Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering
Telecommunication Engineering Graduate Program

Analysing Impact of Seamless MPLS on QoS

By

Habtamu Kumera

Advisor

Dr. Yalemzewd Negash

A Thesis Submitted to the School of Electrical and Computer Engineering in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Telecommunication Engineering

November, 2018

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Telecommunication Engineering Graduate Program

Analysing Impact of Seamless MPLS on QoS

By

Habtamu Kumera

Approval by Board of Examiners

_____	_____	_____
Chairman / School Dean	Signature	Date
Dr. Yalemzewd Negash	_____	_____
Advisor	Signature	Date
Dr. Ephrem Teshale	_____	_____
Examiner	Signature	Date
Dr. Beneyam Berehanu	_____	_____
Examiner	Signature	Date
_____	_____	_____
Director, Postgraduate Program	Signature	Date

Declaration

I, the undersigned, declare that this MSc thesis is my original work, has not been presented for fulfillment of a degree in this or any other university, and all sources and materials used for the thesis have been acknowledged.

Habtamu Kumera _____

Name Signature

Addis Ababa, Ethiopia _____

Place Date of submission

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Yalemzewd Negash _____

Advisor's Name Signature

Abstract

The need for timely delivery of real time and mission-critical applications has led to a high demand for end-to-end QoS guarantees. Service providers, including Ethio telecom, have deployed mobile backhaul in addition to IP core network with separate MPLS domains for each network. Internal network topology of one domain and its policy implementations are proprietary information and inter-domain routing must be done without detailed knowledge of the entire network topology, policies and performance of the other domains. Lacks of global coordination between policies used in different domains, limitations of BGP for implementing end-to-end QoS in inter-domain routing and its slow convergence during network failure are major challenges of the current inter-domain routing architecture.

As stated by different authors such as in RFC 8277 (Using BGP to Bind MPLS Labels to Address Prefixes), Seamless MPLS provides framework for taking MPLS end-to-end in a scalable fashion, extending benefits of traffic engineering (TE) and guaranteed service-level agreements (SLAs) with deterministic network resiliency. It offers an alternative for implementing end-to-end MPLS networks by integrating different domains into a single MPLS domain.

The motivation of this thesis is to investigate and analyze impacts of implementing Seamless MPLS on QoS parameters. The impact on QoS parameters is analyzed by using two scenarios; MPLS with multiple domains and Seamless MPLS integrating three domains in to single MPLS domain. Simulation tools such as eNSP, Ostinato, NQA and MATLAB are used to compare performances of the two scenarios. The analysis results show that in Seamless MPLS throughput is improved by 36.87%, latency is improved by 15.98%, packet loss is improved by 20% and jitter is improved by 12.5% compared to MPLS. From the results one can understand that any service provider can benefit from deploying Seamless MPLS.

Keywords—IP, MPLS, Seamless MPLS, QoS, MPLS domain, AS, Analysis, Performance

Acknowledgment

First and foremost, I would like to express my special thanks to Almighty God for giving me everything and also guiding me in all the ways of my life.

Second, my deep gratitude goes to Dr. Yalemzewd Negash for his continuous follow-up and guidance during the course of this thesis. His observation, unreserved advice and support were very useful and constructive.

Also I would like to express my appreciation to AAiT in collaboration with Ethio Telecom for their devotion and sponsorship to make this postgraduate program fruitful.

It would be difficult for me to complete this thesis without continuous support, encouragement and patience of my family members, specially my wife has a lion's share of my success.

Finally, I would like to thank my classmates and work colleagues for their endless support and advice.

Table of Content

List of Figures

List of Tables

List of Abbreviations

ABR	Area Border Router
AN	Access Node
ARP	Address Resolution Protocol
AS	Autonomous System
ASBR	Autonomous System Boundary Router
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
BGP-LU	Border Gateway Protocol Labeled Unicast
CR	Core Router
DiffServ	Differentiated Service
DoD	Downstream-on-Demand
DSLAM	Digital Subscriber Line Access Multiplexer
eBGP	External Border Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
eNSP	Enterprise Network Simulation Platform
FEC	Forwarding Equivalence Class
FTP	File Transfer Protocol
iBGP	Internal Border Gateway Protocol
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IP	Internet Protocol

IS-IS	Intermediate System-Intermediate System
ITU	International Telecommunication Union
LDP	Label Distribution Protocol
LLC	Logical Link Control
LSP	Label-Switched Path
LSR	Label Switching Router
MLD	Multicast Listener Discovery
MPLS	Multiprotocol Label Switching
NNTP	Network News Transfer Protocol
NQA	Network Quality Analyzer
OSPF	Open Shortest Path First
QoS	Quality of Service
RIP	Routing Information Protocol
RSVP-TE	Resource Reservation Protocol for Traffic Engineering
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time
SIP	Session Initiation Protocol
SLA	Service Level Agreement
S-MPLS	Seamless Multiprotocol Label Switching
SNAP	Subnetwork Access Protocol
TCP	Transmission Control Protocol
TE	Traffic Engineering
UDP	User Datagram Protocol
VPN	Virtual Private Network

1. Introduction

1.1. Background

The need for timely delivery of real time applications like telephony, video conferencing or guaranteed bandwidth for mission-critical applications has led to a high demand for end-to-end quality of service (QoS) guarantees such as delay, Jitter and packet loss [1]. QoS requirements put new challenges to service providers. QoS does not create capacity, but only supports the priorities of traffic and allocation of resources under the terms of congestion [2].

An autonomous system (AS) is a network or group of networks under a common routing policy which is managed by a single authority. Exchanging routing information within an AS is known as intra-domain routing. On the other hand, inter-domain routing focuses on the exchange of routes to allow the transmission of packets between different ASes. When an AS is connected to multiple different ASes, it is referred to as a multi-homed AS. On the other hand, ASes connected to a single AS are known as single-homed ASes.

Border Gateway Protocol (BGP) is the defacto standard for inter-domain (inter-AS) routing protocol used to exchange reachability information throughout the Internet [3]. When reachability information is exchanged between two BGP routers located in different ASes, the protocol is referred to as external BGP (eBGP). On the other hand, when reachability information is exchanged between BGP routers located inside the same AS, the protocol is referred to as internal BGP (iBGP).

QoS based routing within an administrative domain or AS is addressed using resource reserved tunnels in Multiprotocol Label Switching (MPLS) networks [4]. However, many connection requests span across multiple ASes e.g. a multi-national company having offices throughout the world and requiring them to be connected through a virtual private network (VPN) with a specific level of performance. Hence, QoS routing at the inter-domain level is essential. Such a task is significantly more complex and challenging than intra-domain QoS routing. Inherent nature and functionality of BGP makes the QoS extensions rather difficult. BGP suffers from lack of QoS support and currently researchers are looking to address this issue. It is essentially a path-vector protocol working under limitation of local AS policies [1]. BGP also has scalability issues due to the large routing table of sizes. This leads to problems such as long convergence time after link failures, route flaps, forwarding loops [5, 6]. The internal network topology of an AS, its business relationships with other ASes, and its policy implementations are treated as proprietary information and inter-domain routing must be done without detailed knowledge of the network topology, policies, and performance [3].

MPLS has been widely adopted by service providers as replacement of Asynchronous Transfer Mode (ATM) and Frame-Relay. It is universally deployed in most service provider core networks and has seen significant adoption within metro networks driven by increasing demand for video, mobile broadband and cloud services [8, 9, and 10]. The success of MPLS is due to its compatibility with the old technologies and the numerous advantages the technology brings. Some of the technologies supported by MPLS are QoS, Traffic Engineering (TE) and VPN [7]. MPLS is a technology that optimizes the traffic forwarding in a network by avoiding complex lookups in the routing table. The

traffic is directed based on labels contained in an MPLS packet header. This process allows very fast switching through an MPLS network. Figure 1.1 shows MPLS header and how a label is inserted between Layer 3 datagram and Layer 2 header. The total length of MPLS header is 32 bits (4 bytes or octets). The first 20 bits constitute a label values. Next 3 bit value called Traffic Class is used for QoS related functions. Next bit is Stack bit which is called bottom-of-stack bit. This field is used when more than one label is assigned to a packet, as in the case of MPLS VPNs or MPLS TE. Last byte is MPLS Time to Live (TTL) field is a mechanism that limits the lifespan or lifetime of data in a network [2].

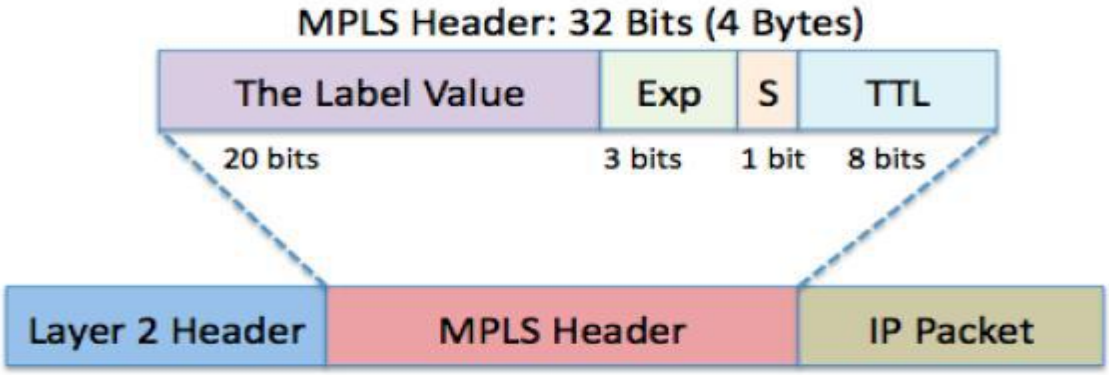

Figure 1.1: MPLS header [2]

The increased adoption of MPLS within operator networks demands for highly flexible, scalable, resilient and manageable network architectures. Seamless MPLS (S-MPLS) offers a superior alternative for implementing end-to-end MPLS networks by integrating access, aggregation and core networks into a single MPLS domain. It is also aimed for optimizing and improving the efficiency of end-to-end MPLS networks. With S-MPLS architecture, the entire network uses unified IP/MPLS networking technology, with an end-to-end control plane [11].

1.2. Statement of the Problem

With growing trends of today's technology, service providers are changing their infra-structures and network architecture where multiple MPLS domains are being formed. In MPLS, provisioning of QoS is limited to within a domain (intra-domain), usually IP core network and/or mobile backhaul. This is because BGP does not support QoS implementation across multiple domains (inter-domain or inter-AS). Another limitation is that due to its inherent nature, BGP has slow convergence time (about 30 seconds) when network encounters a failure. Implementation of QoS in intra-domain network alone does not guarantee end-to-end QoS in the whole network because lack of QoS in inter-domain network has an impact on the overall QoS.

To address these problems one convergent inter-domain network architecture, Seamless MPLS, is proposed to enhance manageability, service provisioning and scalability but its impact on QoS parameters is not tested and quantified to ensure end-to-end QoS guarantees. Even though some researchers have conducted traffic analysis for single MPLS domain, impact analysis of Seamless MPLS on QoS is left open as to knowledge of the author. This analysis is important to identify pros and cons of replacing the existing multi-domain MPLS with Seamless MPLS on the basis of QoS parameters.

1.3. Objective

1.3.1. General Objective

The general objective of this thesis is to analyze impact of an end-to-end MPLS architecture for enhancing service providers' IP network scalability and flexibility in service delivery using QoS parameters in comparison with the classical MPLS architecture.

1.3.2. Specific Objectives

The specific goals of the study are:

- To investigate the technologies supported by Seamless MPLS
- To simulate and evaluate Seamless MPLS architecture
- To assess the impact of Seamless MPLS on QoS parameters
- To compare Seamless MPLS architecture with MPLS architecture and its potential improvements on QoS.

1.4. Methodology

In this thesis state-of-the-art, related works and statement of the problem are used as baseline to achieve the objectives. The methodology starts with investigating different technologies enabling Seamless MPLS architecture. Then the methods of simulating and evaluating the architecture with QoS perspective are followed. A theoretical study of MPLS and QoS features are done thoroughly along with the evaluation of the limitations of the MPLS architecture. Seamless MPLS architecture & implementation scenarios with its benefits compared to MPLS architecture are explained.

In the implementation part, a practical environment is developed using network simulation tool, Enterprise Network Simulation Platform (eNSP), and two scenarios are built in order to collect test results from the simulator. The two scenarios are built in such a way that first an ordinary network is built with an MPLS network. Then the same network topology is implemented with Seamless MPLS features and the test results are collected from the simulator using Network Quality Analyzer (NQA) technology for the two scenarios. To make the scenarios similar to the real network, a network traffic generator called Ostinato is used to generate traffic into the network.

Finally, the test results for the two scenarios are presented in graph using MATLAB for comparison and analysis with respect to QoS parameters.

1.5. Scopes and Limitations

1.5.1. Scopes of the Thesis

In Seamless MPLS large numbers of MPLS domains can be aggregated to single domain. However, in this thesis three representative MPLS domains are used for the implementation of Seamless MPLS, considering the results are equally applicable for the other domains. The three domains can be considered as two access and aggregation MPLS networks connected by a core MPLS network. Also in the thesis MPLS IP unicast is considered and the performance analysis is independent of type of traffic generated. So all network traffic entering in to the network are treated equally.

