and appended with RTs, alongside a VPN label and hence the Multi-Protocol BGP naming, commonly, MPLS VPN designs use a loopback as update source on the PE routers, the following scheme demonstrates the process [8].

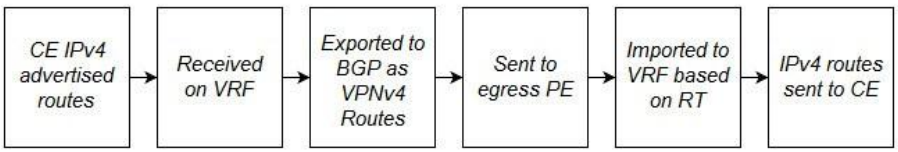

Figure 2.8: PE-PE Operations Steps

1. We advertise CE routes to the PE1.

2. PE1 place them in the appropriate VRF depending on the incoming interface.

3. PE1 performs route redistribution from IGP IPv4 routes to BGP VPNv4 routes.

4. PE1 sends them to the egress PE by LDP-learned label switching.

5. The egress PE will place the BGP-learned routes in the appropriate VRFs based on RT values.

6. The egress PE will now perform a route redistribution to the IGP and then advertise the redistributed routes on the CE site.

2.6 Data plane

Data plane wise, MPLS VPN is different from the MPLS Unicast IP forwarding, where the Ingress PEs need to impose not one but two labels in order to forwards the traffic to the Egress PE while keeping the customers' traffic separate as

follows:

- An outer label with the Bottom-of-Stack set to 0, with a label that allow it to be forwarded to the Egress PE, based on the global CEF FIB and advertised via LDP.

- An inner label with the Bottom-of-Stack set to 1, with a label that defines the VRF this traffic belongs to, signaled by the MPBGP; it also goes by the name of VPN label or BGP label.

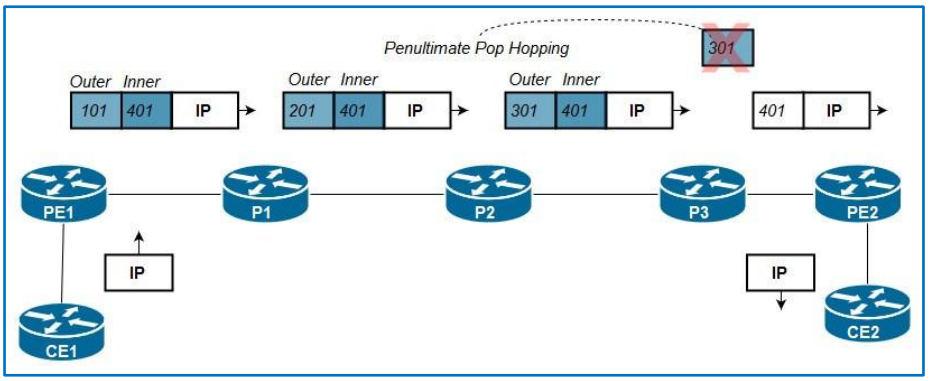

Figure 2.9: MPLS VPN Traffic Forwarding Paradigm

Penultimate Pop Hopping

As seen previously, the MPLS VPN data plane needs two labels in order to function, however, this way, the egress PE would receive two labels stacked and thus do two LFIB lookups just to find out it needs to pop both of these labels and perform an IP lookup in the CEF FIB, which is considered inefficient and unnecessary, so to avoid this and make things more efficient, the Penultimate Pop Hopping (PHP) was introduced, where the second-to-last (penultimate) P router pops the outer label (Global LFIB label) and leaves the VPN label (VRF LFIB label) to be popped by the egress PE router before forwarding the traffic, for this to happen the egress router must advertise label 3 (label 3 is the implicit null label) to the penultimate router via LDP.

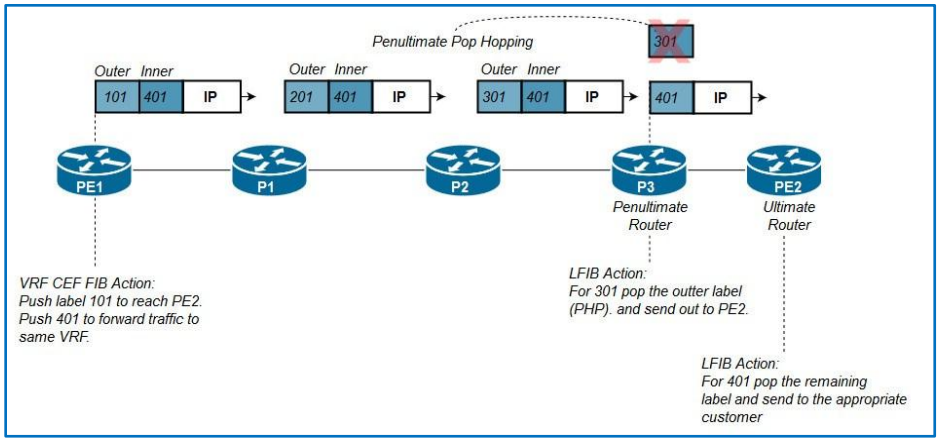

Figure 2.10: Penultimate Pop Hopping

2.7 Overlapping VPN

MPBGP supports multiple extended communities and therefore multiple Route Distinguishers, and this has been in what is called overlapping VPN, it occurs when routes advertisements are meant for different VPNs in the MPLS Cloud. There are many variations to overlapping VPN, the most common is having one site reachable by some of the customers in the same MPLS cloud.

This technology allows us to send VPNv4 route updates to several customers by including a second export RT that matches the customer whom we'd like to have connectivity with, import RT, alongside the primary RT that matches our import RT, the following figure demonstrates how Overlapping VPN can be used:

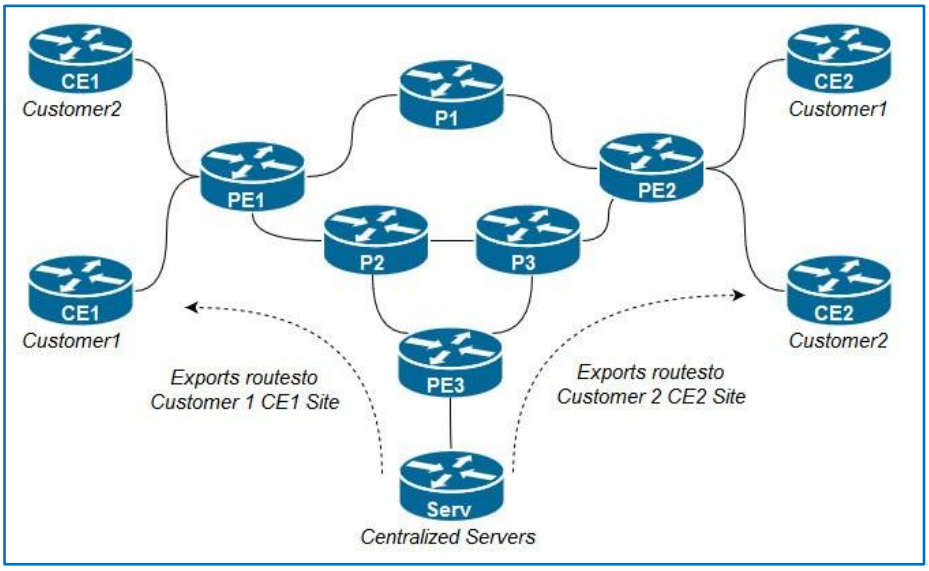

Figure 2.11: Overlapping VPN Design Example

In this example, Customer 1 sites have full connectivity with each other, same goes for Customer 2 sites, however sites Customer1-CE1 and Customer2-CE2 have connectivity with the Centralized Servers connected to PE3 as well, we need to follow these steps to implement this design:

1. Create a VRF for Serv site on PE3.
2. Configure two export RT values for that VRF that match the import RT values of CE1 on PE1 and CE2 on PE2.

We do not need to configure any import RT values for VRF-Serv, as this design doesn't require the centralized servers to have connectivity to the customers'
sites.

Layer2 VPN

The existing MPLS network offers multiple types of VPNs namely Layer2 and Layer3 VPNs, in a layer2 VPN, L2 frames (Could be Ethernet, ATM, Frame Relay or PPP/HDLC) are transported between locations. In the more general case, it's similar to a virtual cable connecting two switches in separate buildings. The VPN has to handle all basic properties of layer 2, taking Ethernet for example: learning MAC addresses (ARP Protocol), replicating broadcast and multicast frames, etc. This can be easily done by tunneling frames over the VPN. Tunneled L2 VPNs are conceptually simpler than L3 VPNs, and if properly implemented, can be completely transparent to applications. On the other hand, it potentially suffers from all problems that plague standard L2 extensions, including security issues and L2 instabilities [1].

In a layer 3 VPN, as previously demonstrated, each side of the connection is on a different subnet, and IP packets are routed through the VPN. The design is potentially more scalable than a L2 VPN, and offers more security than a simple L2 implementation, However, for a host of unrelated reasons L3 VPNs rarely offer the same level of transparency offered by L2 VPNs, and may interfere with applications [10].

2.8 Conclusion

The most popular MPLS-enabled application in operations is MPLS VPN network. This chapter discuss the different operation of MPLS VPN both in control plane in forwarding plan,explain different protocols applied and the label forwarding mechanism.

Chapter 3

MPLS Traffic Engineering

In this chapter, we will see how MPLS TE works. We will get to know how Open Shortest Path First (OSPF) was extended so that it can advertise traffic engineering (TE) information, and that the signaling protocol Resource Reservation Protocol (RSVP) was also extended to suit TE needs. We will discover how the LSR calculates the paths of TE tunnel label switched paths (LSP), how it signals them, and how it calculates their cost. We will also take a look on how to forward traffic into TE tunnels. Finally, we will see how MPLS TE can be tweaked to be used in MPLS VPN.

3.1 Introduction

Traffic engineering TE is a feature that has for a goal to get traffic between two points in the most optimal way, it was first implemented in ATM/Frame relay networks, however, since these technologies started deprecating and IP networks started dominating, it was necessary to implement it, while it is not possible in a pure IP network, it was implemented in an IP/MPLS Network.

IP Routing has one important principle which is to get traffic to the destination in the most efficient way, time-wise, that is why all the metrics of the IGPs are influenced by this principle, which try to choose the most optimal route based on the link cost, bandwidth or number of hops, however, IGPs do not take bandwidth amongst other link attributes such as delay and jitter into account, which might lead to link overutilization and thus data loss, while leaving other links underutilized, we can solve this problem by influencing the traffic flow through the MPLS network depending on the traffic needs and importance to highly optimize our network infrastructure usage, saving us a great deal of planning and cost. TE can dynamically adapt when dealing with unexpected network node failures, where we can prioritize traffic flows according to their importance and in order to preserver the customers SLA agreements

3.2 TE Link Attributes

Every link in the MPLS-enabled network has characteristics that are pertinent to TE, these attributes are flooded between MPLS-TE routers in order to create a TE Database (TED) that is later used to conceive an MPLS-TE Topology that has the best paths for each tunnel to use, for now we will see the following attributes:

Max reservable bandwidth:

As its name indicates, this represent how much of the link bandwidth can be reserved for TE Tunnels.

Attribute Flags:

These are 32-bit flags that have no syntax, operators to indicate 32 different link capabilities and properties for instance use them, and we can indicate that a link has low-latency by configuring its attribute flag to 0 x 4.

In order for the TE Tunnel to take these flags into consideration we must configure the affinity bits and mask on it, which are also 32 bits long. Both the Attribute flags and the Tunnel Affinity bits will be masked according to the Affinity mask and then the unmasked part must be identical in order for the Tunnel to accept this path as a potential path otherwise it the path crossing this node will be dropped.

This could be used for instance to prevent bulk data traffic tunnels from using VoIP links by playing on the affinities of both tunnels.

TE Metric:

Metric are used by IGP protocols in order to pick the best route amongst all the routes provided by one IGP, and since MPLS TE needs a metric to choose the best available path, so by default they use the IGP metric, however there's the TE metric that also could be used as the primary metric for MPLS-TE, note that TE metrics are by default equal to IGP metrics.

This duality can be useful, in case we want to influence the path taken by the TE Tunnel, because instead of influencing the IGP link metric, which would influence IP traffic as well, we can instruct the TE Tunnel to use TE link metric and change it accordingly, this way we'll have two different topologies, one for IP and the other for MPLS TE, this feature is called Dual TE Metrics [2].

3.3 TE Information distribution

In the previous section, we have seen some of the MPLS-TE relevant link attributes, in this section we will see how these characteristics are flooded in the MPLS-TE network.

Unlike Distance vector protocols that only advertise their best routes to their peers, Link-state protocols do advertise a full update about all their links, this way, every router has a complete idea about the topology. In order for the head router (the router that will create the TE-Tunnel) to choose the best path for its Tunnel, it needs to have all the relevant topology information, which are the aforementioned link attributes mostly, and that explains why we use Link State IGPs in MPLS networks when deploying Traffic engineering, the next title will explain how OSPF is used to populate the TE Database [2].

3.4 OSPF-TE Adaptation

In order to adapt the OSPF protocol to Traffic, there has been added three LSAs following the RFC 2370, LSA 9,10 and 11, having different flooding scopes, we'll be using the LSA 10 since it has the perfect flooding scope for us, which is areawide while the LSA 9 an 11 have a link-local scope and an AS scope respectively, these aforementioned LSAs are called Opaque LSAs, in order for routers to recognize routers that can support opaque LSAs, an O-bit was introduced into the OSPF options field, this field exists in:

- Hello Messages.
- Database Description.
- All LSAs.