1.5.2. Limitations of the Thesis

Due to memory limitations of personal computers and the process intensiveness of the simulation tools used, it is not possible to power on more than ten nodes (routers) simultaneously in simulation environments and additional routers for redundancy and load balancing purposes are used only in the core network domain. But it should be noted that increasing number of routers for testing and analysis will not alter the overall result.

1.6. Contributions

MPLS is a key technology in every service providers' networks and its optimum implementation is an important factor. Many service providers, including Ethio telecom, have deployed mobile backhaul in addition to IP/MPLS core networks. The interconnec-

tion as well as service provisioning among these different domains should be done seamlessly without introducing additional delay and flexibility problems. This thesis aims to improve end-to-end network QoS performance by optimizing the classical MPLS architecture using newly emerging technologies. This minimizes limitations of the existing MPLS architecture and enhances the scalability and flexibility of service delivery in any telecommunication industry.

1.7. Literature Review

The problem of how to extend QoS capabilities across multiple provider domains has not been solved satisfactorily to date. The source of the problem lies mainly with the autonomous nature of Internet Service Providers (ISP) and their loose federation that forms the global Internet [13**].**

The authors in [14] have described the approaches for end-to-end QoS support based on extending BGP to advertise inter-domain QoS routing information. Since, the requirements for inter-domain QoS put a large volume of information to be processed at routing plane; it is not scalable for the entire global Internet.

Another attempt was to set up inter-provider, inter-domain MPLS tunnels [15] with specific resource reservations and QoS guarantees. However, in the absence of any accepted framework it requires protracted discussions and agreements between two service providers.

The researchers in [1] present an approach to achieve end-to-end QoS support by proposing a new Alliance Network model. The Alliance Network sets-up inter-domain paths for premium traffic. It requires specific QoS guarantees using inter-domain MPLS

tunnels with resource reservation. Premium traffic that requires the end-to-end QoS guarantees should use the new facility by paying premium rates. Since the model assumes prior agreement on the revenue sharing mechanism and exchange of QoS information, it is not scalable to global Internet.

As in [16], inter-domain routing is considered a challenging research area due to two reasons: First, the inter-domain routing protocol, BGP, currently used in the Internet has several limitations, but its replacement is not a realistic option due to its worldwide deployment. Second, inter-domain routing denotes routing among distinct domains or networks. These domains are completely autonomous entities, which perform their own routing management based on policies that only have local significance.

Griffin and Presmore have showed in [17] that the arbitrary 30 seconds minimum route advertisement interval value has a huge impact on BGP convergence time. They observed that for each network topology and a particular set of experiments, there is an optimal value of the minimum route advertisement interval timer. This optimal value can significantly reduce the convergence time of BGP. Unfortunately, this might be extremely hard to find in practice since it varies from network to network. Several authors have proposed modifications to reduce the BGP convergence time in case of failures. The ghost-flushing approach proposed in [18] improves the BGP convergence by ensuring that the messages indicating bad news are distributed quickly by the BGP routers, while good news propagates more slowly. The downside of ghost-flushing is that it does not tackle the root of the problem. Instead, it only tries to speed up the convergence of BGP.

Some researchers have identified the drawbacks of BGP on QoS at inter-domain level. An important performance metric for a routing protocol is its convergence time (i.e., the time required to reroute packets around a failure). The first significant studies of the convergence of BGP were carried out using measurements in the Internet [19]. These studies showed that the convergence of BGP was rather slow, often measured in tens of seconds. This slow convergence is caused by several factors, some of which are inherent to the utilization of path vectors by BGP, while others are due to implementation choices. Other studies such as [20] have shown that BGP routing tables are growing significantly fast, which imposes a considerable pressure on the scalability of BGP.

Lack of global coordination between the policies used in different domains is a major weakness of the current inter-domain routing paradigm. Studies such as [21] have demonstrated that without coordination, the interaction between independent policies may lead to global routing anomalies, such as inconsistent recovery from link failures or even route oscillations. They showed that the main reasons for the absence of cooperation between domains are the characteristics of the BGP policy's expressiveness and the ASes are not willing to disclose details about their internal configuration and policies.
[22] Shows that the conservative behavior of Transmission Control Protocol (TCP) retransmissions actually aggravates the instability of BGP sessions when network failures occur. Other limitations of BGP stated in [23] are security issues, lack of multipath routing and limited traffic engineering capabilities.

As shown in [12], in traditional IP/MPLS network design and deployments, there is a tight coupling between the network nodes and the services delivered over it and the services are provisioned in multiple segments. The paper presents proposals for extend-

ing MPLS beyond the core to the aggregation and metro area network to reap some of the same benefits that MPLS provides in the core. Taking MPLS to the access, and making MPLS the packet forwarding platform end-to-end across the network, requires new functionality and features and a systematic architecture that can scale to tens of thousands of nodes.

The internet engineering task force (IETF) standard in RFC 8277 [8] and RFC 7032 [24] are aimed to address the drawbacks of traditional MPLS such as scalability and flexibility in service provisioning limitations. The scalability is achieved by using label distribution protocol (LDP) Downstream-on-Demand (DoD) label advertisements. To enhance the flexibility in provisioning, label mapping information for a particular route is piggybacked in the same BGP update message that is used to distribute the route itself.

1.8. Thesis Layout

This thesis is composed of six chapters. Chapter one deals with introduction to the thesis. It includes background information, statement of the problem, objectives of the study, the methods how the objectives are achieved, scopes and limitation of the thesis, contributions of the thesis and related works.

Chapter 2 introduces basic concepts in MPLS technology. It highlights the advantages of MPLS compared to other old technologies and the most common terminologies used in MPLS are briefly explained in this chapter. Chapter 3 is a detail description of Seamless MPLS. The architecture and key technologies supported in Seamless MPLS along with the benefits this architecture brings are presented in this chapter.

Chapter 4 is all about IP QoS principles, parameters and models used in the current IP/MPLS networks. The four QoS parameters such as throughput, latency, packet loss and jitter are discussed in detail and their recommended values are also listed. Chapter 5 presents simulation and result analysis part which describes about the simulation tools used, simulation scenarios, network topology and analysis of the results obtained.

The final chapter concludes the thesis by drawing conclusions from the analysis part. Potential research area for future work is also included in this chapter. References and appendixes are also included at the end of this document.

2. Introduction to MPLS

MPLS is a forwarding technique in telecommunications networks that forwards packets from one node to the next node on the basis of short labels attached in packets instead of looking long IP addresses at every router. This helps in reducing core routers' time which do not need complex routing table lookups. Hence it increases data transfer speed significantly.

MPLS is best described as a "layer 2.5 networking protocol", because it is located between layer 2 and layer 3 of the Open System Interconnection (OSI) model, providing essential features for the data transfer across the network. MPLS is known as "Multiprotocol" as it can work with various network protocols. MPLS supports variety of access technologies, which includes ATM, Frame Relay, etc. [25].

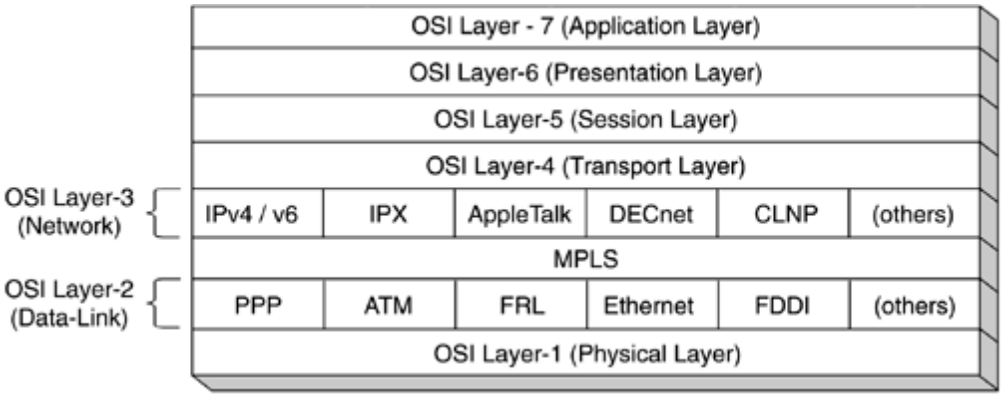

Figure 2.1: Position of MPLS in OSI model [25]

MPLS has been widely adopted by service providers around the world as replacement of ATM and Frame-Relay. The huge success of MPLS is explained not only with the numerous advantages the technology brings, but also with its compatibility with the old technologies.

2.1. Benefits of MPLS

MPLS enables enterprise and service providers to create a next generation intelligent network that offers value-added services over a single infrastructure. The main benefits the technology brings are listed in [27] and summarized as below:

- MPLS can be implemented over the most widely used legacy or new infrastructures (Synchronous Optical Network (SONET), 100M/1000M/10G Ethernet) and networks (IP, ATM, Frame Relay and Ethernet). Thus it enables the migration towards IP-based infrastructures.
- MPLS provides traffic engineering capabilities, which enables more efficient use of the available bandwidth by spreading the traffic more evenly over all available links. The implementation of traffic engineering in an MPLS network helps to deliver the traffic in a path that is different from the least-cost path.
- MPLS supports Quality of Service (QoS)
- MPLS reduces core routers processing requirements since they simply forward packets based on labels
- MPLS provides the appropriate level of security and reduces the need of encryption on public IP networks
- MPLS enables the deployment of provider-network-based VPNs, which provides private and secure networks over the same network topology to end customers. Therefore, it has a cost advantage for the customers.
- MPLS provides scalability. The core devices are not involved in any relationship with the other networks (not meshed), and their task is only to switch packets. The virtual tunnels are built to connect with the core parts of the network that shorten the amount of virtual path.

2.2. Basic Concepts in MPLS

MPLS is a technology that optimizes the traffic forwarding in a network by avoiding complex lookups in the routing table. The traffic is directed based on labels contained in an MPLS packet header. The labels define only the local node to node communication and are swapped on every node. This process allows very fast switching through the MPLS core. MPLS relies on traditional IP routing protocols to determine the best routes and to receive topology updates and predetermines the path the packet will take through the network. This process is performed by the MPLS edge router and thus reduces the processing requirements for the core switching routers [26]. The terminologies in MPLS are summarized in the following sections [27, 29].

2.2.1. Forwarding Equivalence Class (FEC)

MPLS technology is based on classification. It groups the packets that will be forwarded in the same way in forwarding equivalence classes (FEC). The classification criteria may vary and include source address, destination address, source port, destination port, protocol type and VPN. The packets belonging to a specific FEC will then be forwarded to the same label switched path (LSP). When a packet arrives, the router will examine it and determine whether it belongs to an existing FEC. FEC are neither labels, nor packets, but logical entities created by the router.

2.2.2. Label

A label is short fixed-length identifier that points to a specific FEC. A label may represent only a single FEC, but a FEC can correspond to multiple labels. The label is part of the packet header and is only locally significant, as it does not carry any topology information.

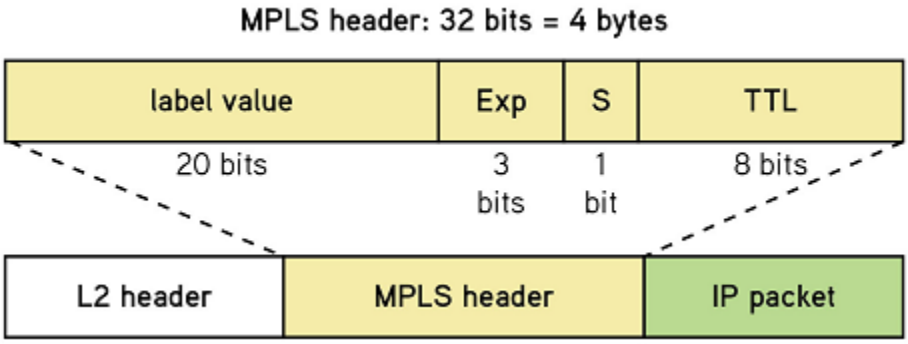

Figure 2.2: MPLS packet header [27]

2.2.3. Label Switching Router (LSR)

LSRs are the fundamental components of the MPLS network and can be three different types depending on their function in the network:

a) **Label Edge Router (LER)** – situated on the periphery of the network and serves as a gateway between the MPLS network and the WAN or the Internet. A LER can be:

- **Ingress router** – it is the entry point of the MPLS network. When a packet arrives it decides whether the packet should be forwarded through the MPLS network, determines the FEC, the packet belongs to, and encapsulates it with an MPLS header, based on the information it carries.
- **Egress router** – it is the exit point of the MPLS network. It performs a normal IP look-up and forwards the packet according to the appropriate IP routing protocol.

b) **Transit router** – it is any router in the middle of the MPLS network and performs simple switching, based on the label value.

c) **Penultimate router** – it is the router before the last hop in the MPLS network. As the packet will not be switched to another transit router, the penultimate router removes the MPLS header, before forwarding the packet to the egress router. The use of penultimate router configuration is optional, as the MPLS header can also be removed by the egress router. In that case the penultimate router operates as a transit router.

2.2.4. Label Switched Path (LSP)

Label switched path defines the path the packets from a particular FEC will follow through the MPLS network. The LSP is a unidirectional path from the ingress to the egress router and functions like a virtual circuit. The LSP is established by a signaling protocol, such as LDP or Resource Reservation Protocol for Traffic Engineering (RSVP-TE). Simple schematic of an MPLS network that illustrates the concept of LSP is presented in Figure 2.3.

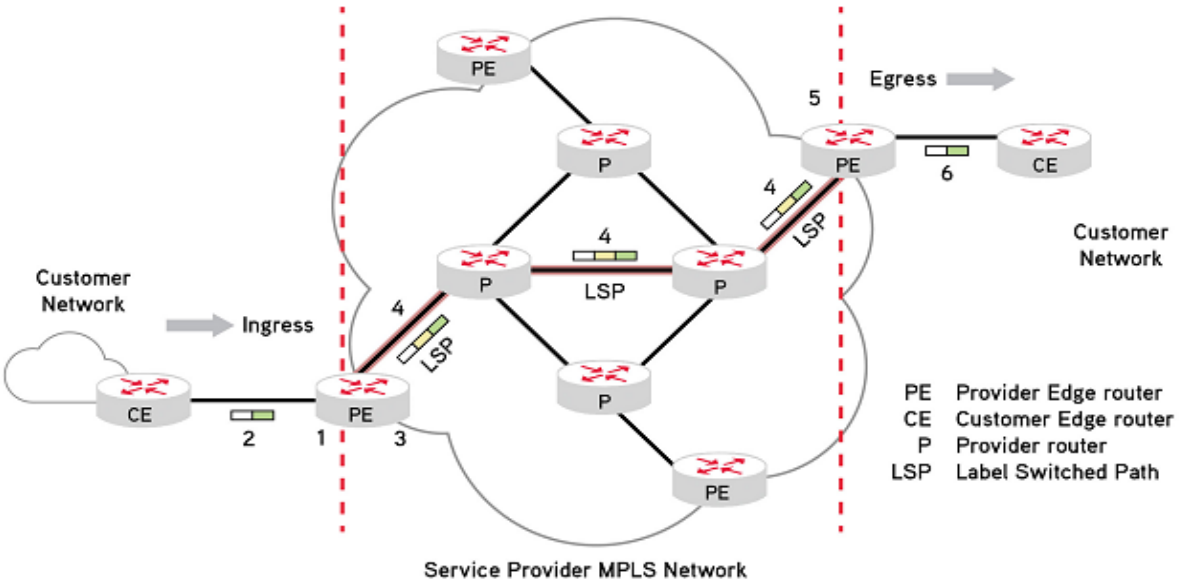

Figure 2.3: MPLS network [27]

2.2.5. Label Distribution Protocols (LDP)

LDP protocol enables the LSRs to request, distribute and release label binding information. The label distribution method is called hop-by-hop forwarding and is performed along the normally routed paths, chosen by the underlying Interior Gateway Protocol (IGP) routing protocol. The resulting LSPs are then used to forward label traffic across the MPLS network.

2.3. Principle of Operation

When an unlabeled packet enters the ingress router and needs to be passed on to an MPLS tunnel, the router first determines the FEC for the packet and then inserts one or more labels in the packets newly created MPLS header. The packet is then passed on to the next hop router for this tunnel. The MPLS header is added between the network header and data link layer header of the OSI model. When a labeled packet is received by an MPLS router, the topmost label is examined. Based on the contents of the label a swap, push (impose) or pop (dispose) operation is performed on the packet's label stack. Routers can have prebuilt lookup tables that tell them which kind of operation to do based on the topmost label of the incoming packet so they can process the packet very quickly [25].

In a swap operation the label is swapped with a new label, and the packet is forwarded along the path associated with the new label. In a push operation a new label is pushed on top of the existing label, effectively "encapsulating" the packet in another layer of MPLS. In a pop operation the label is removed from the packet, which may reveal an inner label below. This process is called "decapsulation". If the popped label is the last on the label stack, the packet leaves the MPLS tunnel. This is usually done by the egress

router. During these operations, the contents of the packet below the MPLS Label stack are not examined. Indeed transit routers typically need only to examine the topmost label on the stack. The forwarding of the packet is done based on the contents of the labels, which allows "protocol- independent packet forwarding" that does not need to look at a protocol-dependent routing table and avoids the expensive IP longest prefix match at each hop. At the egress router, when the last label has been popped, only the payload remains. This can be an IP packet, or any of a number of other kinds of payload packet. The egress router must therefore have routing information for the packet's payload, since it must forward it without the help of label lookup tables. An MPLS transit router has no such requirement [25].

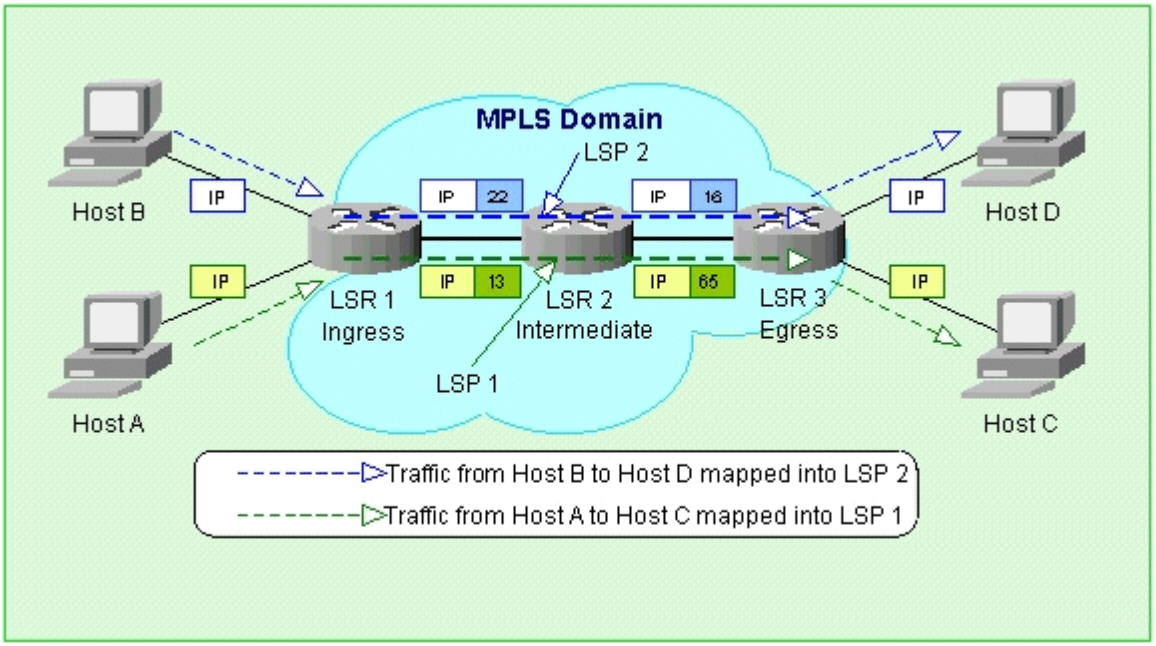

Figure 2.4: MPLS operation [28]

2.4. Label Distributions

When the packet enters the MPLS topology, ingress LSR receives the packet and imposes the MPLS label to the packet and forwards to the next hop via the Label Switch Path. When the packet reaches the next LSR, i.e., the intermediate LSR, it swaps the incoming label with the outgoing label and transmits the packet. When the egress LSR receives the packet, it strips off the packet label and forwards it to the destination router.

All the LSRs present in the MPLS network have IGP (e.g., EIGRP, RIP, OSPF, etc.) running throughout the network [28]. To accomplish the label distribution task, adjacent LSRs must agree on the label that is used for each IGP prefix. Each LSR should be able to identify the swapping of incoming and outgoing labels. Since the labels are local to adjacent routers that do not have global meaning across the network, we need a mechanism to instruct the routers which label should be used while forwarding the packets. Therefore, two adjacent routers need some sort of communication between them to agree on which label to use for a particular prefix. Otherwise, the routers do not get any idea about the swapping packets. For this purpose or to complete label distribution, the Label Distribution Protocol is needed. Label distribution is carried out in two different following ways [28]:

- Piggyback the labels on an existing IP routing protocol.
- Have a separate protocol distribute labels.

2.4.1. Piggyback the Labels on an Existing IP Routing Protocol

In this method, LSRs do not need new protocol but they need to extend the existing routing protocol to carry labels. There is a great advantage of this method because the routing and label distribution are always in sync which means both labels and prefix

should be present. The implementation is very easy for the distance vector routing protocols, e.g., EIGRP, which originate the prefix from the routing table**.** Then the router binds the label with that prefix.

Link state routing protocols (e.g., OSPF) work differently from the distance vector protocols. In link-state routing protocols, each router originates link state updates and forwards the original updates by all the routers in the same area. Nevertheless, the problem with MPLS is that every router needs to distribute labels for each IGP prefix even to the router that does not originate a prefix. A separate protocol is required for label distribution in link state routing protocols. BGP is the one routing protocol in the MPLS VPN which can carry prefixes and distribute labels at the same time [8].

2.4.2. Separate Protocol for Label Distribution

This label distribution method needs a separate protocol to distribute the labels and lets the routing protocol to distribute the prefixes. The advantage of this method is routing protocol independent, and the disadvantage is that a new protocol is needed in each LSR. There are several varieties of protocols that distribute labels including:

- Tag Distribution Protocol (TDP)
- Label Distribution Protocol (LDP)
- Resource Reservation Protocol (RSVP)

TDP was the first protocol developed and implemented by Cisco for label distribution. LDP was later designed and developed by IETF. TDP and LDP operate in a similar way, but LDP has more functionality than TDP. Due to the easy availability of LDP, TDP was replaced by LDP in a very short time frame. RSVP is only used for MPLS traffic engineering.

2.5. Control Plane and Forwarding Plane

Control plane and forwarding plane are the part of router architecture. Control plane collects the information that is used to forward the incoming packets. While forwarding plane decides how to switch the incoming packets after being received at inbound interface [28].

2.5.1. Control Plane

The control plane exchanges routing information and labels with the adjacent routers. Routing information is advertised to any of the routers in the MPLS domain whereas label binding information is advertised to only adjacent routers by link-state routing protocols. It consists of two types of protocols namely routing protocols (e.g., RIP, EIGRP, OSPF, and BGP) and label exchange information protocols (e.g., LDP, TDP, RSVP, etc.).

2.5.2. Data Plane

Data plane has a forwarding plane that is based on the information attached to labels. There are two types of tables, namely Label Information Base (LIB) and Label Forwarding Information Base (LFIB). LFIB is used by the data plane to forward the labeled packets. LIB table contains all the local labels and the mapping of the labels which is received from the adjacent routers. The information in LFIB and label value is used by the MPLS-enabled routers to make forwarding decisions.

3. Seamless MPLS

3.1. Introduction

The need for one converged packet network to deliver all fixed and mobile services, regardless of the access technology, advances from time to time. The success of MPLS in core networks and the benefits it brings have enabled the way for the technology to be implemented in aggregation and access networks as an alternative to ATM or legacy Ethernet-based aggregation. Now the mobile backhaul service has been deployed widely, the requirement of the integration of mobile backhaul networks and core networks has been proposed [30]. Deploying a service from one MPLS region to another requires provisioning at several intermediate points in the end-to-end network, making troubleshooting and fault recovery more complex. A preferred approach would be to deploy a single end-to-end service and transport network architecture [10].

Seamless MPLS provides the framework for taking MPLS end-to-end in a scalable fashion, extending the benefits of traffic engineering and guaranteed service-level agreements (SLAs) with deterministic network resiliency. In Seamless MPLS all forwarding of packets within a network, from the time a packet enters the network until it leaves the network, is based on MPLS labels [12].

The motivation of Seamless MPLS is to provide an architecture which supports a wide variety of different services on a single MPLS platform fully integrating access, aggregation and core networks by the addition of extra features with classical/traditional MPLS and it gives more scalability, security, simplicity, manageability and flexible end-to-end service delivery. In order to obtain a highly scalable architecture Seamless MPLS takes

into account that typical access devices such as Digital Subscriber Line Access Multiplexer (DSLAM) and Multi-service access node (MSAN) are lacking some advanced MPLS features, and may have more scalability limitations. Hence access devices are kept as simple as possible [9]. Seamless MPLS is not a new protocol suite but describes architecture by deploying existing protocols like BGP, LDP, OSPF and ISIS (Intermediate System-Intermediate System).

3.2. Benefits of Seamless MPLS

In the past it was necessary to provide connectivity between the different domains and the core on per service level and not based on MPLS (e.g. by deploying native IP Routing or Ethernet based technologies between aggregation and core). In most cases service specific configurations on the border nodes between core and aggregation were required. New services led to additional configurations and changes in the provisioning tools. With Seamless MPLS there are no technology boundaries and no topology boundaries for the services. Network (or region) boundaries are for scaling and manageability, and do not affect the service layer, since the transport pseudowire (layer 2 VPN) that carries packets from the access node to the service node doesn't care whether it takes two hops or twenty, nor how many region boundaries it needs to cross. The network architecture is about network scaling, network resilience and network manageability; the service architecture is about optimal delivery: service scaling, service resilience (via replicated service nodes) and service manageability. The two are decoupled: each can be managed separately and changed independently [9]. In the following subsections key characteristics offered by Seamless MPLS is discussed [12].