More specifically, OSPF uses TLVs to transport the Link attributes, two TLV types exist: Router TLV, carrying the TE RID and a Link TLVs, that is composed of several sub-TLVs that describe the link characteristics, such as:

- Traffic Engineering Metric (4 bytes).
- Maximum reservable bandwidth (4 bytes).
- Administrative group or Attribute Flags (4 bytes)[2].

OSPF-TE Advertisements TLV (type-length-value or tag-length-value) is an encoding scheme used for optional information element in a certain protocol. The type and length are fixed in size (typically 1-4 bytes), and the value field is variable.

3.4.1 OSPF-TE Advertisements

Same as IP IGP advertisement, TE-IGP advertisements are triggered by certain events which are:

- **Link status change :**

  The link status (up or down) means either that new paths are available that could be the most optimal or that a path in use has failed and thus new calculations are needed.

- **Manual Configuration Change**

- **Periodic flooding:**

  Manual Configuration Change all IGPs have a timer for their periodic flooding, that is usually relatively long, the default periodic flooding timer for OSPF is 30 minutes however opaque LSAs (TE Information) are flooded every three minutes.

- **Changes in bandwidth (reserved/unreserved):**

  Small changes in the available bandwidth do not necessarily trigger an opaque LSAs flooding, however, when the changes are relatively important, TE information is flooded, by default there are several triggering milestones for both ways, meaning for when the available bandwidth is being freed or being reserved, the concept is detailed in the following figure.

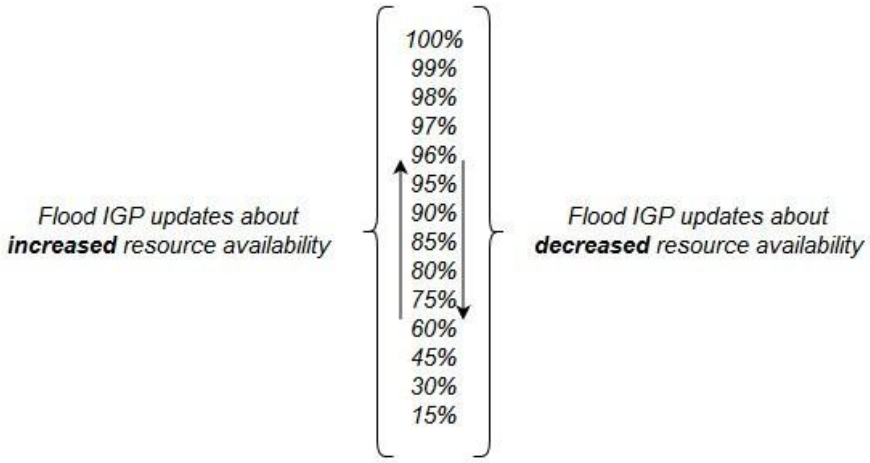

Figure 3.1: IGP-TE Update Triggers Thresholds

- Tunnel Setup Failure:The head end router doesn't request for a path for its Tunnel unless it has done all the needed calculations that prove that a certain path is capable of holding the tunnel, a setup failure means one of the IGP-provided information has changed and the setup cannot be done, this require the updated TE information and hence flooding must happen [2].

3.5 MPLS-TE Tunnels (Attributes and Path calculation PCALC)

We have seen in the previous sections the necessary TE information that the head end router needs and how it is forwarded throughout the MPLS network; in this section, we will discuss the Tunnel attributes and other relevant operations. MPLS TE Tunnel has several characteristics; we will see the following in detail:

Affinity bits and mask

As mentioned in the Link attributes section, these are 32-bit succession that is masked by the affinity mask. As the Figure 0x0 shows, in order for the Tunnel to choose a certain path, each link masked Attribute flags must conform to the masked bits of the Affinity bits.

Tunnel destination

This is the destination (or the tail end) router TE Router ID, where the tunnel traffic should be forwarded.

Desired bandwidth

This attribute specifies the least bandwidth that the Tunnel path should have on every link from the head end router to the tail end router.

Tunnel Priorities

One of the advantages of using Traffic Engineering is the adaptation to network changes be it node failures or a new configuration, where you can set certain priorities to each tunnel and thus prioritize traffic on your network when needed, for this, we use two priorities: Setup and Hold priorities, to define which tunnel should take the link reservation when of course, we're short on link resources, such as bandwidth, they're ranged from 0 to 7, the lower the priority the more important the link is, if Tunnel 1 setup priority is lower than Tunnel 2 hold priority hold, it means that Tunnel 1 is more important than Tunnel 2 and it can preempt it and take over the link reservation, that is why we should not set a hold priority that is lower than the setup priority, because then, the tunnel would preempt another tunnel because of its low Setup priority but then get preempted once it is setup, because of its hold priority.

Path options

Once you configure a TE Tunnel, it needs a TE LSP that meets all its preconfigured requirements to route traffic through the network to the tail end router, these TE LSPs are called paths and we have two types:

1. Explicit Paths: these are the paths that you configure manually, by specifying each node that the Tunnel needs to go through until the tail end, but this is mostly administratively challenging in relatively big MPLS networks.

2. Dynamic Paths:, however, Dynamic paths are calculated by the head end router by running the path calculation algorithm (PCALC or CSPF) against the TE Database (TED), populated by the link state IGP, taking into account all the Tunnel requirements and the link attributes pertinent information the head end router only needs the tail end router address.

There can be multiple dynamic and explicit paths for a single Tunnel to try and use, each path has a preference value (the lower the better) the tunnel will signal the most preferred path by sending RSVP messages (Will be discussed in detail in the next section), if the tunnel happens to not find any working path, it turns to a down interface status.

Re-optimization

As we mentioned before, MPLS Traffic engineering has for a goal to forward traffic between two edges in the most optimal way, which is covered prior to the Tunnel setup by Setup and Hold priorities, however, MPLS TE needs to adapt if a new optimal paths are available after the setup, for that there has been introduced 3 triggers to help reoptimize the Tunnels paths:

1. Periodic Re-optimization: There is a timer that is set to one hour by de-fault.

2. Event-driven Re-optimization: We could instruct head end routers to perform re-optimization once a link is operational and on standby for TE operations, this is isn't functioning by default.

3. Manual Re-optimization: You can instruct the head end router to perform a re-optimization globally or per-tunnel through the CLI [2].

3.6 RSVP Tunnel Establishment (RSVP)

Up until now, we have seen how the head end router finds the most optimal paths for its configured Tunnels after using the information in the TED gathered by the LS IGPs, now we will see how these Tunnels are established to be ready to

forward traffic.

MPLS TE uses Resource Reservation Protocol (RSVP) in order to signal the Tunnel chosen path on each interface in the calculated path and consequently to reserve the link a resource for the said tunnel, for this RSVP uses two messages: PATH and RESV, it operates this way:

1. The head end router sends a PATH message that has a list of IP addresses of the calculated path stored in the Explicit Route Object (ERO) to the first hop of the path.

2. Each LSR in the path receiving the PATH message, removes its address from the ERO and forwards it to the next hop, until it reaches the tail end router.

3. Tail end router returns a RESV message in the exact path to the head end router

4. Once the head end router receives the RESV message error-free, the tunnel has all the link resources reserved and it becomes operational.

RSVP is also responsible of defining the path as an LSP path, making the tunnel have its own labels, for this, the RESV message has a label field, where each router receiving a RESV, places the label in the received message as an outgoing label for that tunnel in its CEF LFIB and then places its local Tunnel label in the RESV to be forwarded to the next hop on the way from the tail end to the router end, note that the tail end sends an explicit null in the label field of the forwarded RESV message, in order for the Penultimate-Pop-Hopping to occur, Fig 0xF sums up the RSVP signaling operation:

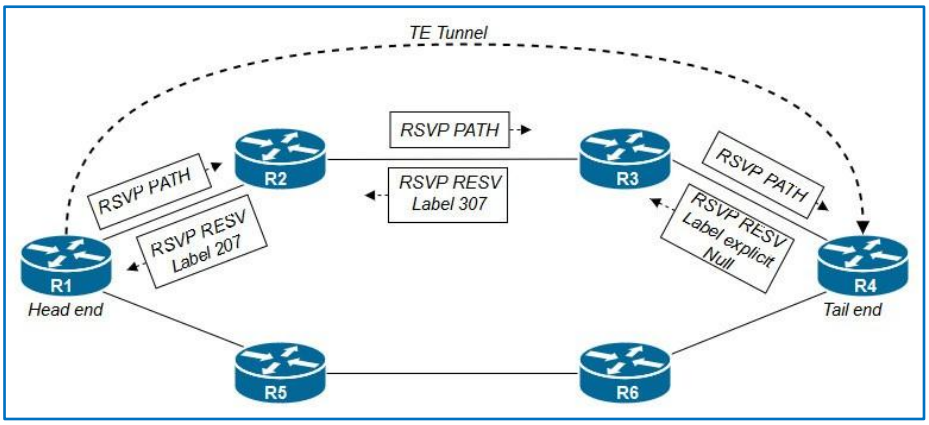

Figure 3.2: RSVP-TE Operations

This means that the RSVP is responsible for the label distribution meaning that LDP is not mandatory for the MPLS TE to operate, we must note that Tunnel paths are unidirectional [2].

3.6.1 Enabling TE Tunnels

After making the tunnel operational, we need to instruct traffic flow to take the tunnel LSP path. There are different ways to accomplish that; these are some of the methods:

1. Static routing : This is done through configuring a static route to a destination with the tunnel interface as the outgoing interface.
2. Policy-based Routing : PBR allow us to direct traffic in a more granular way, based on the source, the protocol used or the kind of traffic, to a specific interface, by creating a policy and map it to the incoming interfaces.

3. Forward Adjacency : This feature makes it possible to advertise for tunnels in the IGP advertisements making it look as if the tunnels were direct links between the head end router and the tail end router, However, we must configure a second tunnel for each existing tunnel that is being advertised, because Tunnels are unidirectional where IGP routes are two-way routes[2].

3.7 MPLS VPN Adaptation

Forwarding from CE to CE works with all VPNv4 traffic carried over the TE tunnels. If the BGP next hop is not the same as the TE tunnel destination, LDP must be run in the core and on the TE tunnel.

This section details how MPLS VPN and TE can coexist in one network, allowing MPLS VPN to benefit from TE advantages, however, this coexistence requires some tweaking to avoid packet forwarding possible failures.

We can distinguish two scenarios for this implementation, setting up a tunnel between the ingress and egress PE routers, or between LSRs in the MPLS network.

3.7.1 PE-PE Routers Tunnel

We can instruct VRF traffic to flow over a TE Tunnel linking both PE routers, or more correctly, a dual-tunnel linking them back and forth, in this scenario, the LSRs do not need LDP to distribute labels because that is taken care of by RSVP RESV messages as seen previously, the traffic will have two distinct labels stacked, the outer label will be the Tunnel label while the inner label will carry the VPN label.

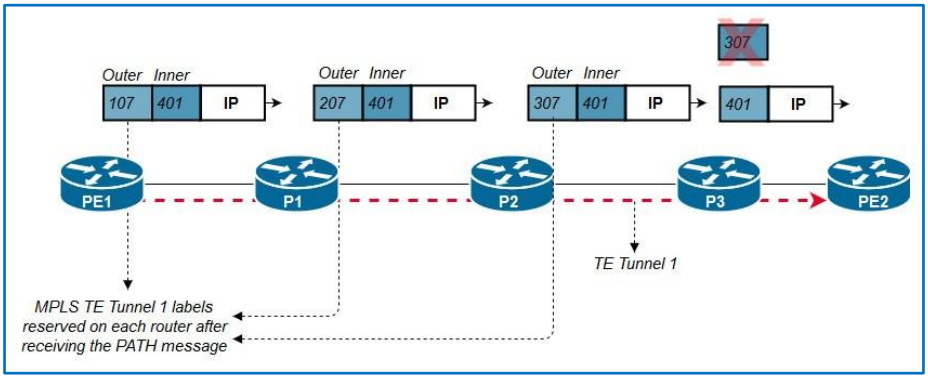

Figure 3.3: PE1-PE2 Tunnel Path

In order to instruct the VPN traffic to choose the Tunnel over the traditional LSP, we must point the tunnel as the next hop for the BGP VPNv4 route instead of the egress PE loopback address conventionally used [2].

3.7.2 PE (or P)-P Tunnel

This following scenario requires tweaking and adaptation as it cannot function properly on its own.

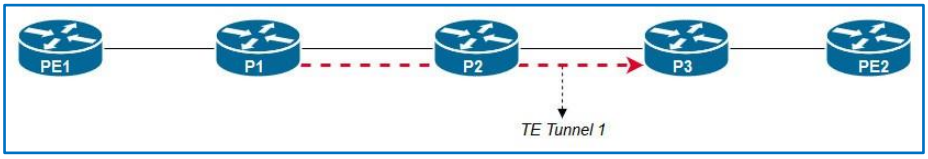

Figure 3.4: P1-P3 Tunnel Path

This way, the traffic will leave the P1 router having two labels, one for TE and one for VPN, however, with the P tail end router sending an implicit null in the RESV message to the upstream router (P2 in this case) for the PHP to occur, P2 router will receive a packet with only a VPN label on it, on which it'll take action by either dropping the packet because there is no match in the LFIB table or by forwarding it to the wrong destination in case a match for that VPN label in its LFIB, and consequently never reach the egress PE router, since VPN labels are only meant to be used by the PE egress routers.

In order to solve this problem, we need to follow these two steps:

1. Enable LDP on all links, in order to forward labeled traffic to the head-end router and from the tail-end router to the egress PE router.

2. Configure a targeted LDP session between the head-end and tail-end routers, as a result, the tail-end router, will advertise its labels to the head-end router, consequently, the head-end will have an LFIB entry for the egress PE router, so it will have not two but three labels:

   a. TE Label (Top label): contains the labels of the tunnel path LSP towards the tail-end.

   b. LDP label (Second label): will contain the advertised labels to reach the egress PE router.

   c. VPN label (Bottom of Stack): contains the VPN label previously advertised by MPGBP.

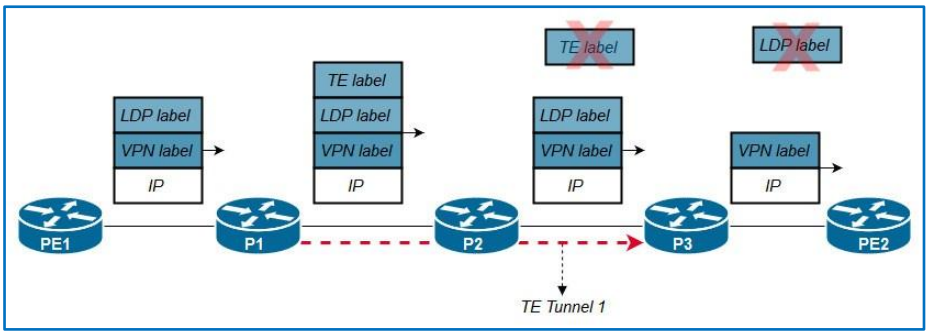

Figure 3.5: P-P Tunnel Data plane Overview [2]

3.8 Conclusion

In this chapter, we explained the need for Traffic Engineering in networking, we walked through the link attributes related to TE and how OSPF was tailored to disribute these information to the head-end router, how these information are used to calculate and establish a TE tunnel.

Chapter 4

4 MPLS Quality of Service

In this chapter, we will see different QoS models, we will learn about QoS tools, how to protect voice, video and data traffic using various QoS mechanisms, we will be Describing the various classification options for distinguishing one packet from another, also the various marking options, we will Discusses the various tools that can be used to meter and regulate packet flows, including policies, shapers and markers, we will discuss about Congestion Management and Avoidance Tools, finally we will see different kind off DiffServ tunneling models and how to implement them on the MPLS network .

4.1 Introduction

Quality of service (QoS) has become popular the past few years. Few networks have unlimited bandwidth, so congestion is always a possibility in the network. QoS is a means to prioritize important traffic over less important traffic and make sure it is delivered; there are four characteristics of network traffic that we must deal with:

Bandwidth is the speed of the link, so each traffic has one queue we can configure the router so the queue can get a percentage of the bandwidth. Delay is the time it takes for a packet to get from the source to a destination, this is called the one-way delay. The time it takes to get from a source to the destination and back is called the round-trip delay. Jitter is the variation of one-way delay in a stream of packets. Loss is the amount of lost data, usually shown as a percentage of lost packets sent.

4.2 End to End Qos Models

End-to-end QoS is the ability of the network to deliver service required by specific network traffic from one end of the network to another, there are three types of service models: best effort, integrated, and differentiated services.

Best-Effort Service

Best effort is a single service model in which an application sends data whenever it must, in any quantity, and without requesting permission or first informing the network. For best-effort service, the network delivers data if it can, without any assurance of reliability, delay bounds, or throughput. Best-effort service is implemented by FIFO

Integrated Service

In this model the application requests a specific kind of service from the network before it sends data. So we use the signaling protocol Resource Reservation Protocol (RSVP). The hosts signal to the network via RSVP what the QoS needs are for the flows of traffic that they send.

Differentiated Service

The DiffServ bits in the IP header to qualify the IP packet to be of a certain QoS. The routers look at these bits to mark, queue, shape, and set the drop precedence of the packet. The big advantage of DiffServ over IntServ is that the DiffServ model needs no signaling protocol.

4.3 Quality of Service Tools

4.3.1 Classification and Marking

Classification: an action that sort packets into different traffic types, to which different policies can then be applied. Classification of packets can happen without marking. Marking: Writes a value into the packet header. It is usually establishes a trust boundary at the network edge or at the intersection between two different networks, where preexisting packet markings are accepted or rejected (and as such, re-marked). Marking also can be used in other locations in the network and is not always used solely for purposes of classification [6].

IP Precedence: Differentiated QoS

DiffServ with IP Packets: IP precedence utilizes the 3 precedence bits in the IPv4 header's Type of Service (ToS) field to specify class of service for each packet, as shown in Figure 4.1:

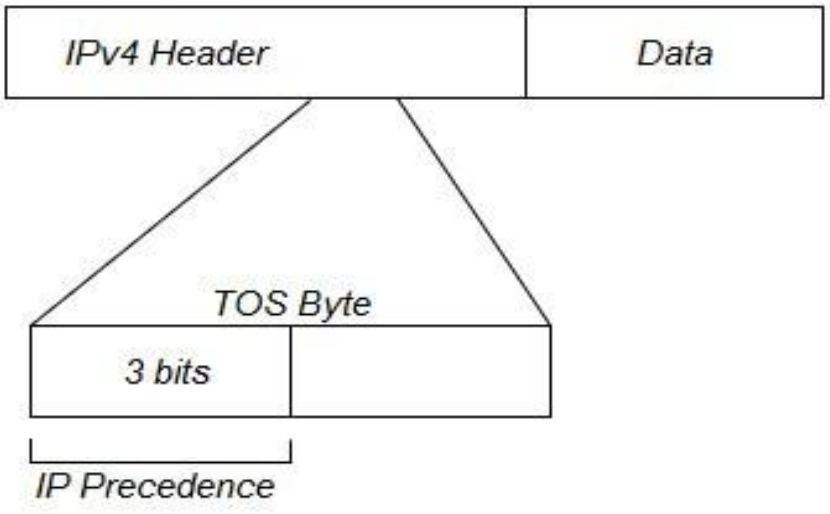

Figure 4.1: IP Precedence ToS in an IP Packet Header

The usage of the precedence bits for QoS is now widely used throughout the world for many networks. The drawback of the precedence bits, however, is that only three exist, which means there are eight levels of service. Therefore, the IETF decided to dedicate more bits for QoS. The four TOS bits were deprecated, and three of them were assigned to DiffServ QoS, in addition to the three precedence bits. DiffServ ended up with six bits, providing more than enough levels of QoS [5].

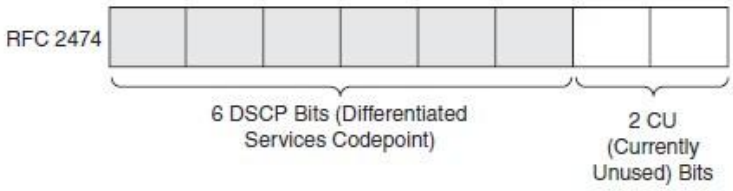

Figure 4.2: DiffServ with IP packets

DiffServ with MPLS Packets : There are three EXP we can use these bits in the same way that we use the three precedence bits in the IP header, so the label switched path is changed to an E-LSP

indicating that the label switching router will use the EXP bits to schedule the packet and decide on the drop precedence .

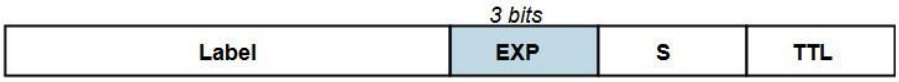

Figure 4.3: DiffServ with MPLS Packets

PBR: Policy-Based Routing

Policy-Based Routing (PBR) allows to classify traffic based on extended access list criteria, set IP precedence bits, and even route to specific traffic engineered paths that may be required to allow a specific QoS through the network. By setting precedence levels on incoming traffic and using them in combination with the queuing tools, you can create differentiated service. These tools provide powerful, simple, and flexible options for implementing QoS policies in your network [9].

Using policy-based routing, route maps are made to match on certain flow criteria and then set precedence bits when ACLs are matched. The capability to set IP precedence bits should not be confused with PBR's primary capability: routing packets based on configured policies. Some applications or traffic can benefit from QoS-specific routing-transferring stock records to a corporate office (for example, on a high-bandwidth, high-cost link for a short time), while transmitting routine application data such as e-mail over a lower-bandwidth, lower-cost link. PBR can be used to direct packets to take different paths than the path derived from the routing protocols. It provides a more flexible mechanism for routing packets, complementing the existing mechanisms provided by routing protocols.

Also available using route maps is the capability to identify packets based on Border Gateway Protocol (BGP) attributes such as community lists and AS paths. This is known as QoS policy propagation via Border Gateway Protocol[3].

CAR: committed access rate

Similar in some ways to PBR, the CAR feature allows to classify traffic on an incoming interface, it also allows specification of policies for handling traffic that exceeds a certain

bandwidth allocation. CAR looks at traffic received on an interface, or a subset of that traffic selected by access list criteria, compares its rate to that of a configured token bucket, and then takes action based on the result (for example, drop or rewrite IP precedence) [9].

CAR is used to police traffic flows to a committed access rate. CAR does this with a token bucket. A token bucket is a bucket with tokens in it that represent bytes (1 token = 1 byte). The bucket is filled with tokens at a user-configured rate. As packets arrive to be delivered, the system checks the bucket for tokens. If there are enough tokens in the bucket to match the size of the packet, those tokens are removed and the packet is passed (this packet conforms). If there aren't enough tokens, the packet is dropped (this packet exceeds) [6].

NBAR: Dynamic Identification of Flows

Cisco's newest method of classification is Network Based Application Recognition (NBAR). This method is used to identify the traffic it can identify various applications that use ephemeral ports .NBAR does this by looking at control packets to determine which ports the application decides to pass data on. NBAR adds a couple of interesting features that make it extremely valuable. One feature is a protocol discovery capability. This allows NBAR to baseline the protocols on an interface. NBAR lists the protocols that it can identify and provides statistics on each one. Another feature is the Packet Description Language Module (PDLM), which allows additional protocols to be easily added to NBAR's list of identifiable protocols. These modules are created and loaded into Flash memory, which then is uploaded into RAM. Using PDLMs, additional protocols can be added to the list without upgrading the IOS level or rebooting the router.

4.3.2 Congestion-Management

FIFO: First in, First out

FIFO is an acronym for First In First Out. This expression describes the principle of a queue or first-come-first-serve behavior: what comes in first is handled first, what comes in next waits until the first is finished etc. Simply, FIFO queuing involves storing packets when the network is congested and forwarding them in order of arrival when the network is no longer congested. FIFO is the default queuing algorithm in some instances, thus requiring no configuration, but it

has several shortcomings. Most importantly, FIFO queuing makes no decision about packet priority; the order of arrival determines bandwidth, promptness, and buffer allocation. Nor does it provide protection against ill-behaved applications.

Sensitive application traffic cannot be delivering in real time ,FIFO was the first method in controlling network traffic , today there are another queuing algorithms to avoid the shortcomings of FIFO queuing.

PQ: Priority Queueing

Prioritizing traffic which means that important traffic can be handled first , it can allow you to ensure that important traffic, application, and users take precedence .We have four queues high, medium, normal or low, each packet is placed in one of four queues based on assigned priority . During transmission Real-time priority is typically used for applications that are particularly sensitive to latency, such as voice and video applications , Packets in the outgoing traffic flow are queued based on their priority until the network is ready to process the packets.

Priority queuing is used to ensure that important traffic gets priority treatment . PQ currently use static configuration and thus does not automatically adapt to changing network requirements.

CQ: Custom Queuing

Guaranteeing bandwidth was designed to allow various applications or organizations to share the network among applications with specific minimum bandwidth or latency requirements. In these environments, bandwidth must be shared proportionally between applications and users. we use it to provide guaranteed bandwidth at a potential congestion point, ensuring the specified traffic a fixed portion of available bandwidth and leaving the remaining bandwidth to other traffic. Custom queuing handles traffic by assigning a specified amount of queue space to each class of packets and then servicing the queues in a round-robin fashion.

WFQ: flow-based Queuing

Flow-based queuing algorithm used in Quality of Service (QoS) that does two things simultaneously: It schedules interactive traffic to the front of the queue to reduce response time, and it fairly shares the remaining bandwidth between high bandwidth flows.

WFQ is a flow-based method that sends packets over the network and ensures packet transmission efficiency which is critical to the interactive traffic. This method automatically stabilizes network congestion between individual packet transmission flows.

The WFQ method has the advantage of being fast, reliable and easy to implement. WFQ follows these main criteria:

- Dedicated queues for each flow (referred to as conversations), messages are sorted into conversations reducing starvation, delay, and jitter within the queue.

- Allocating bandwidth fairly and accurately among all flows, reducing scheduling delay and guaranteeing service.

- IP Precedence is used as weight when allocating bandwidth.

WFQ supports flows with different bandwidth requirements by giving each queue a weight that assigns it a different percentage of output port bandwidth. WFQ also supports variable-length packets, so that flows with larger packets are not allocated more bandwidth than flows with smaller packets.

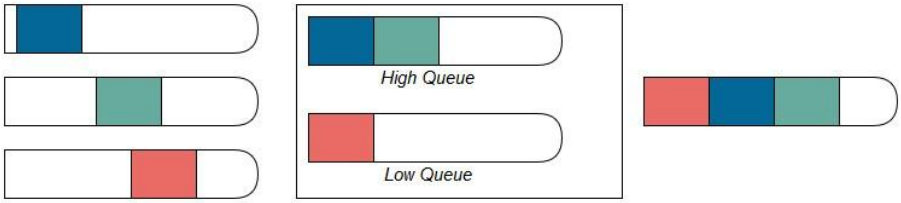

Figure 4.4: Flow-Based Queuing

4.3.3 Congestion Avoidance

WRED: Weighted Random Early Detection

Queuing mechanism is about managing the tail of out queue. When the queue is fills up all new packets arriving will be dropped that called tail dropped, Data traffic is usually bursty so when tail drop occurs, the router probably drops multiple packets. Tail drop is bad, especially for TCP

traffic, because when TCP segments are dropped, it reduces back to one segment also all the packet discarded may have been a high-priority packet and the router did not have chance to queue it .