3.2.1. Deterministic End-to-end Service Restoration

Seamless MPLS is a resilient network which provides a deterministic end-to-end service restoration (Sub-50ms), and there are two broad categories of functions that help achieve this. The first set of functions includes ways to enable speedy detection of performance degradation events and location of failures. The second set of functions is comprised of the appropriate recovery actions needed to reroute and restore services.

Failure Detection Mechanisms: There are various failure detection mechanisms available. Layer2 failure detection relies on Ethernet Operation, Administration, and Maintenance (OAM) capabilities, as well as integration of Bidirectional Forwarding Detection (BFD) mechanisms with LSP and pseudowires. For Layer3 fault detection and to test data plane consistency of pseudowires, both single hop and multi-hop BFD specified in RFC 5883 and RFC 5884 are supported for BGP sessions and targeted LDP sessions.

Failure Recovery and Service Repair: The choice of recovery mechanism used to restore services is often dependent on the location and type of failure on the network. In order to achieve sub-second convergence subsequent to a network failure, the first line of defense is to use local repair techniques and precomputed paths to reroute around the failures. These are implemented on top of nonstop active routing (NSR)-enabled control plane protocols. Local repair techniques include loop-free alternate (LFA) support for ISIS, OSPF and LDP. Link and node protection can also be enabled with RSVP-TE MPLS fast reroute and provide deterministic service restoration [10].

3.2.2. Decoupled Network and Service Architectures

With other end-to-end MPLS options (e.g., end to-end LDP in a flat network) IGP or MPLS signaling information is not contained within the region and is exchanged across

regions. This increases the size of routing/forwarding tables as well as the MPLS state within individual routers. The Seamless MPLS model addresses this challenge by introducing a hierarchy of transport and service layers. The Seamless MPLS transport layer consists of an inter-region tunnel and an intra-region tunnel.

One of the main motivations of Seamless MPLS is to get rid of service specific configurations between the different MPLS islands. Seamless MPLS connects all MPLS domains on the MPLS transport layer providing a single transport layer for all services - independent of the service itself. The Seamless MPLS architecture therefore decouples the service and transport layer and integrates access, aggregation and core into a single platform supporting residential, wholesale, mobile, and business subscribers [10]. One of the big advantages is that problems on the transport layer only need to be solved once (and the solutions are available to all services). With Seamless MPLS it is not necessary to use service specific configurations on intermediate nodes; all services can be deployed in an end-to-end manner [9]. This allows services to be provisioned wherever they are needed with flexible topological placement of services – enabler for per service de-centralization, no matter how the underlying transport is laid out. Figure 3.1 shows how network and service layers are decoupled.

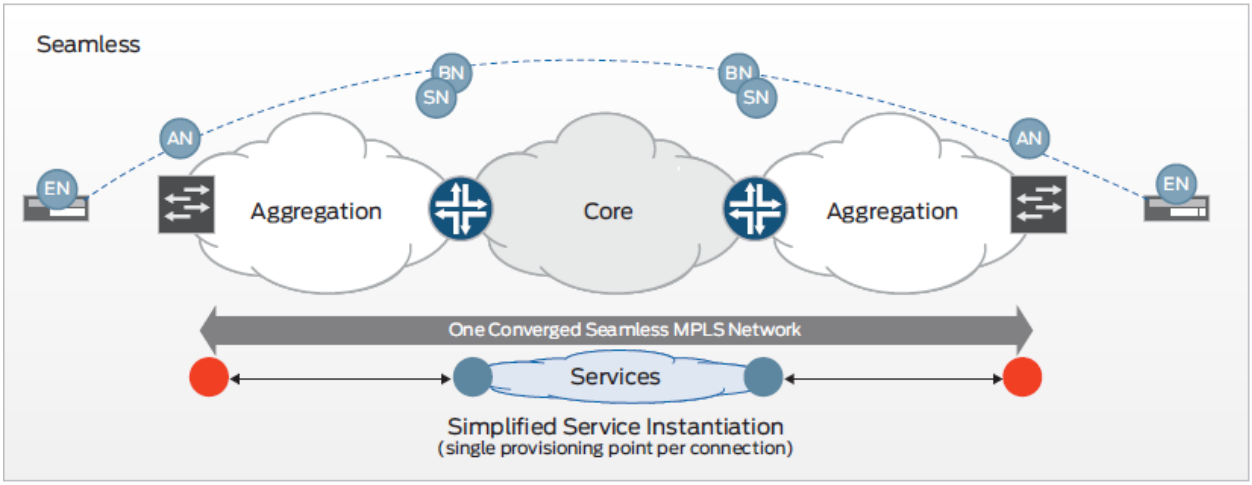

Figure 3.1: Decoupled service and network architecture

3.2.3. Service Flexibility with Simplified Provisioning and Operations

Seamless MPLS architecture suggests a systematic way of enabling MPLS end-to-end between access nodes, with all forwarding based on MPLS labels. Using this approach, packets are labeled at the access network entry point and are transported as labeled packets all through the network to the receiving end. This means that all service provisioning and operations are MPLS-based. There is a clean and homogenous separation of control plane, management plane, and data plane operations throughout the network that allows decoupling the service provisioning plane from the underlying transport technology. It also makes way for optimizing and simplifying service provisioning and operations. Seamless integration of different regions into single domain makes managing and troubleshooting the transport and services layer more efficient. Hence, service delivery and operations are greatly simplified, minimizing the number of service provisioning points, and making the topological placement of service delivery points highly flexible.

3.2.4. Traffic Engineering

Advantage of using labels and not the destination IP address is that packet forwarding decision can be made on the other factors such as traffic engineering and QoS requirements. The ability to traffic engineer based on real-time network attributes supports strict SLAs with guaranteed service availability. One of the goals of Seamless MPLS is to extend traffic engineering capability end-to-end across the access network.

3.2.5. Building Scalable Networks

Seamless MPLS helps scale the end-to-end network to more than 100,000 MPLS devices, recognizing that some nodes (e.g., access) have limited capabilities and they are typically optimized for simplicity and lower cost. Clearly this requires some new thinking and innovative techniques to deliver this scale.

3.3. Seamless MPLS Architecture

One of the key elements to be considered when designing architecture for a Seamless MPLS network is to handle the total size of the necessary routing and MPLS label information control plane and forwarding plane state resulting from the stated scalability goals especially with respect to the total number of access nodes. This needs to be done without affecting the technical scaling limits of any of the involved nodes in the network (access, aggregation and core) and without introducing too much complexity in the design of the network while at the same time still maintaining good convergence properties to allow for quick MPLS transport and service restoration in case of network failures [9].

The MPLS domains are connected in a hierarchical fashion that enables the seamless exchange of loopback addresses and MPLS label bindings for transport LSPs across the entire MPLS internetwork while at the same time preventing the flooding of unnecessary routing and label binding information into domains or parts of the network that do not need them. Such a hierarchical routing and forwarding concept allows scalability in different dimensions and allows hiding the complexity and size of the aggregation and access networks [9].

The intra-domain routing within each of the MPLS domains (i.e. aggregation domains and core) utilize standard IGP protocols like OSPF or ISIS. The intra-domain MPLS LSP setup and label distribution utilize standard protocols like LDP or RSVP [9]. The inter-domain routing is responsible for establishing connectivity between and across all MPLS domains. The inter-domain routing establishes a routing and forwarding hierarchy in order to achieve the scaling goals of Seamless MPLS. Note that the IP aggregation usually performed between region (IGP areas or AS) in IP routing does not work for MPLS as MPLS is not capable of aggregating FEC (because MPLS forwarding use an exact match lookup, while IP uses longest match) [9].

Because of the large quantity of access nodes, the cost of these nodes is extremely relevant for the overall costs of the entire network, i.e. access nodes are very cost sensitive. This makes it desirable to design the architecture such that the access node functionality can be kept as simple as possible [9].

Seamless MPLS architecture describes a systematic way of enabling MPLS end-to-end in a single domain. It enables complete virtualization of network services with service

origination and termination at the access nodes. Seamless MPLS is an architectural enabler for scale, resiliency, and service flexibility.

Classical MPLS network deployments are built with an implicit tight coupling between the network nodes, the underlying transport technology, and the services delivered over the network. This model provides limited flexibility in provisioning, as it is tightly coupled with the topological placement of the network nodes and operationally one has to deal with multiple technologies for troubleshooting and fault recovery. Figure 3.1 shows MPLS with multiple domains.

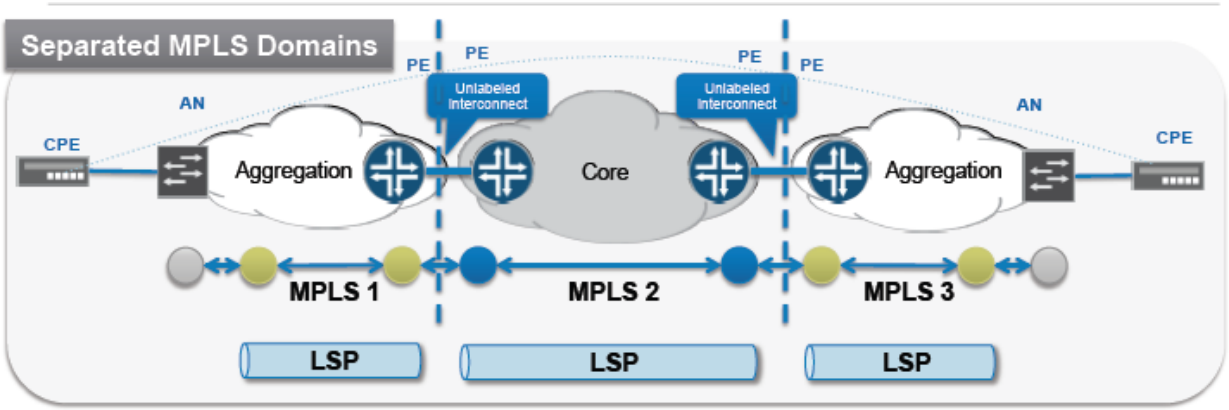

Figure 3.2: MPLS with multiple domains [12]

With Seamless MPLS, the idea is to provision the service end-to-end and minimize the number of provisioning points. The service provisioning is in-line with the network architecture, maintains simplicity in the access network, and relies on increased capabilities and intelligence on the service nodes. At the same time, it also simplifies operations and makes efficient use of network resources by reducing the number of provisioning points and relying on a single MPLS-based forwarding scheme in the data plane.

MPLS domains (regions) can be of different types: IGP area, IGP instance or BGP AS, all spanned by a single MPLS network, with any to any MPLS connectivity. Each region is responsible for connectivity (both IP and MPLS) within the region and can independently decide whether to run LDP or RSVP-TE or even LDP-over-RSVP. Region border nodes are responsible for inter-region connectivity and this is done by LSP hierarchy based on "labeled BGP" [8, 10].

Since the service is initiated as an MPLS pseudowire from the origination point at the access node, any topological changes in the access can be easily made without having to completely re-provision the service layer. This can be a significant operational asset to mobile backhaul access, for example, where re-parenting of cell site routers to a different base station controller/radio network controller (BSC/RNC) is a common occurrence.

Design Considerations:

- Split the network into domains: access, aggregation and core
- Single IGP with AS per domains
- Hierarchical LSPs to enable end-to-end LSP signaling across all domains
    - Large scale is achieved with hierarchical design
- IGP + LDP for intra-domain transport LSP signaling
- RSVP-TE alternative to LDP for traffic engineering
- BGP labeled unicast for cross-domain hierarchical LSP signaling
    - enables any-to-any connectivity between more than 100,000 devices
- LDP DoD and Static routing on access devices

New protocols are not used here, only existing protocols such as MPLS, LDP, IGP, and BGP are used. Since it is not required to distribute the loopback prefixes of the edge routers from one part of the network into another part, it is carried in BGP. The iBGP is used in one network, so the next hop address of the prefixes is the loopback prefixes of the edge routers, which is not known by the IGP in the other parts of the network. This means that the next hop address cannot be used to recurse to an IGP prefix. The trick is to make the ABR routers Route Reflectors (RR) and set the next hop to self. Since the RRs advertise the BGP prefixes with the next hop set to "themselves", they assign a local MPLS label to the BGP prefixes. This means that in the data plane, the packets forwarded on these end−to−end LSPs have an extra MPLS label in the label stack. Figure 3.2 shows inter-AS Seamless MPLS architecture. The access and aggregation layers are within a single AS, and the core layer belongs to another AS. Then the three domains are integrated end-to-end into one MPLS domain using Seamless MPLS architecture.