So we can use a technique called RED (Random Early Detection) Instead of waiting for tail drop to happen, we monitor the queue depth. When the queue starts to fill up, we discard some random packets with the goal of slowing down TCP.

The "weighted" part of WRED is that WRED monitors the average queue depth. When the queue starts to fill, it will only drop a few random packets. When the queue length increases, it becomes more aggressive and drops even more random packets until it hits a certain limit. When this limit is reached, all packets are dropped.

In this graph we have average queue depth (the length of the queue), and we have the discard probably, so when the average queue depth is below the minimum threshols (20), WRED does not drop any packet at all , when the average queue is above the minimum threshold WRED stars to drop a small number of random packets, when the average queue depth increases even further, WRED drops a larger percent of random packets until we reach the maximum threshold (45), When the average queue depth reaches the maximum threshold (45), WRED drops all packets. When the average queue depth reaches the maximum threshold (45), WRED drops all packets [2].

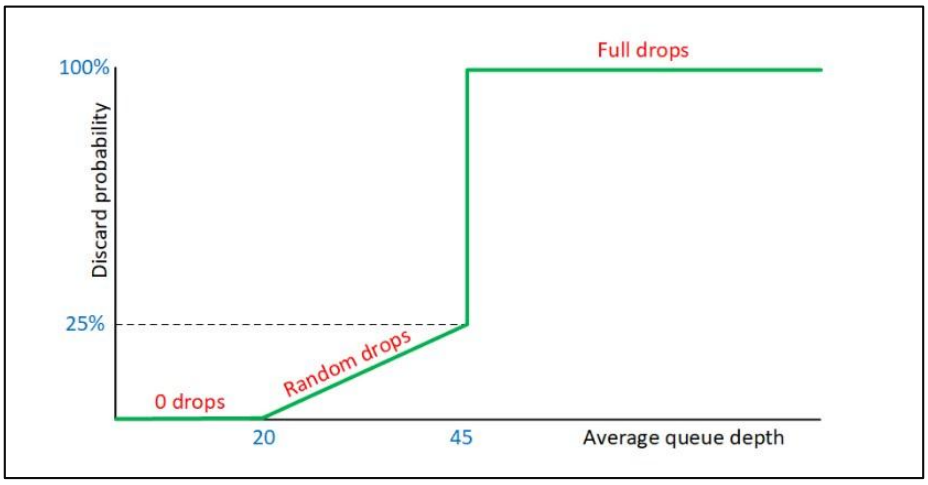

Figure 4.5: WRED Graph

4.3.4 Shaping and Policing

Shaping is a QoS technique that we can use to enforce lower bitrates than what the physical interface is capable of. Most ISPs use shaping or policing to enforce “traffic contracts” with their customers. When we use shaping we will buffer the traffic to a certain bitrate, policing will drop the traffic when it exceeds a certain bitrate.

There are two reasons why you might want to configure shaping:

- Instead of waiting for the policer of the ISP to drop your traffic, you might want to shape your outgoing traffic towards the ISP so that they don’t drop it.

- To prevent egress blocking. When you go from a high speed interface to a low speed interface you might get packet loss (tail drop) in your outgoing queue. We can use shaping to make sure everything will be sent (until its buffer is full).

We use policing to specify that a class of traffic should have a maximum rate imposed on it and if that rate is exceeded, an immediate action must be taken. In other words, with the police command, it is not possible to buffer the packet and send it out later, as is the case for the shape command.

In addition, with policing, the token bucket determines whether the packet exceeds or conforms to the applied rate. In either case, policing implements a configurable action, which includes setting the IP precedence or Differentiated Services Code Point (DSCP).

The two methods Commited Access Rate (CAR) and Class-based Policing (PBR) are used for traffic policing , and the two mechanisms have important functional difference as explained in classification section .

4.4 MPLS DiffServ Tunneling Modes

Since mpls labels contain three experimental bits that are commonly used for qos marking, it is possible to use “tunnel DiffServ” that is, preserve Layer 3 DiffServ markings through a SP’s MPLS VPN cloud while still performing re-marking (via MPLS EXP bits) within the cloud to indicate in- or out-of-contract traffic, there are three distinct modes of MPLS DiffServ tunnelling.

4.4.1 Uniform Model

In the Uniform model, The LSP DiffServ Information must be derived from the tunneled Information on the ingress LSR, on the engress the LSP DiffServ information must be distributed to the tunneled diffserv information .

The QoS information is always present in the topmost label or in the IP header if the packet is not labeled. The MPLS network does not have an impact on the QoS information, but it does switch the packets through the MPLS network.

To change the EXP bits of the top label(s) we use the MQC, This only changes the outer QoS information, or the LSP DiffServ information , and this change in the LSP DiffServ information is propagated on the engress LSR because the EXP bits are mapped to IP Precedence values [2].

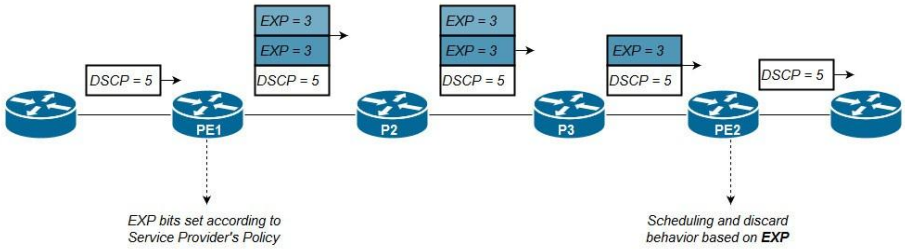

Figure 4.6: uniform model

4.4.2 Pipe Model

In the pipe model the LSP DiffServ is not derived from the Tunneled DiffServ Information on the ingress LSR .on the engress LSR the forwarding treatment (classifying the packet for scheduling and discarding behavior at the output interface) of the packet is based on MPLS PHB (EXP bits), and the LSP DiffServ Information is not propagated to the tunneled DifferServ Information [2].

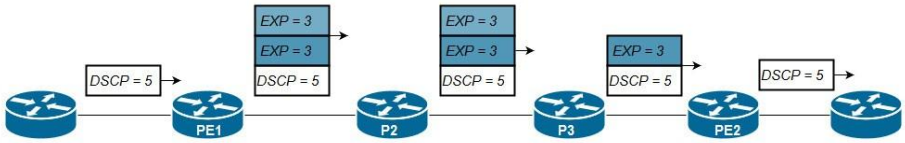

Figure 4.7: pipe model

4.4.3 Short Pipe Model

The Short Pipe model is similar to the Pipe model, with one difference. The forwarding treatment on the egress LSR is different from the Short Pipe model. On the egress LSR, the forwarding treatment from the packet is based on the Tunneled DiffServ information, and the LSP DiffServ information is not propagated to the Tunneled DiffServ information [2].

If the MPLS network is receiving IP packets on the ingress LSR , On the egress LSR, the forwarding treatment of the packet is based on the IP PHB (IP precedence), and the EXP bits are not propagated to the IP precedence.

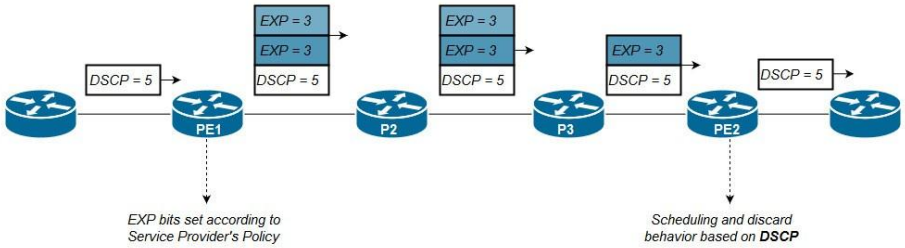

Figure 4.8: short pipe model

4.5 Conclusion

The usage of quality of service (QoS) has become widespread, MPLS network use QoS at some routers, it can be implemented on three MPLS DifferServ Models, all the three models are distinct and have their own merits. The distinction between the three models is only at the edge LSRs.

Chapter 5

MPLS Applications Deployment

In this chapter of thesis, we will deploy the previously seen MPLS applications, MPLS VPN, MPLS TE and MPLS QoS, virtually on GNS3 the network software emulator, under Cisco vendor. Each lab will be detailed on both the configuration and verification levels in the following pages; we tried to cover all the points covered in the previous chapters in order to render our knowledge more tangible. Each lab will have few pertinent WireShark captures for demonstration purposes; the full captures are available on demand.

5.1 The work environment

5.1.1 Choice of software

Before starting the realization of our project, we must choose the necessary tools to implement it. For this we have chosen to work with:GNS3 and Wireshark.

GNS3: Gns3 Graphical Network Emulator is an open source program that simulates complex networks by being as similar as possible to the way real networks operate. All of this without requiring network's equipment, such as routers and switches.

This software offers an excellent graphical user interface for the design and configuration of virtual networks, runs on standard computer hardware and can be used on various operating systems like Windows , Linux,and MacOS X.

In order to provide complete and precise simulations, gns3 actually uses the following emulators to run the same operating systems as in real networks:

Dynamips: the well known Cisco IOS emulator.

VirtualBox: runs desktop and server operating systems as well as Juniper JunOS.

Qemu: a generic open source machine emulator, it runs Cisco ASA, PIX and IPS.[12]

Figure 5.1: GNS3 Emulator

Wireshark is a network packet analyzer shows the captured packet data as detailed as possible

You can think of a network packet analyzer as a measurement tool to check what's going on within a network cable as an electrician uses a voltmeter, for examining what's happening inside an electric cable (but at a higher level, of course).

In the past, such tools were either very expensive, proprietary, or both. Although, this has improved with the introduction of wireshark. Wireshark is an open source, and is one of the best packet analyzers available today.[13]

Figure 5.2 : Wireshark network protocol analyzer

5.1.2 Choice of material

The materials we used are:

Ethernet crossover cable or serial cable are used to interconnect cisco routers.

Figure 5.3: Crossover cable

Figure 5.4: Serial cable

Cisco C7200 Router : We used a Cisco 7200 router since it supports MPLS, VPN, TE, QOS.

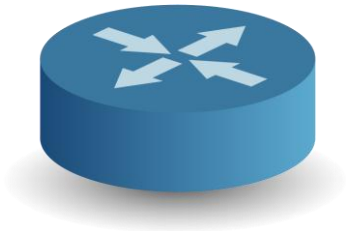

Figure 5.5: Cisco C7200 router

5.2 MPLS VPN Lab

In this Lab, we have a Service Provider providing WAN connectivity through MPLS VPN to two customers, who have two remote sites each. The customers also have connectivity from some of their sites (Customer-1 CE1 and Customer2 CE2) to a centralized server, we will see a configuration walkthrough in order to implement the following design.

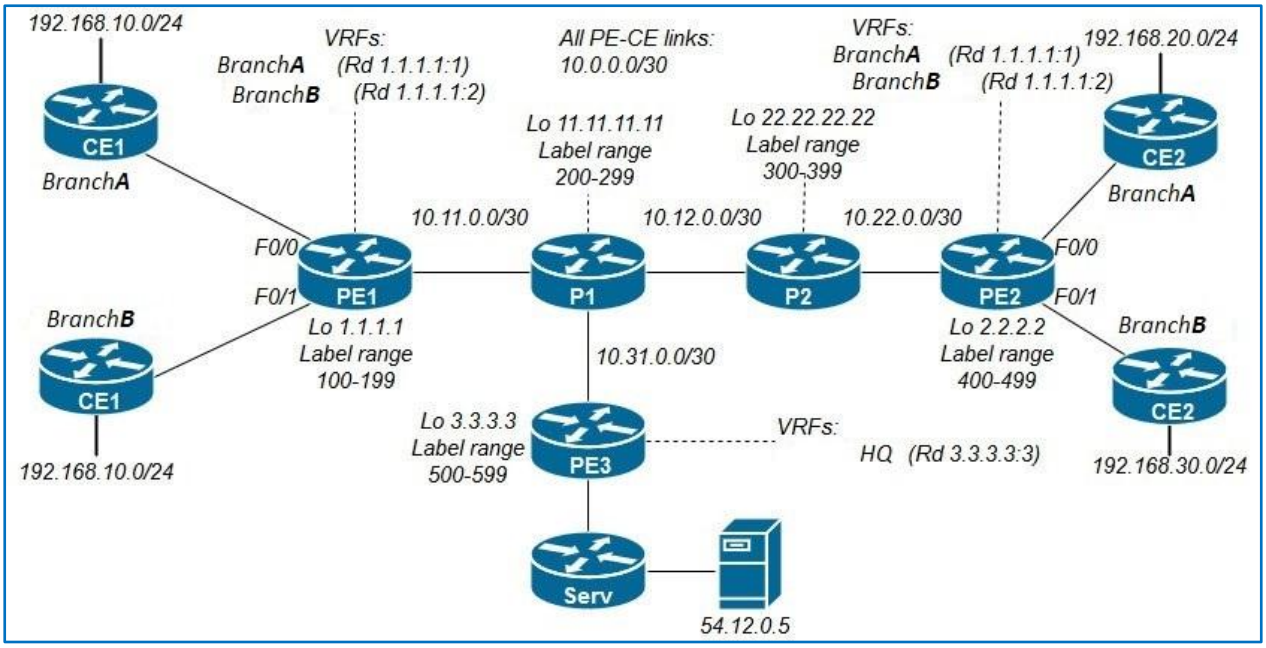

Figure 5.6: MPLS VPN Topology

5.2.1 MPLS Configuration

We first need to establish full IPv4 connectivity in the Service Provider MPLS network, and since we will use the loopback addresses on each MPLS aware router as LDP router-id, we need to advertise the loopbacks' addresses. In this lab we will be using OSPF protocol as the SP IGP. Next, we need to enable MPLS on each router and on each link in the network and assign a label range for LDP to make verification easier.

Router1 MPLS Config:

```
interface Loopback0
ip address 1.1.1.1 255.255.255.255
!
router ospf 1
network 1.1.1.1 0.0.0.0 area 0
network 10.11.0.0 0.0.0.255 area 0 !
mpls ip
mpls label range 100 199
!
interface FastEthernet1/0
ip address 10.11.0.1 255.255.255.0
mpls ip
```

5.2.2 MPLS Verification

We can verify that the routers have created local labels for each FIB entry and advertised them, by showing the MPLS LDP bindings on Router P1.

```
R2-P1#show mpls ldp bindings
  tib entry: 1.1.1.1/32, rev 2
        local binding:  tag: 200
        remote binding: tsr: 1.1.1.1:0, tag: imp-null
        remote binding: tsr: 22.22.22.22:0, tag: 300
        remote binding: tsr: 3.3.3.3:0, tag: 500
  tib entry: 2.2.2.2/32, rev 4
        local binding:  tag: 201
        remote binding: tsr: 1.1.1.1:0, tag: 100
        remote binding: tsr: 22.22.22.22:0, tag: 301
        remote binding: tsr: 3.3.3.3:0, tag: 501
  tib entry: 3.3.3.3/32, rev 6
        local binding:  tag: 202
        remote binding: tsr: 1.1.1.1:0, tag: 101
        remote binding: tsr: 22.22.22.22:0, tag: 302
        remote binding: tsr: 3.3.3.3:0, tag: imp-null
```

Figure 5.7: LDP Bindings

Data plane wise, Traceroute now shows the LSP taken next to the IP addresses when displaying paths, tracing 2.2.2.2 route on PE1 sourcing from its Loopback0 displays the following output.

```
R1-PE1#traceroute 2.2.2.2 source 1.1.1.1

Type escape sequence to abort.
Tracing the route to 2.2.2.2

  1 10.11.0.11 [MPLS: Label 201 Exp 0] 64 msec 84 msec 64 msec
  2 10.12.0.22 [MPLS: Label 301 Exp 0] 52 msec 36 msec 56 msec
  3 10.22.0.2 44 msec 52 msec 52 msec
```

Figure 5.8: Traceroute Output

5.2.3 PE-PE Configuration

First, we need to configure an iBGP instance between PE routers sourcing from their respective loopbacks in order to exchange IPV4 prefixes.

Router1 iBGP Config

router bgp 100

bgp log-neighbor-changes

neighbor 2.2.2.2 remote-as 100

neighbor 2.2.2.2 update-source Loopback0
neighbor 3.3.3.3 remote-as 100
neighbor 3.3.3.3 update-source Loopback0
!

This will result in an iBGP dual adjacency between PE provided they have the same configuration.

Now we need to instruct PE routers to support VPNv4 address families, which is a BGP address family, in order to support RD. Route targets are part of the VPNv4 addresses, and as we've seen in chapter 2, they're carried as extended communities in the MPBGP VPNv4 updates, and they must be enabled manually.

Router1 MPBGP Config

address-family vpnv4
neighbor 2.2.2.2 activate
neighbor 2.2.2.2 send-community extended
neighbor 3.3.3.3 activate
neighbor 3.3.3.3 send-community extended
exit-address-family
!

5.2.4 PE-PE Verification

We can verify that PE routers now advertise and receive both IPv4 and VPNv4 updates.

```
R1-PE1#show ip bgp neighbors | section capabilities
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
```

Figure 5.9: BGP Adjacency

5.2.5 PE-CE Configuration

In order to configure PE-CE connections, we need to configure the PE routers first following the MPBGP instances (VRF, RD and RT values) shown in the topology figure.
First, we must create two distinct VRFs for each customer on both PE routers, then configure the RD and RT values.

PE1 VRF BranchA Config

```
ip vrf BranchA
rd 1.1.1.1:1
route-target export 1.1.1.1:10
route-target import 2.2.2.2:10
```

PE1 VRF BranchB Config

```
ip vrf BranchB
rd 1.1.1.1:2
route-target export 1.1.1.1:20
route-target import 2.2.2.2:20
```

Now we need to assign the interfaces connected to the sites to their VRF.

PE1 VRF Interface Config

```
interface FastEthernet0/1
description Connected to BranchB site 1
mac-address 0000.1111.2222
ip vrf forwarding BranchB
!
interface FastEthernet0/0
description Connected to BranchA site 1
mac-address 0000.1111.1111
ip vrf forwarding BranchA
```

Note that assigning an interface to a VRF wipes the configured IP address, therefore we need to re-configure it. We will be using OSPF as the IGP between the PE and CE routers in order to advertise routes between CE and PE routers.

PE1 Per-VRF OSPF Config

```
router ospf 100 vrf BranchA
network 10.0.10.0 0.0.0.3 area 0
!
router ospf 200 vrf BranchB
network 10.0.20.0 0.0.0.3 area 0
```

The OSPF adjacency will come up after the conventional OSPF configuration on the CE routers, we will create loopback interfaces on the CE sites to simulate network branches.

CE1 BranchA OSPF Config

```
interface Loopback0
ip address 192.168.10.1 255.255.255.0
!
router ospf 100
network 10.0.10.0 0.0.0.3 area 0
network 192.168.10.0 0.0.0.255 area 0
```

In order to verify the adjacency, we should consult PE1 VRFS, for instance, we can see that PE2 BranchA VRF has learned about the 192.168.20.0/24 route via OSPF.

```
192.168.20.0/32 is subnetted, 1 subnets
  192.168.20.1 [110/2] via 10.0.30.2, 00:00:48, FastEthernet0/0
```

Figure 5.10: PE2 BranchA-VRF Routing Table Entry

The last step is to export routes from the VRF routing tables into the MPBGP and the other way around, meaning, exporting routes from MPBGP to

the VRF, this operation is called route redistribution.

PE1 OSPF to BGP redistribution Config

```
router ospf 100 vrf BranchA
redistribute bgp 100 subnets
network 10.0.10.0 0.0.0.3 area 0
!
router ospf 200 vrf BranchB
redistribute bgp 100 subnets
network 10.0.20.0 0.0.0.3 area
```

PE1 BGP to OSPF redistribution Config

```
address-family ipv4 vrf BranchA
redistribute ospf 100 vrf BranchA
no synchronization
exit-address-family
!
address-family ipv4 vrf BranchB
redistribute ospf 100 vrf BranchB
no synchronization
exit-address-family
```

Now in order to verify, we should consult the routing table on the Site1 for each customer to confirm that they have received the OSPF updates from the Site2 branches.

```
Kingslanding-Site1#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

     192.168.30.0/32 is subnetted, 1 subnets
O IA    192.168.30.1 [110/3] via 10.0.20.1, 00:04:45, FastEthernet0/1
C    192.168.10.0/24 is directly connected, Loopback0
     10.0.0.0/30 is subnetted, 2 subnets
C       10.0.20.0 is directly connected, FastEthernet0/1
O IA    10.0.40.0 [110/2] via 10.0.20.1, 00:04:45, FastEthernet0/1
```

Figure 5.11: BranchB Site2 Routing Table

```
Winterfell-Site1#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    192.168.10.0/24 is directly connected, Loopback0
     192.168.20.0/32 is subnetted, 1 subnets
O IA    192.168.20.1 [110/3] via 10.0.10.1, 00:02:18, FastEthernet0/0
     10.0.0.0/30 is subnetted, 2 subnets
C       10.0.10.0 is directly connected, FastEthernet0/0
O IA    10.0.30.0 [110/2] via 10.0.10.1, 00:02:18, FastEthernet0/0
```

Figure 5.12: BranchA Site2 Routing Table

We can also verify by consulting the VRF routing tables on PE routers, for instance, we have the BranchA VRF routing table on PE1 shown below

```
R1-PE1#show ip route vrf Winterfell

Routing Table: Winterfell
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

     192.168.20.0/32 is subnetted, 1 subnets
B       192.168.20.1 [200/2] via 2.2.2.2, 00:03:35
     10.0.0.0/30 is subnetted, 2 subnets
C       10.0.10.0 is directly connected, FastEthernet0/0
B       10.0.30.0 [200/0] via 2.2.2.2, 00:03:35
```

Figure 5.13: BranchA-VRF Routing Table on PE1

However, note that the advertised routes are learned via BGP with the advertising PE loopback as next-hop and route redistribution only happens between the PE VRF and the CE, which will eventually get the routes as OSPF-learned routes.

5.2.6 Adding Centralized Servers

Now we will add a site that is connected to PE3 that will be accessible by only BranchA Site 1 and BranchB Site 2. First, we will configure another VRF on PE3 that will have two route target export addresses: 2.2.2.2:10 and 1.1.1.1:20 which are the Route Target Import addresses for BranchA site1 and BranchB Site 2. Alongside a different Route Distinguisher: 3.3.3.3:3.

PE3 VRF Config

```
ip vrf HQ
rd 3.3.3.3:3
route-target export 2.2.2.2:10
route-target export 1.1.1.1:20
```

Now we will configure the IGP, per-VRF OSPF in this case, between PE3 and Serv site.

PE3 Per-VRF OSPF Config

```
router ospf 100 vrf HQ
log-adjacency-changes
redistribute bgp 100 subnets
network 10.0.10.0 0.0.0.3 area 0
```

The adjacency between PE3 and Serv site is expected to come up once we assign the PE3 interface to HQ VRF and OSPF configuration on Serv site.

PE3 iBGP Peering with PE1 and PE2 Config

```
router bgp 100 bgp log-neighbor-changes
neighbor 1.1.1.1 remote-as 100
```

neighbor 1.1.1.1 update-source Loopback0

neighbor 2.2.2.2 remote-as 100

neighbor 2.2.2.2 update-source Loopback0

!

address-family vpnv4 neighbor 1.1.1.1

activate neighbor 1.1.1.1 send-community extended

neighbor 2.2.2.2 activate

neighbor 2.2.2.2 send-community extended

exit-address-family

!

Now, VRF BranchA on PE1 and VRF BranchB on PE2 should have the route updates about Serv site connected Server with the address of 54.12.0.5 For verification purposes, we can check the routing tables of BranchA-Site1 and BranchB-Site2, to find a route leading to 54.12.0.5/32 learned via Inter Area OSPF.

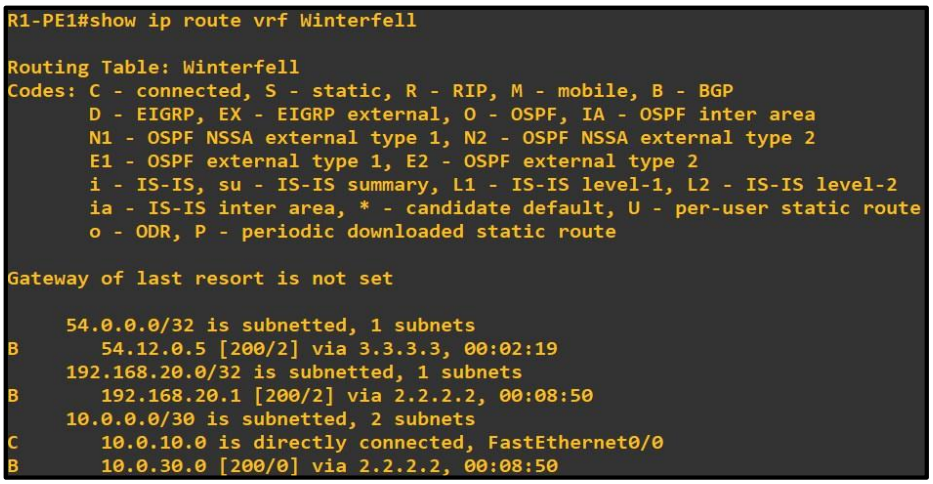

Figure 5.14: BranchA-VRF Routing Table on PE1

5.2.7 Wireshark Captures

MPBGP Peering

Upon configuring a BGP adjacency, PE1 and PE2 exchange OPEN messages, this OPEN Message was sent from PE2 to PE1.

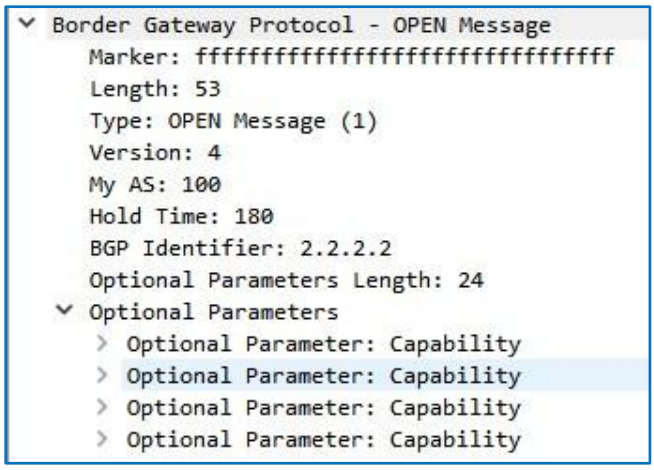

Figure 5.15: BGP OPEN Message

MPBGP Update

We captured this BGP update message on the link between PE1 and P1.