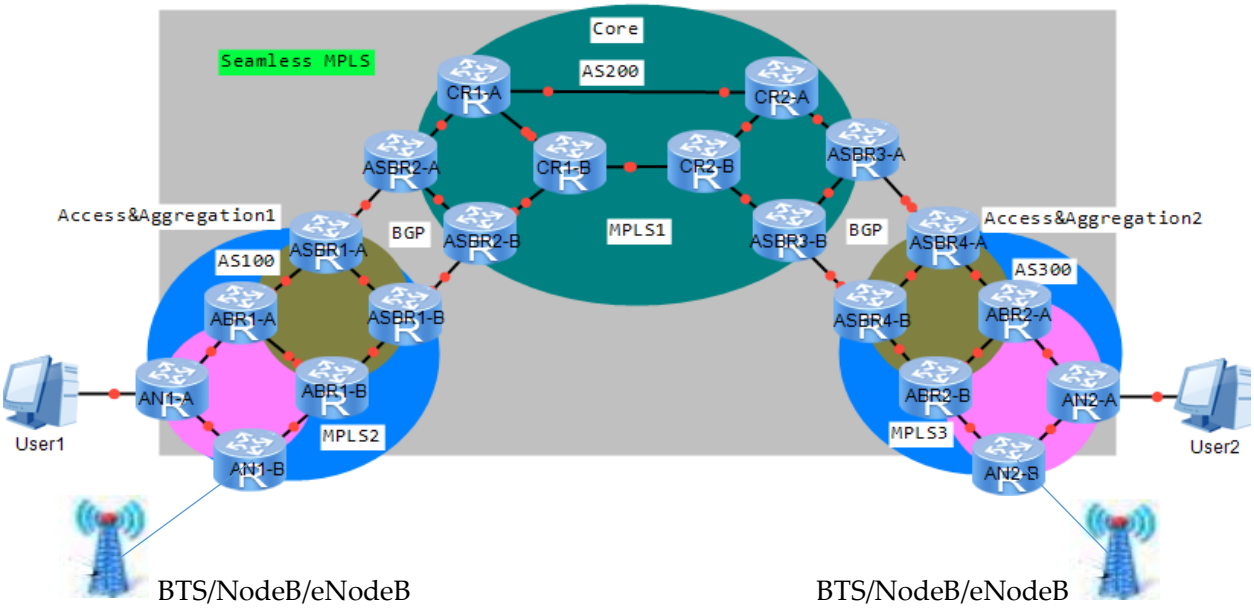

Figure 3.3: Inter-AS seamless MPLS architecture

3.4. Key Technologies Supported in Seamless MPLS

a. BGP Labeled Unicast (BGP-LU)

The Seamless MPLS architecture must scale to support end to-end MPLS in a network on the order of 100,000 nodes. The key to scaling is to create hierarchical end-to-end LSPs by distributing the right amount of routing intelligence with BGP-LU. Hierarchy is created by segmenting the network into regions, running closed IGP within the regions, and restricting inter-region IGP communication. All inter-region control plane information is shared via BGP-LU. When BGP is used to distribute a particular route, it can also be used to distribute an MPLS label which is mapped to that route.

[8] Specifies encodings and procedures for using BGP to indicate that a particular router has bound either a single MPLS label or a sequence of MPLS labels to a particular address prefix. This is done by sending a BGP UPDATE message whose Network Layer Reachability Information (NLRI) field contains both the prefix and the MPLS label(s). Each such UPDATE also advertises a path to the specified prefix via the specified next hop. When we enable RFC 8277 (BGP-LU) on both BGP routers, the routers advertise to each other that they can then send MPLS labels with the routes. If the routers successfully negotiate their ability to send MPLS labels, the routers add MPLS labels to all outgoing BGP updates. The label exchange is needed in order to keep the end–to–end path information between segments. As a result, each segment becomes small enough to be managed by operators and at the same time there is circuit information distributed for path awareness between two different IP speakers. BGP 8277 inserts one extra label in the forwarding label stack in the Seamless MPLS architecture as shown in Figure 3.4.

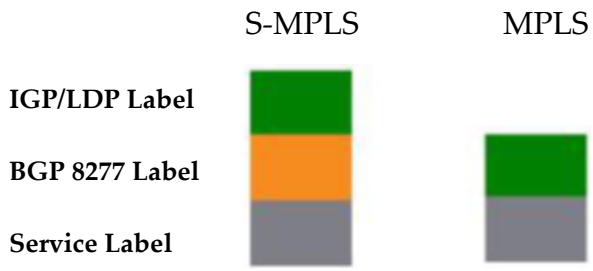

Figure 3.4: Label stack in seamless MPLS architecture

b. LDP DoD

In addition to the use of BGP-LU to create LSP hierarchy, to keep the access nodes cost-effective and functionally simple but still operationally intelligent, LDP DoD [24] with static routes, limiting number of IP Routing Information Base (IP RIB) and IP Forwarding Information Base entries required on the access node can be used. Given that the simplest access router infrastructure may only use static routes, LDP DoD enables on-request label distribution ensuring that only required labels are requested, provided and installed.

The general MPLS use of LDP DU (Downstream Unsolicited) advertises labels for all routes in the routing table [31]. Establishing a large number of LSPs burdens an LSR such as a digital subscriber line access multiplexer (DSLAM) that is a low-performance access device deployed on an MPLS network. On a large-scale network, a DSLAM can be configured to send label mapping messages to only upstream LSRs only after receiving requests for labels. This minimizes the number of unwanted MPLS forwarding entries forwarded by the DSLAM [9, 10].

c. Route Reflector (RR)

Typically, all BGP speakers within a single AS must be fully meshed and any external routing information must be re-distributed to all other routers within that AS. For n BGP speakers within an AS that requires to maintain n*(n-1)/2 unique iBGP sessions. This full-mesh requirement clearly does not scale when there are a large number of iBGP speakers each exchanging a large volume of routing information.

One means of alleviating the need for a full-mesh is using Route Reflector. This approach allows a BGP speaker (Route Reflector) to advertise iBGP learned routes to certain iBGP peers [32]. If area border routers are made Route Reflectors, the number of iBGP peering is reduced to the number of BGP speakers per segment instead of between all BGP speakers of the complete AS. If a set of BGP speakers are exchanging routes via a Route Reflector, then by piggybacking the label distribution on the route distribution, one is able to use the Route Reflector to distribute the labels as well. This improves scalability quite significantly. Note that if the Route Reflector is not in the forwarding path, it need not even be capable of forwarding MPLS packets.

d. Next–Hop–Self

BGP operates on the base of recursive routing lookups. This is done in order to accommodate scalability within the underlying IGP that is utilized. For the recursive lookup, BGP uses Next–Hop attached to each BGP route entry. Thus, for example, if a source node desires to send a packet to a destination node and if the packet hits the BGP router, then the BGP router does a routing lookup in its BGP routing table. It finds a route toward destination node and finds the Next–Hop as a next step. This Next–Hop must

be known by the underlying IGP. As the final step, the BGP router forwards the packet onwards based upon the IP and MPLS label information attached to that Next−Hop.
In order to make sure that within each segment only the Next−Hops are needed to be known by the IGP, it is needed that the Next−Hop attached to the BGP entry is within the network segment and not within a neighbor or further away segment. If we rewrite the BGP Next−Hop with the Next−Hop−Self feature, ensure that the Next−Hop is within the local segment [33].

e. End-to-end Resiliency

Seamless MPLS provides end-to-end resiliency at the transport and service layers. The framework supports Pseudowire (PW) redundancy at the service layer. The transport layer supports protection of the inter-region transport tunnel (BGP tunnel), as well as the intra-region (LDP or RSVP tunnel) transport tunnel. During failures, this ensures local fast protection (i.e., LDP fast reroute (FRR), RSVP FRR or BGP anycast) is initiated while end-to-end protocol convergence occurs which eventually results in new set of BGP transport tunnels being created end-to-end [10].

4. IP Quality of Service

In [34] QoS is defined as the manipulation of traffic such that the network device forwards it in a fashion consistent with the required behaviors of the applications generating that traffic. It can also be defined as the ability to provide performance guarantee in the network [35]. QoS enables a network device to differentiate traffic and then apply different behaviors to that different traffic. To provide QoS solutions, network devices such as routers and switches differentiate the traffic by examining the packets as they enter the device and then classifying the traffic into groups, called classes of service. QoS behaviors tell the device how to treat the traffic as it travels from ingress interface all the way until it is sent out the egress interface of the network device. The result is that we can treat traffic assigned to any one class of service differently from any other class of service, and in any manner we want in order to provide the desired QoS solution. A network device can apply QoS behaviors to the traffic in each class of service depending on the QoS capabilities and how the device is configured [34]:

- Prioritizing traffic over other traffic based on type of protocol, a source or destination address, or a source or destination port number
- Filtering traffic upon ingress or egress
- Controlling the allowed bandwidth transmitted or received on the interfaces of the device
- Applying QoS behavior requirements in the packet header
- Controlling congestion and packet loss

4.1. QoS Parameters

QoS is a traffic management strategy that allows allocating network resources based on the traffic characteristics. These traffic characteristics must be controlled and managed on a hop by hop basis in order to achieve the QoS needed by the traffic. The core QoS parameters which influence the traffic in IP network are: Throughput, Delay, Jitter and Packet Loss [36, 37]. The factors affecting QoS parameters and the implementation of performance recommendations are shown in the below Figure 4.1.

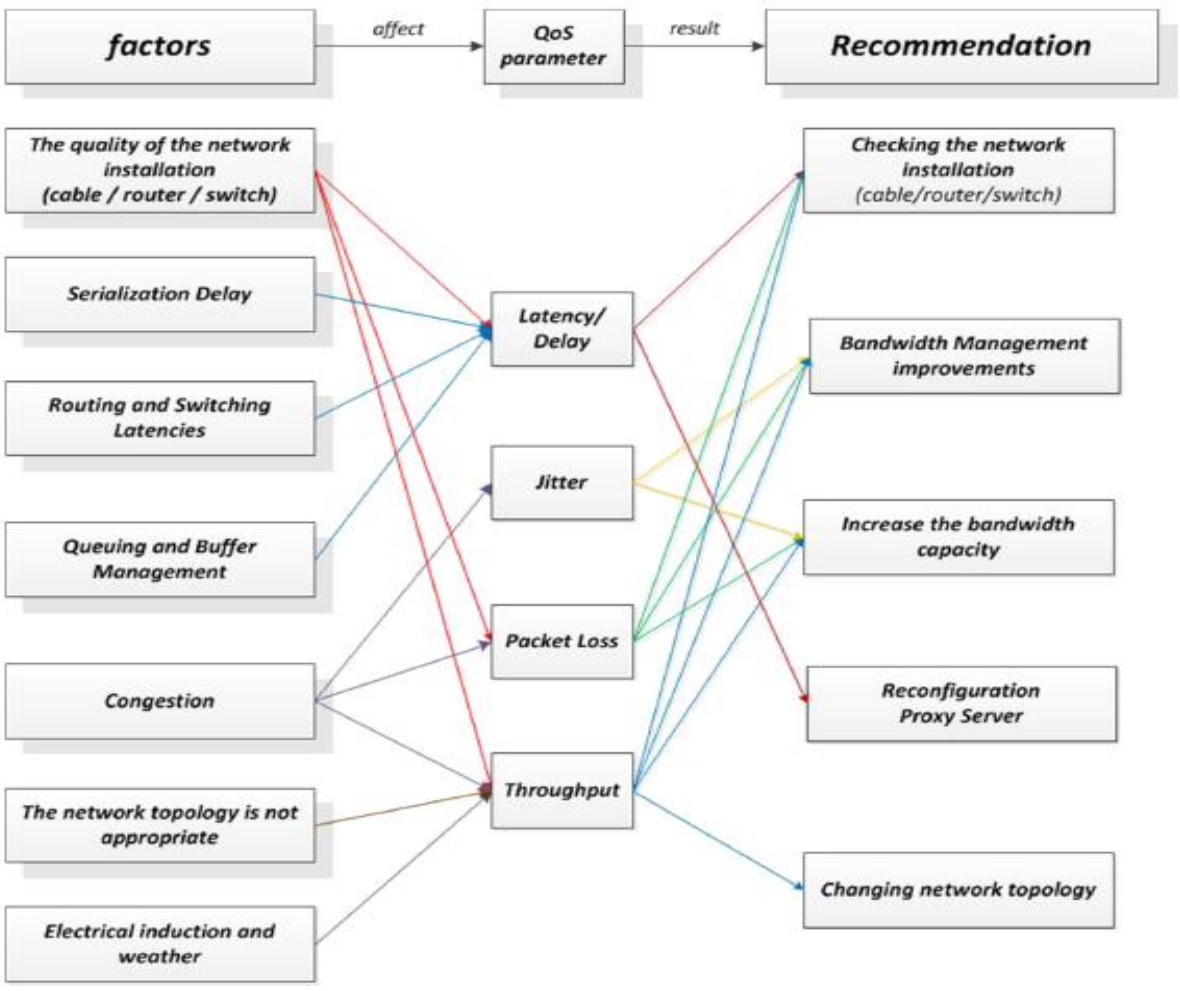

Figure 4.1: Recommendations for QoS parameters [35]

4.1.1. Throughput

Throughput is a measure of how many units of information a system can process in a given amount of time. Given the dynamic nature of traffic flow across a network, different resources can become bottlenecks at different times. Some of the factors that determine the bandwidth and throughput are: network devices, network topology, and number of network users, etc. The formula for calculating throughput value is shown in Equation (1).

$$\text{Throughput} = \frac{\sum sent\ data\ (bit)}{time\ data\ delivery\ (s)} [bps] \ldots (1)$$

Table 4.1: Quality standards for throughput [35]

	Category	Throughput/Bandwidth
Throughput standard	Excellent	100%
	Good	75%
	Medium	50%
	Poor	<25%