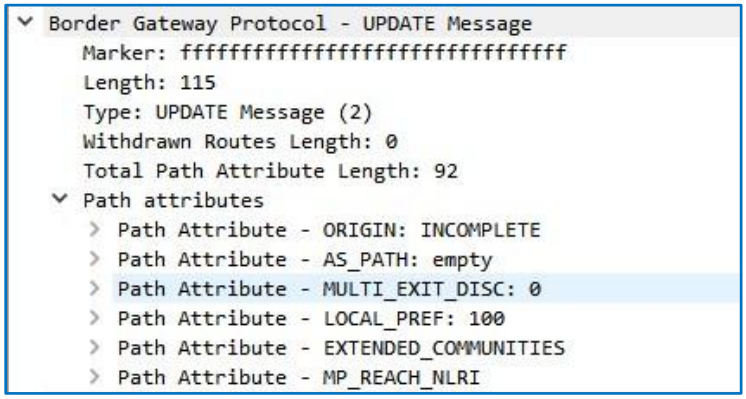

Figure 5.16: BGP Update

It contains the extended communities carried as a Path Attribute.

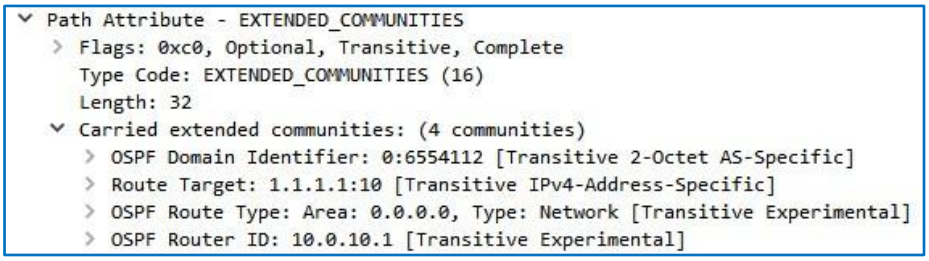

Figure 5.17: BGP Extended Communities

It also has the NLRI information about the exported routes.

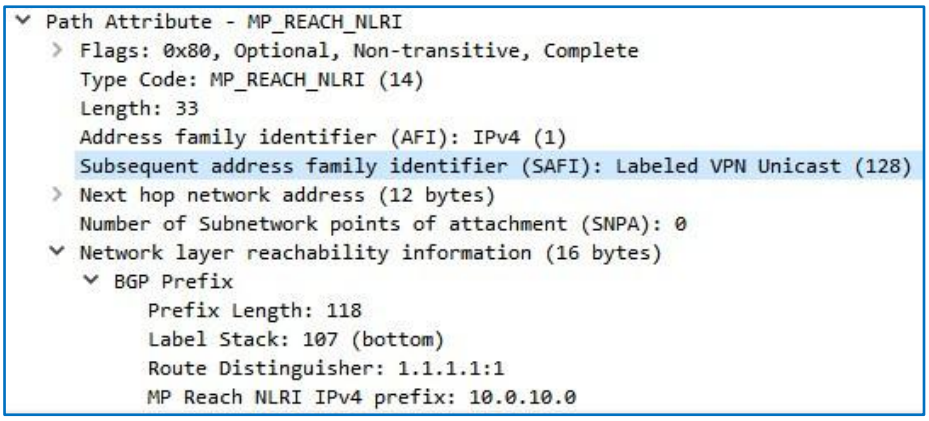

Figure 5.18: NLRI Information

LDP

Both PE1 and P routers end LDP hello messages periodically to 224.0.0.2, in order to form an adjacency.

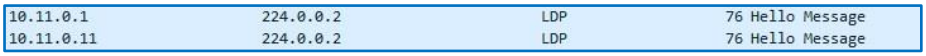

Figure 5.19: LDP Hello Messages

After initializing an LDP session, PE1 and P1 exchange labels.

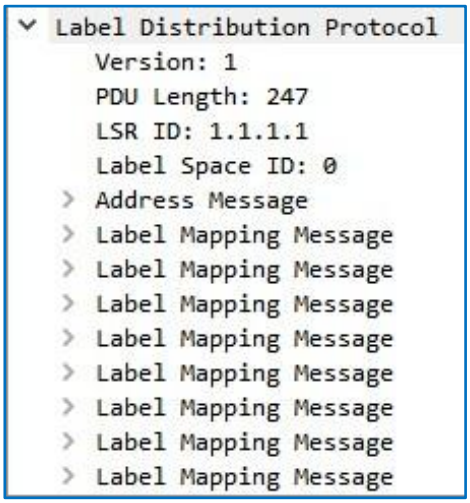

Figure 5.20: LDP Label Mapping

CE-CE Ping

We tried to ping BranchA second site from the first site, these two ICMP requests were captured between PE1 and P1 and P1 and P2 respectively, we can see that the inner label is reserved for BranchA (Label 408) and that the outer label is LDP-advertised and is used to forward traffic from the PE1 to PE2.

```
MultiProtocol Label Switching Header, Label: 304, Exp: 0, S: 0, TTL: 253
    0000 0000 0001 0011 0000 .... .... .... = MPLS Label: 304
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1101 = MPLS TTL: 253
MultiProtocol Label Switching Header, Label: 408, Exp: 0, S: 1, TTL: 254
    0000 0000 0001 1001 1000 .... .... .... = MPLS Label: 408
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
```

Figure 5.21: CE-CE ICMP Request on PE1-P1 Link

```
MultiProtocol Label Switching Header, Label: 204, Exp: 0, S: 0, TTL: 254
    0000 0000 0000 1100 1100 .... .... .... = MPLS Label: 204
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
MultiProtocol Label Switching Header, Label: 408, Exp: 0, S: 1, TTL: 254
    0000 0000 0001 1001 1000 .... .... .... = MPLS Label: 408
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
```

Figure 5.22: CE-CE ICMP Request on P1-P2 Link

5.3 MPLS TE Lab

In this lab, we will configure two TE tunnels linking the edge routers of the service provider MPLS network, one tunnel will require a minimum bandwidth of 1800 kbps and will have a dynamic path and the other will require 512 kbps and will have an explicit path-option and a second dynamic path-option for backup.

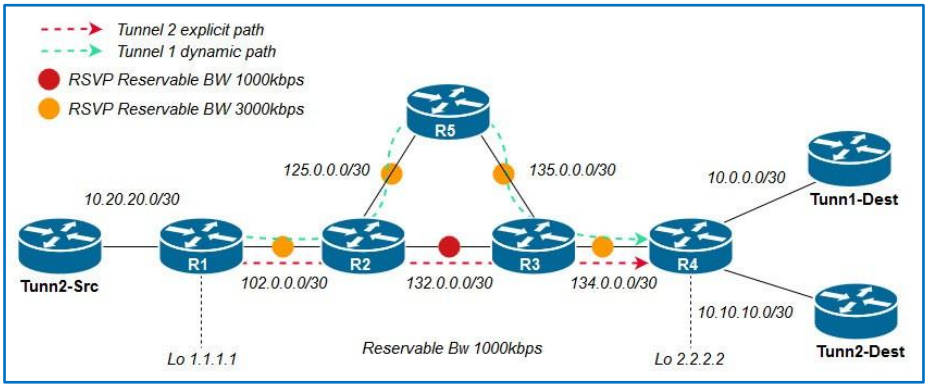

Figure 5.23: MPLS TE Topology

5.3.1 MPLS Configuration

We first need to establish full IPv4 connectivity in the Service Provider MPLS network, and since we will use the loopback addresses on each MPLS aware router as LDP router-id, we need to advertise the loopbacks' addresses. In this lab we will be using OSPF protocol as the SP IGP.Next, we need to enable MPLS on each router and on each link in the network and assign a label range for LDP to make verification easier.Note that although RSVP distributes labels, it is mandatory to use LDP here as tunnels 1 and 2 are unidirectional, we can eliminate the LDP use if we configure a second tunnel for each tunnel, that take the exact path, but in the opposite direction, making tunnels bidirectional.

Router1 MPLS Config

```
interface Loopback0
ip address 1.1.1.1 255.255.255.255 !
router ospf 100
network 1.1.1.1 0.0.0.0 area 0
network 10.20.20.0 0.0.0.3 area 0
network 102.0.0.0 0.0.0.3 area 0
!
mpls ip
mpls label range 100 199
!
interface FastEthernet1/0
ip address 102.0.0.1 255.255.255.252
mpls ip
```

5.3.2 MPLS TE Configuration

Now that we have MPLS enabled we should enable the TE feature on the routers and their interfaces.

Router1 TE Config

```
mpls traffic-eng tunnels
    !
interface FastEthernet0/0
mpls traffic-eng tunnels
```

We must specify the maximum reservable bandwidth by RSVP on each interface, we'll limit it to 3000 kbps on every interface except for the link between R2 and R3 that will be set to 1000 kbps only.

Router1 Bandwidth Config

```
interface FastEthernet0/0
ip rsvp bandwidth 3000
```

Router2 Bandwidth Config

```
interface FastEthernet0/1
ip rsvp bandwidth 1000
```

Now we should enable MPLS TE for OSPF on Area 0 and specify the traffic engineering router-id for the node to be the IP address associated with interface loopback0.

Router1 OSFP-TE Config

```
router ospf 100
mpls traffic-eng router-id Loopback0
mpls traffic-eng area 0
```

5.3.3 Tunnels Configuration

As previously mentioned, we will configure two tunnels, where Tunnel1 will use a dynamic path and Tunnel2 will have an explicit path. Tunnel1 will require a 1800 Kbps on each link, it will be destined to 4.4.4.4 (Router4 as a tail-end) and sourced from 1.1.1.1(Router1 as a head-end) and it

will have a 0 hold priority and a 1 setup priority. It will have an unnumbered IP address because tunnels are unidirectional.

Router1 Tunnel1 Config

Interface Tunnel1 ip unnumbered Loopback0
tunnel destination 4.4.4.4
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 1 0
tunnel mpls traffic-eng bandwidth 1800
tunnel mpls traffic-eng path-option 1 dynamic

we should configure Tunnel2, that will have an explicit path as its first pathoption and a dynamic path as its second, it will be sourced from Rrouter 1 to Router4 and it will require 512 Kbps, with a 2 hold priority and a 3 setup priority.

First, we need to configure the explicit path from the head end to the tail end with the R1-R2-R3-R4 LSP, which will go through these outgoing inter-faces:102.0.0.1, 132.0.0.2 and 134.0.0.1

Router1 Explicit path Config

ip explicit-path identifier 1 enable
next-address 102.0.0.1
next-address 132.0.0.2
next-address 134.0.0.1

Router1 Tunnel2 Config

interface Tunnel2
ip unnumbered Loopback0
tunnel destination 4.4.4.4
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 3 2

tunnel mpls traffic-eng bandwidth 512

tunnel mpls traffic-eng path-option 1 explicit identifier 1

tunnel mpls traffic-eng path-option 2 dynamic

5.3.4 Tunnels Verification

To check the tunnels status, we can check this exempt from the show MPLS traffic-eng tunnels command, to see that they are both operational.

```
R1#show mpls traffic-eng tunnels tunnel 1

Name: R1_t1                               (Tunnel1) Destination: 4.4.4.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
```

Figure 5.24: Tunnel1 Status

```
R1#show mpls traffic-eng tunnels tunnel 2

Name: R1_t2                               (Tunnel2) Destination: 4.4.4.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
```

Figure 5.25: Tunnel2 Status

```
R1#show mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:            running
    RSVP Process:                   running
    Forwarding:                     enabled
    Periodic reoptimization:        every 3600 seconds, next in 3351 seconds
    Periodic auto-bw collection:    disabled
TUNNEL NAME                 DESTINATION      UP IF     DOWN IF   STATE/PROT
R1_t1                       4.4.4.4          -         Fa0/0     up/up
R1_t2                       4.4.4.4          -         Fa0/0     up/up
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

Figure 5.26: TE Tunnels Status

And here we can check the LSP each tunnel upon the setup, we can see that Tunnel1 dynamic path chose the R1-R2-R5-R3-R4 LSP because it requires a minimum of 1800kbps bandwidth and R2-R3 link has only 1000kbps available for RSVP reservation.

```
Explicit Route: 102.0.0.2 125.0.0.2 125.0.0.1 135.0.0.1
                135.0.0.2 134.0.0.1 134.0.0.2 4.4.4.4
```

Figure 5.27: Tunnel1 Dynamic Path

```
Explicit Route: 102.0.0.2 132.0.0.2 132.0.0.1 134.0.0.1
                134.0.0.2 4.4.4.4
```

Figure 5.28: Tunnel2 Explicit Path

5.3.5 Traffic Forwarding

Now, we need to assign traffic to these tunnels. First, we will assign all traffic destined to 10.0.0.0/30 (link connecting Router4 to Tunn1-Dest) to go through Tunnel1 by using the static route method for traffic forwarding.

Router1 Tunnel1 Traffic Forwarding Config

ip route 10.0.0.0 255.255.255.252 Tunnel1

Here, using Traceroute command, we can verify Tunnel1 path.

```
R1#traceroute 10.0.0.1

Type escape sequence to abort.
Tracing the route to 10.0.0.1

  1 102.0.0.2 [MPLS: Label 206 Exp 0] 76 msec 40 msec 96 msec
  2 125.0.0.1 [MPLS: Label 507 Exp 0] 80 msec 24 msec 88 msec
  3 135.0.0.2 [MPLS: Label 306 Exp 0] 40 msec 80 msec 64 msec
  4 134.0.0.2 60 msec 68 msec 52 msec
R1#ping 10.0.0.1
```

Figure 5.29: Traceroute Output

We will use Tunnel2 for traffic destined to 10.10.10.0/30. We can achieve this by using a different method for traffic forwarding, which is Policy-based routing. First, we need to configure the access-list that will define all traffic destined to 10.10.10.0/30. Then we need to configure the route-map with a match clause to allow it to use the ACL entry and a set clause to instruct it to forward traffic through the Tunnel2 interface. Finally, we need to assign the policy route-map to the F0/1 interface on Router1 (connected to Tunn2-Src) to forward incoming traffic from the Tunn2-Src router into the Tunnel2.

Router1 PBR Config

access-list 101 permit ip any 10.10.10.0

0.0.0.3

!

route-map tun2traffic permit 10

match ip address 101

set interface Tunnel2

!

interface FastEthernet0/1 ip address 10.20.20.1 255.255.255.252

ip policy route-map tun2traffic

5.3.6 Traffic Forwarding Verfication

Now we can verify that Tunn2-Src is using Tunnel2 first explicit-path to forward traffic to 10.10.10.0/24.

```
Tunn2-Src#traceroute 10.10.10.1

Type escape sequence to abort.
Tracing the route to 10.10.10.1

  1 10.20.20.1 8 msec 12 msec 8 msec
  2 102.0.0.2 [MPLS: Label 204 Exp 0] 120 msec 92 msec 88 msec
  3 132.0.0.1 [MPLS: Label 304 Exp 0] 80 msec 84 msec 84 msec
  4 134.0.0.2 84 msec 80 msec 84 msec
```

Figure 5.30: Traceroute Output

We can verify that Tunnel2 shifts to its second dynamic path-option by meddling in the explicit LSP, we had the following outputs when we stopped the link between R2 and R3

```
Tunn2-Src#traceroute 10.10.10.1

Type escape sequence to abort.
Tracing the route to 10.10.10.1

  1 10.20.20.1 12 msec 16 msec 36 msec
  2 102.0.0.2 [MPLS: Label 200 Exp 0] 124 msec 104 msec 104 msec
  3 125.0.0.1 [MPLS: Label 510 Exp 0] 120 msec 96 msec 108 msec
  4 135.0.0.2 [MPLS: Label 310 Exp 0] 104 msec 108 msec 104 msec
  5 134.0.0.2 100 msec 108 msec 92 msec
```

Figure 5.31: Traceroute Output

```
History:
  Tunnel:
    Time since created: 13 minutes, 26 seconds
    Time since path change: 2 seconds
  Current LSP:
    Uptime: 2 seconds
    Selection: reoptimation
```

Figure 5.32: Tunnel2 Path Change History

5.3.7 Wireshark Captures

OSPF-TE Update

We captured an OSPF update that contains 3 LSA10, used to flood the TE link information.

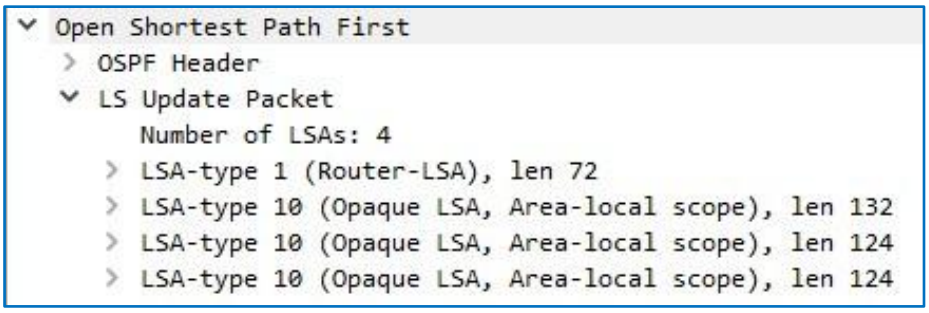

Figure 5.33: OSPF LSU

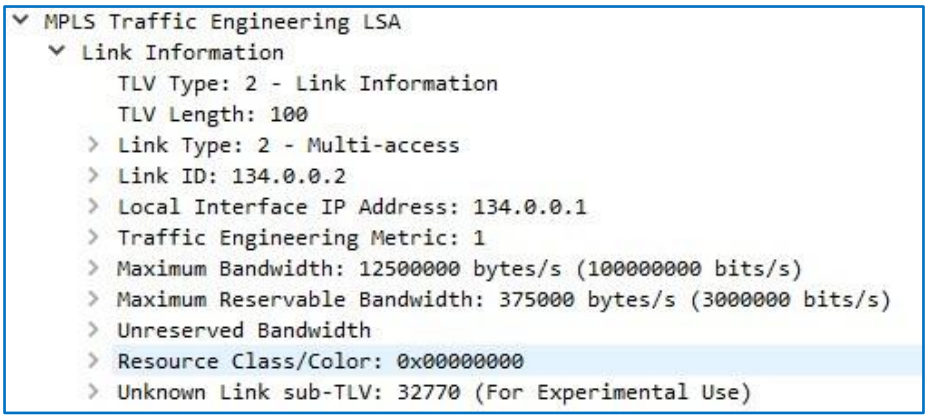

Figure 5.34: Link Information

RSVP PATH Message

We captured this RSVP PATH packet sent from R1 to R4 to establish Tunnel1 dynamic path on the link between R1 and R2.

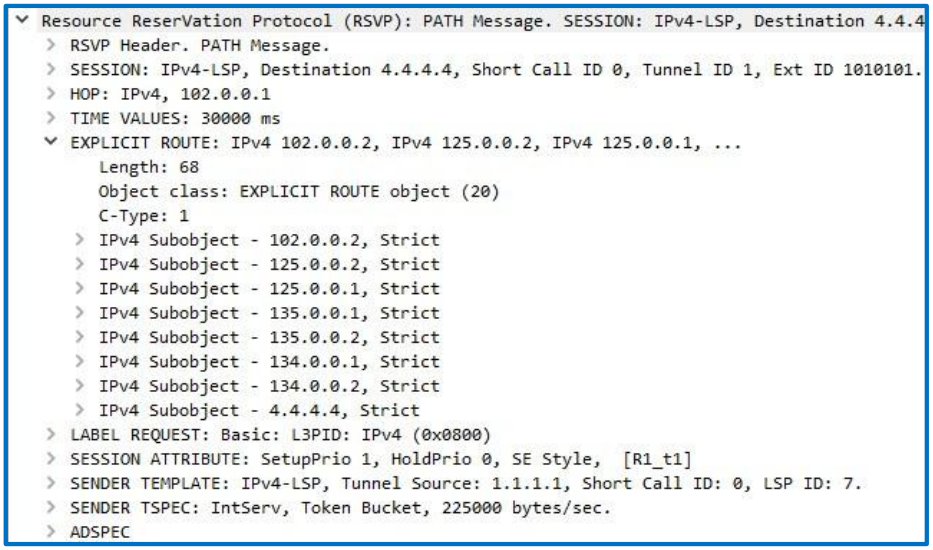

Figure 5.35: RSVP PATH Message

RSVP RESV Message

Here we can see the RESV message sent from R2 to R1, R2 reserved label 202 for tunnel2.

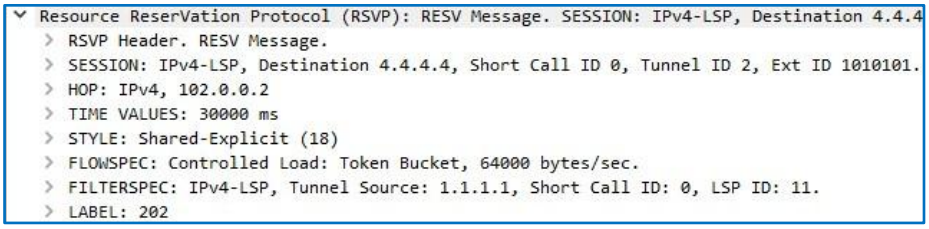

Figure 5.36: RSVP RESV Message

Tunn2-Src to Tunn2-Dest Ping

This ICMP request was captured after a ping from Tunn2-Src to Tunn2-Dest, on the link between R1 and R2, it was sent with a 202 label, that was advertised via RSVP from R2 to R1.

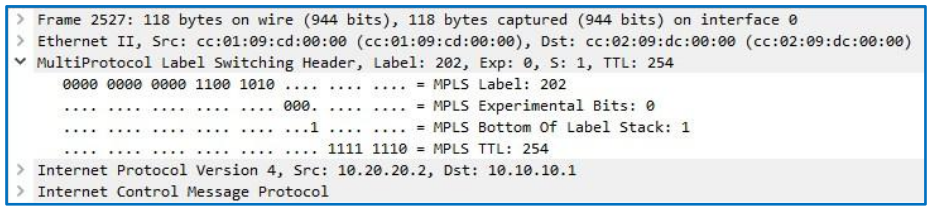

Figure 5.37: ICMP Request on R1-R2 Link

Tunn1-Dest Ping

This ICMP request was captured after a ping from R1 to Tunn1-Dest, on the link between R1 and R2, it was sent with a 203 label, that was advertised via RSVP from R2 to R1 for Tunnel1.

```
> Frame 2968: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
> Ethernet II, Src: cc:01:09:cd:00:00 (cc:01:09:cd:00:00), Dst: cc:02:09:dc:00:00 (cc:02:09:dc:00:00)
v MultiProtocol Label Switching Header, Label: 203, Exp: 0, S: 1, TTL: 255
    0000 0000 0000 1100 1011 .... .... .... = MPLS Label: 203
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
> Internet Protocol Version 4, Src: 1.1.1.1, Dst: 10.0.0.1
> Internet Control Message Protocol
```

Figure 5.38: ICMP Request on R1-R2 Link

5.4 MPLS QoS Lab

In this Lab, we will implement the uniform tunnelling model on the BranchAsite, the network consists of the BranchA router site 1 and site 2, the SP network we have PE1, PE2,P1 and P2. There are three simultaneous streams, each with IP Precedence values:

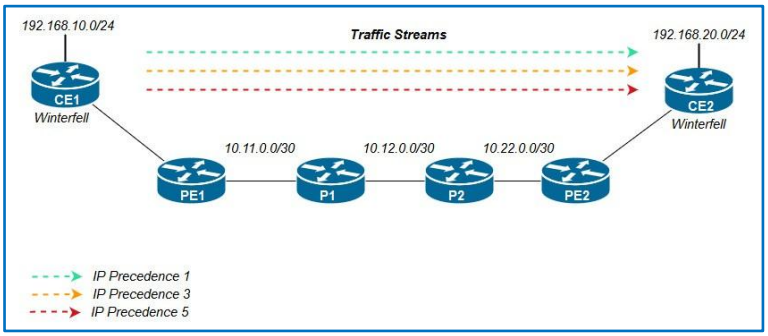

Figure 5.39: MPLS QoS Topology

5.5 MPLS QoS configuration

5.5.1 CE 1 configuration

We create a class-map for each traffic with different IP precedence value , the higher IP precedence value is given to the voice traffic , then the mission-critical traffic , and finally the Bulk-Data traffic :

BranchA1 Router class-map config

```
class-map precedence5
match ip precedence 5
class-map precedence3
match ip precedence 3
class-map precedence1
match ip precedence 1
```

And we create a policy map to specify the QoS settings applied to each class of traffic , the highest value of IP Precedence is the higher priority of ip packet which is the voice traffic :

BranchA 1 Router policy-map config

```
policy-map WinQos
class precedence5
priority
!
```

we specify the bandwidth applied for each class, and the queuing mechanism WRED to manage the tail to the queue .

Bandwidth not associated with any of the classes will be applied to the nonmarked IP Precedence traffic :

BranchA 1 router bandwidth config

```
class precedence3
bandwidth percent 30
random-detect
!
class precedence1
bandwidth percent 20
!
```

We apply the policy map on the interface f0/0, then the packets are forwarded to their attached PE routers :

BranchA 1 Router Service-Policy Config

interfacef0/0

service-policy output WinQos

5.5.2 IP to MPLS Domain Configuration

The ingress PE router copies the ip precedence value of customer packets that it receives from its connected BranchA router to the MPLS EXP setting of labels using the qos group value , it can be set in the ingress direction , and matched in the engress direction:

Router 1 class-map config

class-map match-all precedence5

match ip precedence 5

!

Router 1 policy-map config

policy-map mplsin

class precedence5

set qos-group 5

!

we apply the policy map on the interface f0/0 :

service-policy config

int f0/0

service-policy in mplsin

!

Router 1 class-map config

class-map qosgroup5

match qos-group 5

!

we use the policy-map to set the exp value of the label , and apply it in the engress direction :

router 1 policy-map config

policy-map mplsout

class qosgroup5

set mpls experimental topmost 5

priority 20000

police 100000 !

the same configuration to the other traffics with the change of the ip precedence value, then we apply the policy-map to the interface f1/0 :

router 1 service-police config

int f1/0

service-policy in mplsout

5.5.3 Pop label operation

The propagation of the Exp bits must be done on the P routers, in this case we have a pop label operation, we configure router 2 to rewrite the MPLS EXP bit to 1 for all traffic coming in with EXP value of 3. This is done using a class map matching all packets with MPLS EXP bit value of 3 And rewriting the same

using policy-map:

Router 2 Class-map config

class-map mplsexp3

match mpls experimental 3

!

Router2 policy-map config

policy-map mplsin

class mplsexp3

set qos-group 3

!

Router2 service-policy config

int f1/0

service-policy in mplsin

After the implementation of the service policy on the ingress interface, the QoS group is matched and is mapped to the topmost label EXP value on the engress labeled packet :

Router 2 class-map config

class-map qosgroup3

match qos-group 3

!

Router 2 policy-map config

policy-map mplsout

class qosgroup3

set mpls experimental topmost 1

!

Router 2 service-policy config

int f0/0

service-policy output mplsout

5.5.4 MPLS to IP domain configuration

In the PE2 router when the packets transits the MPLS domain into the IP domain the EXP value of the top-most label is propagated into the IP domain from the MPLS domain and is written as the IP Precedence value of the IP packet, we configure A class map to match all packets with MPLS EXP of 5 and 1 :

Router 5 class-map config