4.1.2. Delay

RFC 7679 defines a metric for measuring one-way delay as the difference in the time at which the datagram crosses two reference points. The delay of a datagram experienced within a service provider network is defined as the difference in the time at which the datagram enters the network and the time at which it leaves the network. It is also commonly referred to as latency. Each element through which a datagram flows in a traffic path will increase the delay experienced by the datagram. From a SLA perspective, the delay is the average fixed delay that an application's traffic will experience within the service provider's network. Delay in TCP/IP networks can be classified as

packetization delay, queuing delay, propagation delay, transmission delay and processing delay [35]. The formula for transmission delay is as shown in Equation (2).

$$\text{Delay} = \frac{\text{packet length (bit)}}{\text{link bandwidth (bit/s)}} [\text{second}] \dots (2)$$

Table 4.2: Quality standards ITU-T G.114 for delay

	Category	Delay (ms)
Delay standard	Good	0-150
	Medium	150-400
	Poor	>400

4.1.3. Jitter

RFC 3393 has defined a metric for measuring one-way jitter. Jitter is the variation in the network delay experienced by datagrams. More specifically, it is measured as the delay variation between two consecutive datagrams belonging to a traffic stream. In order to avoid dropping datagrams when a resource is temporarily congested, buffer space is made available in network nodes and the datagrams are queued. Queuing within a network node introduces delay variation between different datagrams of a traffic stream. Although queuing is the main cause of traffic jitter, lengthy reroute propagation delays and additional processing delays can also affect traffic jitter. The formula for calculating jitter value is shown in Equation (3).

$$\text{Jitter} = \frac{\sum variation\ delay}{\sum packet\ received} [second] \dots (3)$$

Table 4.3: Quality standards ITU-T G.114 for jitter

	Category	Jitter (ms)
Jitter standard	Good	0 - 20ms
	Medium	20 - 50ms
	Poor	>50ms

4.1.4. Packet Loss

Traffic loss characterizes the datagram drops that occur in the path of a one-way traffic flow between source and destination node. Having buffer space to temporarily queue datagrams in network nodes helps reduce datagram loss, but it cannot be completely eliminated. Some of the factors that contribute to datagram loss are [35, 36]:

- **Congestion** - Bursty traffic can cause queue overflows resulting in datagram loss.
- **Traffic rate limiting** - In order to ensure customer traffic is conforming to a negotiated SLA, service providers may rate-limit incoming traffic and drop nonconforming datagrams.
- **Physical layer errors** - Noise in physical layers can cause bit errors. As a result, upper-layer protocols may drop datagrams.
- **Network element failures**—Network element failures may cause datagrams to drop until the failure is detected and the connectivity is restored. The formula for calculating the percentage of packet loss value is shown in Equation (4).

$$\text{Packet loss} = \frac{packets\ sent - packets\ received}{packets\ sent} \times 100\% \quad \ldots (4)$$

Table 4.4: Quality standards for packet loss [35]

	Category	Packet loss
Packet loss standard	Excellent	0%
	Good	3%
	Medium	15%
	Poor	25%

4.2. QoS Models

To manage the loss, latency, and jitter in today's networks, IETF defined two models relevant to QoS in IP packet-based networks [38].

- Integrated Services,
- Differentiated Services

4.2.1. Integrated Services (IntServ)

In the IntServ QoS model, hosts or routers using RSVP can specify the resource requirement or quality of service required for the end-to-end path for each individual flow or data stream. Each node in the path that receives the RSVP message checks to see if it has sufficient resources to accept the flow. If the check fails, an error notification is sent to the sender that originated the RSVP. If the RSVP signaling is successful, then each node in the path makes the requested reservation for the connection and the data transmission begins.

4.2.2. Differentiated Services (DiffServ)

Differentiated Services is based on an architecture [RFC 2475] that pushes complex decision making to the edge routers. This results in less processing load on core routers and, thus, faster operation, due to less signal state processing and storage. According to this architecture, a differentiated services code point (DSCP) is carried in every packet. This is carried in the IP type of service (ToS) field. Classification, rate shaping, and policing are done at the edge routers and packets are mapped onto service levels. Per-hop queuing and scheduling behaviors (PHBs) are defined through which a number of edge-to-edge services might be built [34, 39]. The DiffServ model has the following core functions performed within the network devices. These are:

- Classifiers
- Policers
- Shapers
- Queues
- Schedulers and Remarker

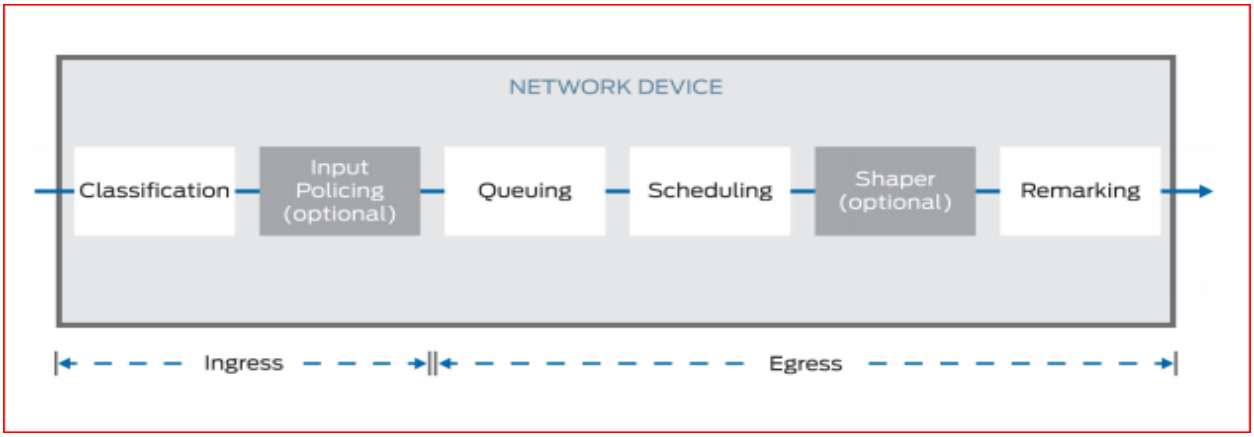

Figure 4.2: DiffServ model QoS functions [34]

The network device receives packets on the ingress interface; it classifies the packets into the appropriate class. If there is an optional policer configured, it rate-limits the traffic or assigns the traffic to a different class. The scheduler takes the packets out of the queues and transmits them in the order configured for the scheduler. If there is a shaper configured, it shapes the traffic to the configured shaping-rate. Lastly, if remarking is configured, the device remarks the value of the DS-field so that the next device to receive the packet knows how to classify it.

5. Simulation Results and Analysis

This chapter aims to present a brief introduction of simulation tools used for quality of service analysis. Then simulation scenarios and network topologies used for the analysis follows. At the end, simulation results and analysis of the results is discussed.

5.1. Overview of Simulation Tools

5.1.1. Enterprise Network Simulation Platform (eNSP)

eNSP is a free, extensible, graphical network simulation tool platform provided by Huawei. Mainly it is used for simulation of functions and features of routers, switches, personal computers, hubs, clouds, device configuration, packet capture on interfaces, etc. Also it has a capability of connecting real device to simulation tool via network card. Its support for large-scale network simulation and ability to simultaneously support both single-server environment and multi-server environment enables us to use the tool for experimental test and analysis in the case of that there is no real device [40].

5.1.2. Network Quality Analyzer (NQA)

NQA analyzes network performance and service quality by sending test packets, providing us with network performance parameters such as jitter, TCP connection delay, FTP connection delay, file transfer rate, packet loss ratio, etc. NQA monitors network QoS in real time and locates and diagnoses network faults. It requires two test ends, an NQA client and an NQA server (or called the source and destination). The NQA client (or the source) initiates an NQA test. Test instances can be configured through command lines or NMS (Network Management System). Then NQA places the

test instances into test queues for scheduling. NQA supports different test types: ICMP-echo, DHCP, DNS, FTP, HTTP, UDP jitter, SNMP, TCP, UDP echo, voice, etc. [41].

5.1.3. Ostinato

Ostinato is a packet crafter, network traffic generator and analyzer with a friendly GUI. Craft and send packets of several streams with different protocols at different rates. The common standard protocols supported by Ostinato are Ethernet/802.3/LLC SNAP; ARP, IPv4, IPv6, IP-in-IP, IP Tunneling (6over4, 4over6, 4over4, 6over6); TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD and many text based protocols like HTTP, SIP, RTSP, NNTP etc. It also supports client server architecture. It can create and configure sequential and interleaved streams of different protocols at different rates. Flexibility to add any unimplemented protocol is also provided through a user defined script [42].

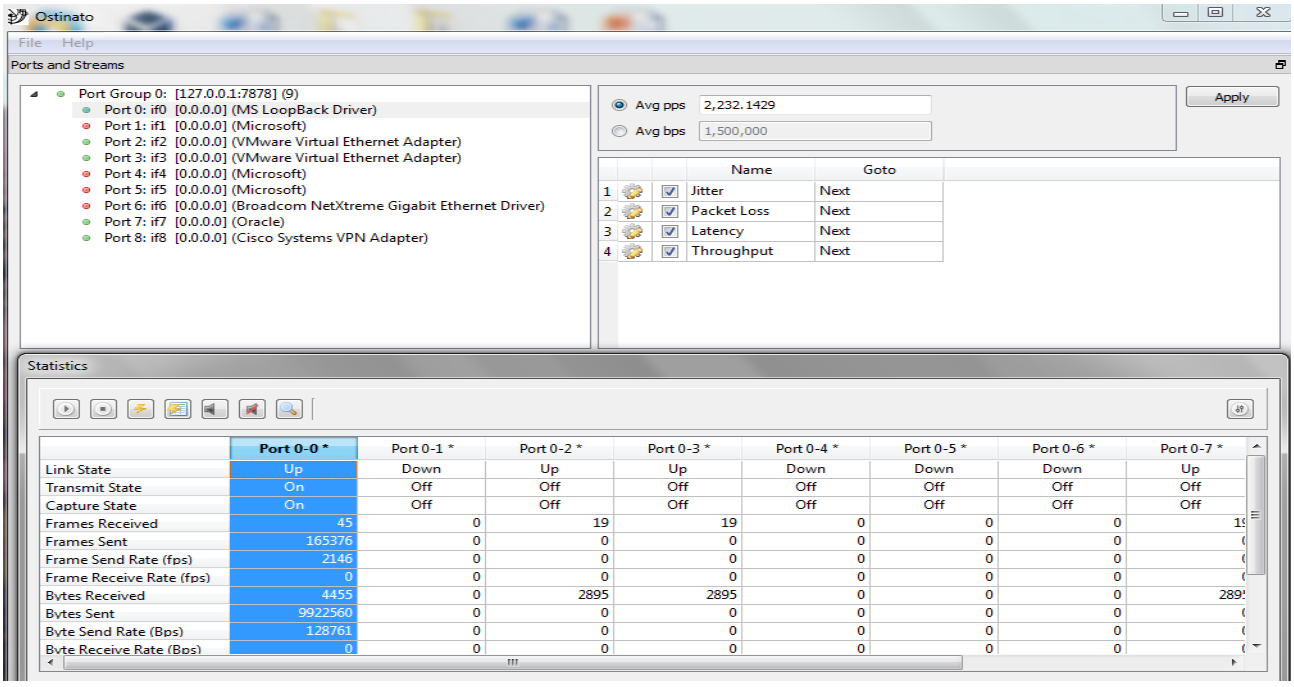

Figure 5.1: GUI for ostinato

5.2. Simulation Scenarios and Network Topology

In the implementation part, a practical environment is developed using enterprise network simulation platform and two scenarios are built in order to test the performance of MPLS and Seamless MPLS. The two scenarios are built in such a way that first an ordinary network is built with classical MPLS. Then the same network topology is implemented with Seamless MPLS technologies. In both cases Ostinato is used for generating network traffic in order to test the performance of the networks with respect to four QoS metrics. Two network traffic generators are implemented in such a way that the first traffic generator injects the required amount of traffic in to the network for end-to-end performance analysis while the second network traffic generator injects random network traffic. This random traffic is not directly used for testing and analysis purpose rather it is to create competition for resources among the network traffics. Due to memory limitations of personal computers and the process intensiveness of the simulation tools used, few redundant nodes and some redundant links are used at the core domains. These approaches make our simulation environment resemble a real network. Finally, the test results are collected from the simulator using NQA technology for the two scenarios. Figure 5.2 and 5.3 show MPLS architecture (scenario 1) and Seamless MPLS architecture (scenario 2) respectively. In both scenarios three MPLS domains are used where access and aggregation regions are in one MPLS domain and the core network is in another MPLS domain. These topologies are representative of today's MPLS architecture supporting any type of network traffic end-to-end. Note that Figure 5.2 and 5.3 are used for the subsequent QoS parameters analysis.

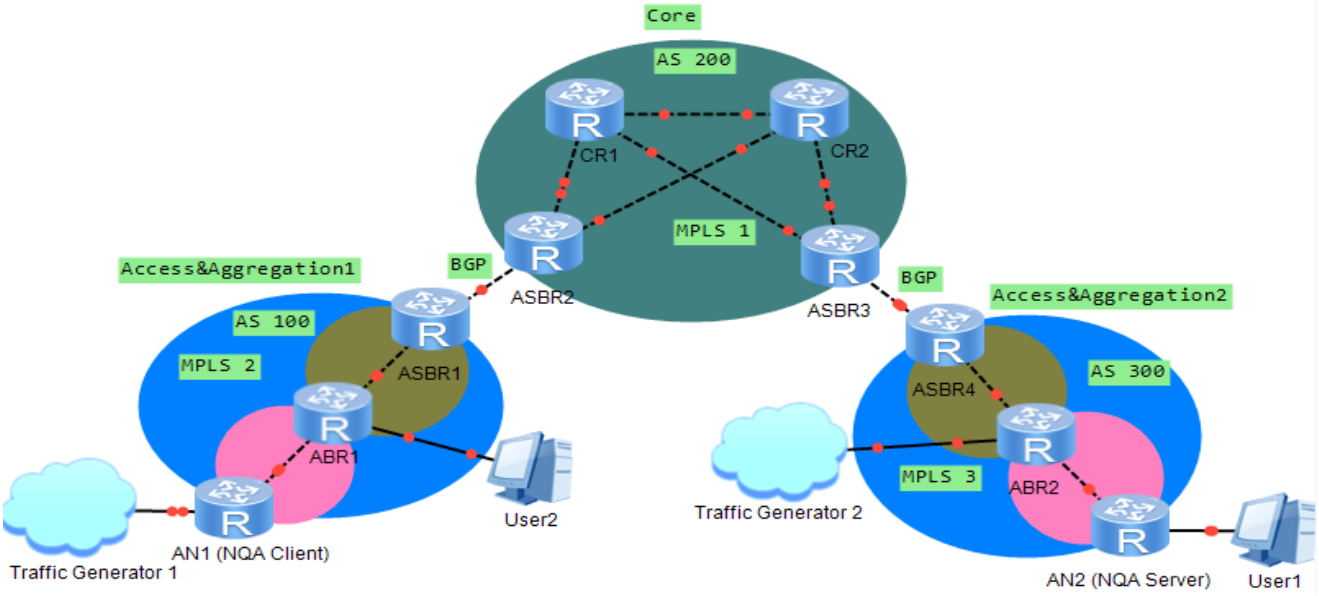

Figure 5.2: MPLS architecture

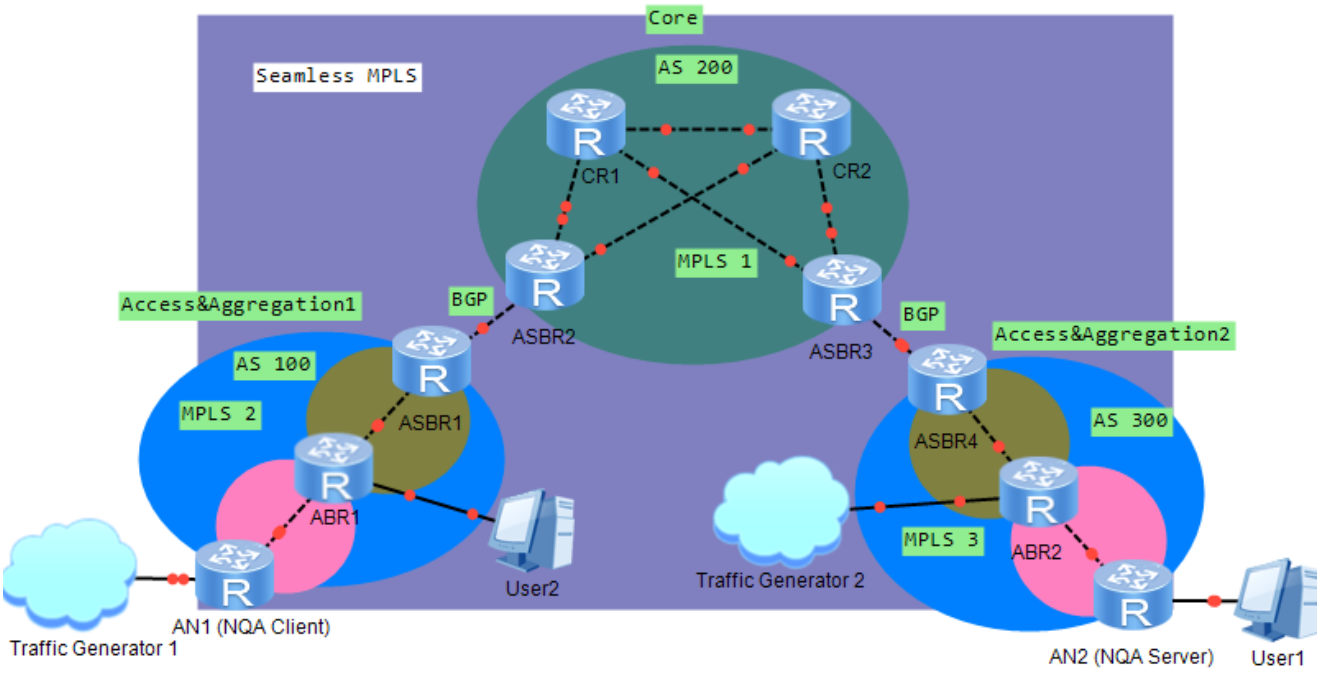

Figure 5.3: Seamless MPLS architecture

5.3. Simulation Parameters Analysis

5.3.1. Throughput Analysis

Throughput is one of the QoS parameters which measures how many units of information a system can process in a given amount of time. To compute the throughput we need to know the amount of data transferred and how long it takes to complete the file transfer. Traffic generator 1 is used to generate traffic at a rate of 1.5Mbps into the network for both scenarios (MPLS and S-MPLS). AN1 and AN2 are used as NQA-agent/client and NQA-Server respectively and file is downloaded from the server to the client at different file sizes using FTP. During the simulation, NQA collects relevant information including average round trip time required to complete the file download for each file size. Figure 5.4 shows sample output of NQA for FTP file size of 950 bytes for scenario 1. From this output, throughput is computed as the ratio of file downloaded to the average of round trip time i.e. (950bytes*8bits/byte)/14.850sec = 511.78Kbps.

```
[AN1]display nqa results test-instance ADMINISTRATOR ftp

 NQA entry(ADMINISTRATOR, ftp) :testflag is inactive ,testtype is ftp
  1 . Test 1 result   The test is finished
    SendProbe:1                          ResponseProbe:1
    Completion:success                   RTD OverThresholds number: 0
    MessageBodyOctetsSum: 1038848        Stats errors number: 0
    Operation timeout number: 0          System busy operation number:0
    Drop operation number:0              Disconnect operation number: 0
    CtrlConnTime Min/Max/Average: 360/360/360
    DataConnTime Min/Max/Average: 14490/14490/14490
    SumTime Min/Max/Average: 14850/14850/14850
    Average RTT:14850
    Lost packet ratio:0 %
[AN1]
```

Figure 5.4: Sample output of test result (scenario 1)

For different file sizes the test results are tabulated for both scenarios as shown in Table 5.1.

Table 5.1: Throughput for MPLS and seamless MPLS at different file sizes

File size (Kbyte)	100	200	300	500	700	800	900	1000
Average RTT (Sec) for MPLS	1.79	3.22	4.51	7.26	10.61	12.19	13.67	16.52
Average RTT (Sec) for SMPLS	1.61	2.67	3.85	6.03	8.16	9.51	10.44	12.07
Throughput MPLS (Kbps)	446.93	496.89	532.15	550.96	527.80	525.02	526.70	484.26
Throughput SMPLS(Kbps)	496.89	599.25	623.78	663.35	686.27	672.98	689.66	662.80

Figure 5.5 shows the graph of simulation results of throughput versus FTP file size while downloading file from server (AN2) to client (AN1). As can be seen in the throughput graph, at small FTP file size there is no significant difference in the performances of both scenarios but as the FTP file size increases the performance of Seamless MPLS is better than that of classical MPLS. For example, if we take file size of 1000Kbytes for comparison, the throughput difference is about 178.54Kbps which is 36.87%. There are different reasons for the better throughput of Seamless MPLS over MPLS. As we have discussed in the previous chapters, for the inter-domain (inter-AS) routing MPLS uses BGP for advertising the reachability information between peer routers. Due to inherent nature and the protocols used in BGP, it requires additional processing time compared to Seamless MPLS. For example, BGP uses TCP port 179 for reliable communication but TCP has an additional overhead (about 40 bytes per packet) for circuit establishment such as 3-way handshake before transferring the actual data, hence reducing throughput. The sliding window used in TCP and the retransmission of

packets due to packet drops can limit data transmission rate hence they affect the throughput. In BGP, each router in the network has to make independent routing decisions for each incoming packet. When a packet arrives at a router, the packet is stored in data plane of router. Each port of router is in its data plane. Now first layer 2 processing will be done on packet to check whether the packet is destined for that particular MAC of router. If yes then now layer 3 processing of packet is performed. Layer 3 process will check routing table, which is in control plane, the router to find the next hop for that packet based on the packets destination address in the packets IP header (longest match prefix lookup). For entire decision making process, there will be transfer of processing from control plane to data plane many times. So this is time consuming process. Also IP routing is performed at each hop of the packets path in the network. Entire IP header analysis is done at each hop between ASes which is time consuming.

In Seamless MPLS all forwarding of packets within a network, from the time a packet enters the network until it leaves the network, is based on MPLS labels [BGP-LU]. The key scaling in Seamless MPLS is to create hierarchical end-to-end LSPs by distributing the right amount of routing intelligence with BGP-LU. Hierarchy is created by segmenting the network into regions, running closed IGP within the regions, and restricting inter-region IGP communication. All inter-region control plane information is shared via BGP-LU. The absences of detailed IP address lookup and routing intelligence have resulted in better throughput of Seamless MPLS. The main difference between a lookup in the routing table of BGP and the MPLS LFIB is that the routing table lookup is concerned with longest prefix match, i.e. having potentially many matches and selecting the one that most closely resembles the destination IP address. On the other hand, the

MPLS LFIB used in BGP-LU always performs lookups on fixed-length values and with equality operation, not with prefix-based logic. Hence, a routing table lookup is algorithmically more complex than a lookup in the LFIB, as finding a longest prefix match is more computationally intensive than simply finding a single matching value. Therefore the LFIB lookups is faster than BGP lookups.

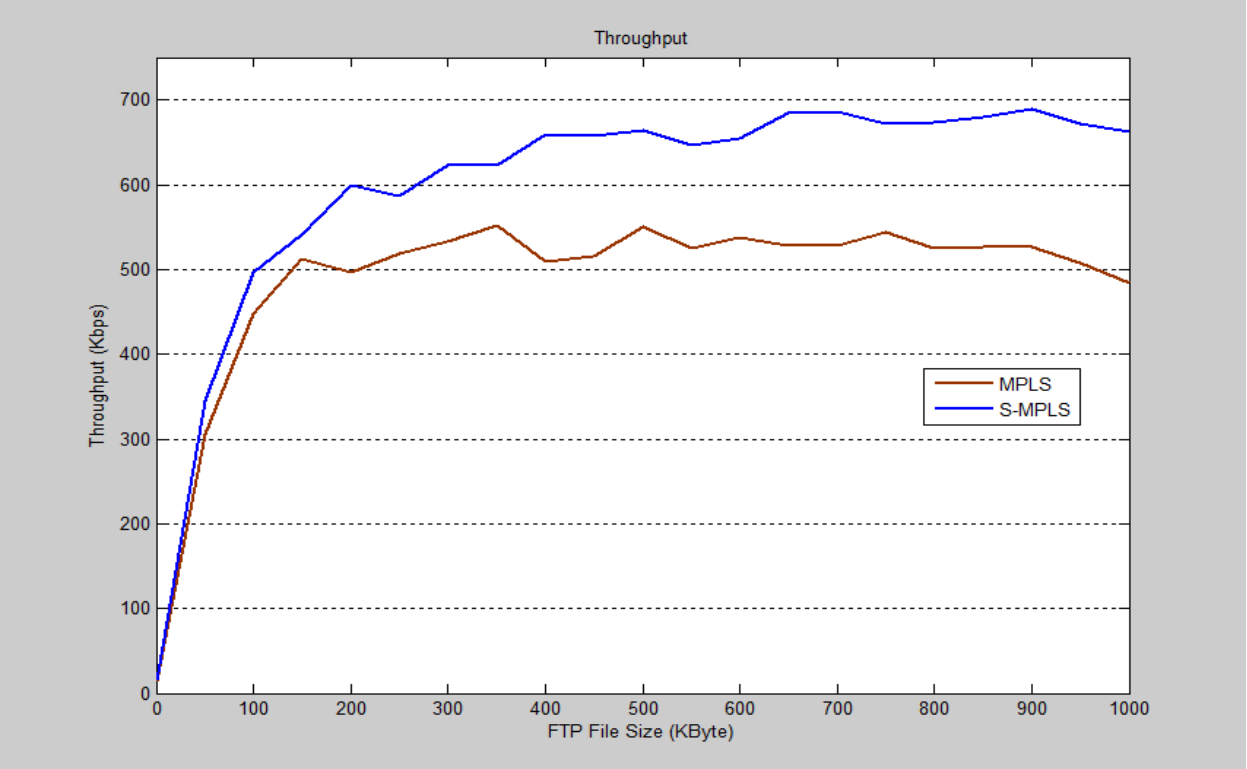

Figure 5.5: Graph of throughput for scenarios 1 & 2

5.3.2. Latency Analysis

The ITU-T recommendation for maximum end-to-end latency is 150ms. Although the same topology setup as in the throughput analysis is used for latency analysis, the test instances implemented on both server and client are different (refer Appendix A1&A2). The ICMP (Internet Control Message Protocol) ping test type is used to send test probe

to collect the average completion time of each test probe at different data sizes as shown in Table 5.2. Network traffic is generated end-to-end and NQA collects the minimum, maximum and average delay of sending test message from AN1 to AN2 and vice versa. A sample simulation output of sending 20 byte test message using scenario 2 is shown in Figure 5.6.

```
[AN1-nqa-ADMINISTRATOR-icmp]display nqa results test-instance ADMINISTRATOR icmp


 NQA entry(ADMINISTRATOR, icmp) :testflag is inactive ,testtype is icmp
  1 . Test 1 result   The test is finished
   Send operation times: 5              Receive response times: 5
   Completion:success                   RTD OverThresholds number: 0
   Attempts number:1                    Drop operation number:0
   Disconnect operation number:0        Operation timeout number:0
   System busy operation number:0       Connection fail number:0
   Operation sequence errors number:0   RTT Status errors number:0
   Destination ip address:9.9.9.9
   Min/Max/Average Completion Time: 50/70/60
   Sum/Square-Sum  Completion Time: 300/18200
   Last Good Probe Time: 2018-06-20 09:17:50.6
   Lost packet ratio: 0 %
[AN1-nqa-ADMINISTRATOR-icmp]
```