```
class-map match-all mplsexp5
match mpls experimental 5
!
class-map match-all mplsexp1
match mpls experimental 1 !
```

and corresponding policy map to configure qos-group value of the packet with the corresponding IP Precedence value :

Router 5 policy-map config

```
policy-map mplsin
class mplsexp5
set qos-group 5
!
class mplsexp1
set qos-group 1
!
```

Router 5 service-policy config

```
int f1/0
service-policy in mplsin
```

After the implementation of the service policy on the ingress interface, we create a class-map to match the qos group value, then a policy-map to mark the ip precedence value to the qos group value:

Router 5 class-map config

```
class-map qosgroup5
match qos-group 5
!
class-map qosgroup1
match qos-group 1
!
```

Router 5 policy-map config

```
policy-map mplsout class qosgroup5
set ip precedence 5
class qosgroup1
set ip precedence 1
!
```

Router 5 service-policy

```
int f0/0
service-poliy output mplsout
```

The number of packets matching the MPLS EXP value of 1 is twice the number of packets matching the MPLS EXP value of 5 due to the rewrite of EXP value performed at P1.

5.5.5 CE 2 Configuration

We apply a service policy on the ingress interface of the BranchA 2 router, and all the IP precedence value have been rewritten based on the MPLS EXP bit rewrite in the MPLS domain :

BranchA 2 Router class-map config

```
class-map match-all precedence1
match ip precedence 1
!
class-map match-all precedence5
match ip precedence 5
```

Here we use the MQC to create the policy-map, we police the traffic to 10 Mbps (configured in kbps) while allowing a committed burst of up to 1000000 bytes. so all traffic within the rate of 10Mbps+1Mbps takes the conforming condition which is configured to transmit the packet with the existing marking (if present). Any traffic above 10Mbps+Bc is considered to be exceeding the policy and dropped:

BranchA2 Router policy-map config

policy-map CEqos
class precedence5 police 10000000 1000000 conform-action
transmit exceed-action drop
class precedence1 police 10000000 1000000 conform-action
transmit exceed-action drop

finally we use the service-policy to attach the policy map to the interface :

BranchA2 Router service-policy config

int f0/0
service-policy input CEqos

5.6 QoS Verification

Let's check the QoS Configuration, we need a ping from site 1 to the second with a different TOS value corresponding to the IP precedence value 1,3 and 5.

```
Winterfell-Site1#ping
Protocol [ip]: ip
Target IP address: 192.168.20.1
Repeat count [5]: 5
Datagram size [100]: 100
Timeout in seconds [2]: 2
Extended commands [n]: y
Source address or interface: 10.0.10.2
Type of service [0]: 40
Set DF bit in IP header? [no]: yes
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
Packet sent with a source address of 10.0.10.2
Packet sent with the DF bit set
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 308/362/400 ms
```

Figure 5.40: Ping IP Precedence 1

We captured these packets ICMP sent from router 1 to 5 on the link between R1 and R2 :

```
MultiProtocol Label Switching Header, Label: 204, Exp: 1, S: 0, TTL: 254
    0000 0000 0000 1100 1100 .... .... .... = MPLS Label: 204
    .... .... .... .... .... 001. .... .... = MPLS Experimental Bits: 1
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
MultiProtocol Label Switching Header, Label: 408, Exp: 1, S: 1, TTL: 254
    0000 0000 0001 1001 1000 .... .... .... = MPLS Label: 408
    .... .... .... .... .... 001. .... .... = MPLS Experimental Bits: 1
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
Internet Protocol Version 4, Src: 10.0.10.2, Dst: 192.168.20.1
Internet Control Message Protocol
```

Figure 5.41: EXP Information 1

The second ping between site 1 and 2 with 104 TOS value:

```
Winterfell-Site1#ping
Protocol [ip]: ip
Target IP address: 192.168.20.1
Repeat count [5]: 5
Datagram size [100]: 100
Timeout in seconds [2]: 2
Extended commands [n]: y
Source address or interface: 10.0.10.2
Type of service [0]: 104
Set DF bit in IP header? [no]: y
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
Packet sent with a source address of 10.0.10.2
Packet sent with the DF bit set
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 340/384/452 ms
```

Figure 5.42: EXP Information 3

```
MultiProtocol Label Switching Header, Label: 204, Exp: 3, S: 0, TTL: 254
    0000 0000 0000 1100 1100 .... .... .... = MPLS Label: 204
    .... .... .... .... .... 011. .... .... = MPLS Experimental Bits: 3
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
MultiProtocol Label Switching Header, Label: 408, Exp: 3, S: 1, TTL: 254
    0000 0000 0001 1001 1000 .... .... .... = MPLS Label: 408
    .... .... .... .... .... 011. .... .... = MPLS Experimental Bits: 3
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
Internet Protocol Version 4, Src: 10.0.10.2, Dst: 192.168.20.1
Internet Control Message Protocol
```

Figure 5.43: EXP Information 3

The third ping between site and 2 with 184 TOS value :

```
Winterfell-Site1#ping
Protocol [ip]: ip
Target IP address: 192.168.20.1
Repeat count [5]: 5
Datagram size [100]: 100
Timeout in seconds [2]: 2
Extended commands [n]: y
Source address or interface: 10.0.10.2
Type of service [0]: 184
Set DF bit in IP header? [no]: y
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
Packet sent with a source address of 10.0.10.2
Packet sent with the DF bit set
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 344/364/388 ms
```

Figure 5.44: EXP Information 5

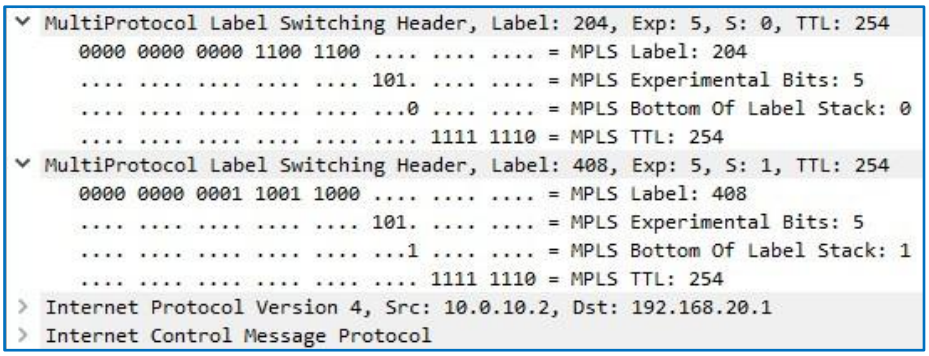

Figure 5.45: EXP Information 5

To verify the policy-map on each router we use show policy-map on the two interfaces , so here we can see the classes that are mapped on each router from router 1 to 5:

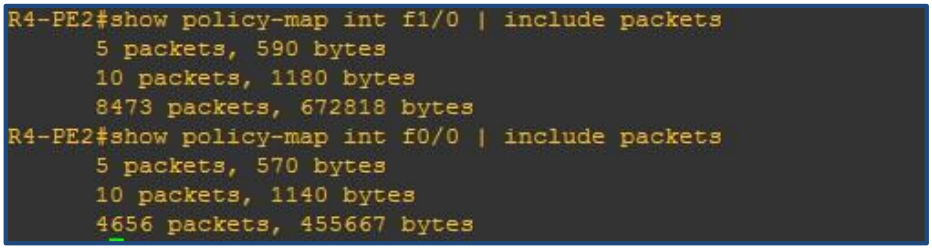

Figure 5.46: PE1 Policy-map Information

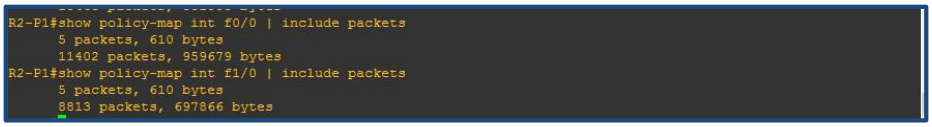

Figure 5.47: P1 Policy-map Information

On the router 4 we can see all the packets of classes which are transmitted via the network on the input interface and the output interface:

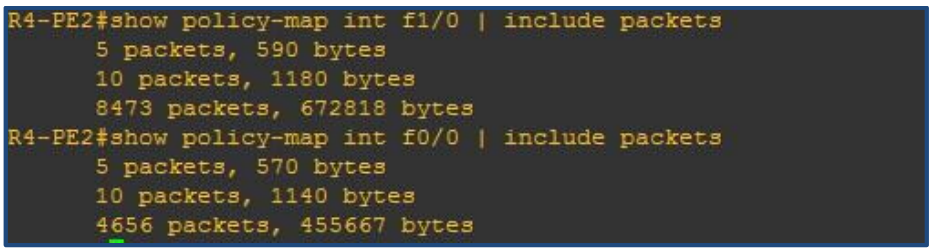

Figure 5.48: PE2 Policy-map Information

To check that all packets are received from site 1 we use:

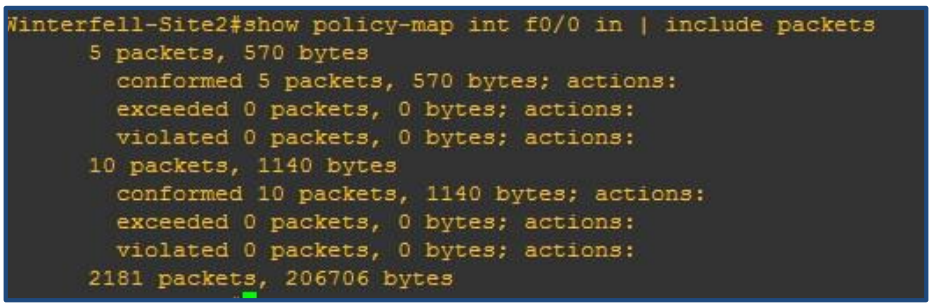

Figure 5.49: CE policy-map Information

5.7 Conclusion

In this chapter we have presented all the necessary steps to configure an MPLS network based on specific IGP and EGP protocols such as OSPF and BGP ,we've applied all diffrents mpls applications using the network emulator GNS3 and packet analyser wireshark.

this simulation allowed us to put in evidence several observations , First of all, it's obvious that the MPLS takes less processing time in forwarding the packets due to label switching,the implementation of MPLS with TE reduces network congestion and provides the better utilizations of network links ; we also appliesd MPLS VPN networks ,it means same IP address scheme can be given to two or more different VPNs it's more scalable than traditional IP VPNs.

Finally,we have implemented end-to-end quality of service DiffServ tunneling mode, we started with classify the traffic regarding his priority and importance, ip precedent and exp field used to marking traffic , and for each class we creat a policy that applied on the interface, then we checked the differents qos policies by using data analyser Wireshark.

General conclusion

This thesis confirmed the fact that MPLS is designed for its multitude applications. In this thesis we emphasized on the need for MPLS and its applications, VPN, TE and QoS. We examined the MPLS technology and found out that MPLS is a competent technique because it provides well-organized packet transmission, QoS, load balancing, scalability, consistency, and end-to-end connectivity. MPLS is mainly a connection oriented architecture which easily integrates in existing IP networks. Finally, we concluded that the MPLS architecture can help reducing network congestion together with an amazing possibility for added TE. This also provides an efficient way to utilize all available network resources infrastructure-wise.

Perspectives

MPLS has emerged as a powerful technology that forced several service providers to move their services from ATM or Frame Relay networks to MPLS networks. For instance, Cisco is working continuously to provide improved QoS in their network nodes and thus satisfy its users. IP high speed links were the dominant backbone of the Internet, but MPLS is gradually taking control and becoming an efficient replacement.

Applications such as VPN, VoIP, ERP, video conferencing and streaming need high QoS requirements and these types of services are being well served in an MPLS environment. Nevertheless, a progressive research work is required in order to make sure MPLS meets all such requirements and fulfills these high demands.

Bibliography

[1] CCIE Routing and Switching Study Guide

[2] Luc De Ghein, MPLS Fundamentals Cisco CCIE NO.* 1897

[3] Lee, Donn. Enhanced IP Services. Indianapolis: Cisco Press, 1999.

[4] Black, Uyless D. MPLS and Label Switching Networks. Upper Saddly River, New Jersey: Prentice Hall PTR,2001.

[5] Cisco Systems. Cisco IOS 12.0 Quality of Service. Indianapolis: Cisco Press, 1999

[6] Chrisitina Hattingh. END To END QoS Network Design, November 09,2004

[7] Design of MPLS networks VPN and TE with testing its resiliency and reliability, thesis by Michal Aron at Masaryk University, 2014.

[8] MPLS Fundamentals Video Course by Keith Barker (CBT nuggets)

[9] Differentiated Services.Wikipedia.http://en.wikipedia.org

/wiki/DSCPClassification.and.Marking.Accessed June 1, 2019.

[10] https://www.quora.com/Difference-between-layer-2-vpn-and-layer-3-vpn accessed may 25, 2019

[11] https://anetworkerblog.com/2008/08/10/ldp-transport-address/

[12] http://www.electricmonk.org.uk/2014/02/07/what-is-gns3/

[13] https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html