Figure 5.6: Sample output of latency test (scenario 2)

To increase the accuracy of the values for each of the latencies, the average of about 15 to 20 sample tests are used. For the simulation three congestion levels are considered: data rate less than link bandwidth, data rate equal to link bandwidth and data rate greater than link bandwidth. For the different congestion levels average values of simulation results are tabulated in Table 5.2.

Table 5.2: Output of latency for scenarios 1 & 2

Datasize (bytes)	20	1000	2000	3000	4000	5000	6000	7000	8000
Ave. Completion Time (ms) for MPLS	64	74	84	93	109	120	134	152	169
Ave. Completion Time (ms) for SMPLS	60	66	77	87	96	108	119	129	142

The latency versus data size graph in Figure 5.7 shows that the classical MPLS (Scenario 1) has higher latency than Seamless MPLS (Scenario 2). At small data size (uncongested level) the latency difference is smaller compared to higher data size (congested level). For example, at 8000bytes the latency of Seamless MPLS is improved by 27ms (i.e. 15.98%) compared to the MPLS counterpart. This is a significant improvement because a 1ms decrease in latency will increase data rate by 1000bps. Had it been on physical network in real scenario, the improvement in latency would be higher than this value because some factors like propagation delay has insignificant effect on the total latency in this simulation. Typically, all BGP speakers within a single AS must be fully meshed and any external routing information must be re-distributed to all other routers within that AS. This full mesh requirement clearly does not scale when there are IBGP speakers each exchanging a large volume of routing information. One means of alleviating the need for a full-mesh is using route reflector [RR]. This approach allows route reflector to advertise iBGP learned routes to certain iBGP peers and reduce processing delay in Seamless MPLS. Longest prefix match table lookups in BGP compared to exact prefix label match in Seamless MPLS gives rise to higher latency in inter-domain routing.

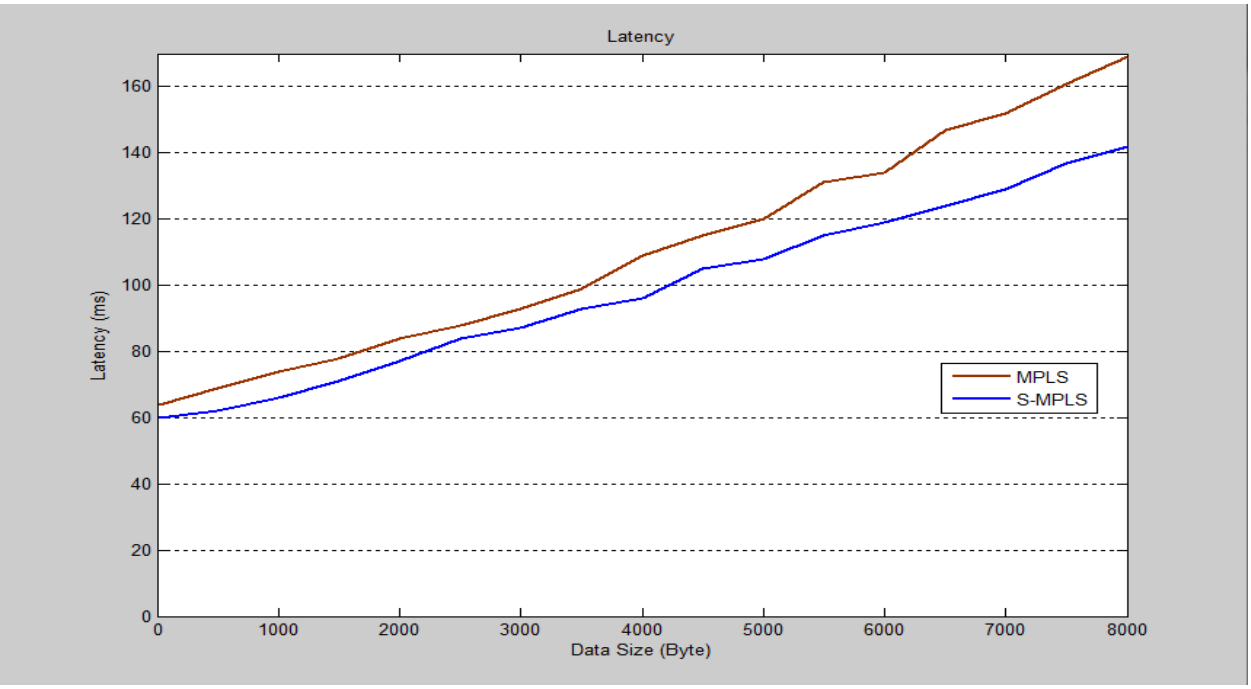

Figure 5.7: Graph of latency for scenarios 1 & 2

5.3.3. Packet Loss Analysis

As already discussed in Section 4.1.4 some of the factors contributing for packet loss are: congestion, traffic rate limiting, physical link errors and network element failures. In ITU standards the recommended value for packet loss is less than 3%. For this packet loss analysis, congestion is selected to be a factor for packet loss i.e. the links in the network are deliberately congested by injecting more network traffic into the network using the network traffic generators. This enables us to compare the tolerance of both network scenarios towards congestion. ICMP test type is used to send test probes at three conditions i.e. data rate less than link bandwidth, data rate equal to link bandwidth and data rate greater than link bandwidth. A sample snapshot of simulation result of a congested link with data rate greater than the bandwidth has a packet loss ratio of 20% as shown in Figure 5.8.

```
[AN1-nqa-ADMINISTRATOR-icmp]display nqa results

 NQA entry(ADMINISTRATOR, icmp) :testflag is inactive ,testtype is icmp
  1 . Test 1 result   The test is finished
    Send operation times: 5              Receive response times: 4
    Completion:success                   RTD OverThresholds number: 0
    Attempts number:1                    Drop operation number:0
    Disconnect operation number:0        Operation timeout number:1
    System busy operation number:0       Connection fail number:0
    Operation sequence errors number:0   RTT Status errors number:0
    Destination ip address:9.9.9.9
    Min/Max/Average Completion Time: 140/820/322
    Sum/Square-Sum  Completion Time: 1290/746900
    Last Good Probe Time: 2018-06-20 09:46:51.0
    Lost packet ratio: 20 %
[AN1-nqa-ADMINISTRATOR-icmp]
```

Figure 5.8: Packet loss sample output

As shown in Table 5.3 and Figure 5.9, there is no packet loss in both scenarios when the links are not congested but packets in Scenario 1(MPLS) start to drop prior to Scenario 2 (Seamless MPLS) and in both scenarios the packet drop increases very fast when the links are being congested. The performance of Seamless MPLS is much better than that of MPLS. For example, at data size of 8000 bytes there is 20% less packet loss in S-MPLS. The general MPLS use of LDP DU (Downstream Unsolicited) advertises labels for all routes in the routing table. Establishing a large number of LSPs burdens an LSR deployed on an MPLS network. To reduce this burden and reduce packet loss, Seamless MPLS uses LDP DoD to keep routers cost-effective and functionally simple but still operationally intelligent by limiting number of IP Routing Information Base (IP RIB) and IP Forwarding Information Base entries required on the routers.

Table 5.3: Output of packet loss for scenarios 1 & 2

Data Size (bytes)	20	1000	2000	3000	4000	5000	6000	7000	8000
Packet Loss (%) for MPLS	0	0	0	1	3	9	18	35	50
Packet Loss (%) for SMPLS	0	0	0	0	0	1	4	15	30

The other method by which S-MPLS minimizes the effects of congestion and link failure is by using fast reroute and pre-computed alternative routes. These mechanisms help to use alternative paths and nodes in a sub-second to reduce packet loss.

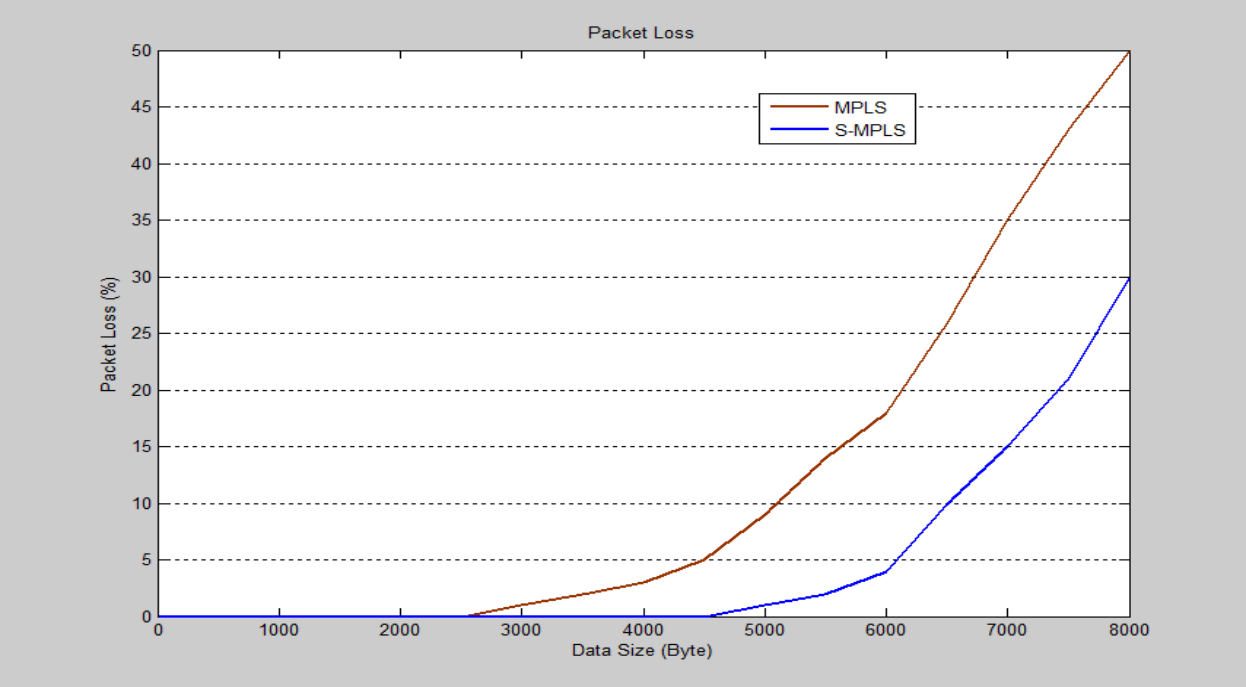

Figure 5.9: Graph of packet loss for scenarios 1 & 2

5.3.4. Jitter Analysis

Recall that Jitter is a variation in the delay of received packets. At the transmitting side, packets are sent in a continuous stream with the packets spaced evenly apart. Due to network congestion, improper queuing, or configuration errors, the delay between each packet can vary instead of remaining constant. In IP data networks packets can take different alternative routes to a destination and may arrive at different time and this causes variation of delay or jitter.

Using the same procedures and scenarios as for the other parameters, we can simulate and measure the jitter values. In our network topology there are redundant and alternative routes to a destination so that packets of one stream can take different alternative routes with variable delay. Since network congestion is a common factor for jitter, our simulation has mainly considered it for jitter analysis. Though the test results can provide us with average jitter from source node, AN1, to destination node, AN2, independently, we have used the average jitter for the round trip path to compare the performance of the scenarios as shown in Figure 5.10.

```
[AN1-nqa-ADMINISTRATOR-ICMPJITTER]display nqa results

 NQA entry(ADMINISTRATOR, ICMPJITTER) :testflag is inactive ,testtype is jitter
  1 . Test 1 result   The test is finished
   SendProbe:60                          ResponseProbe:59
   Completion:success                    RTD OverThresholds number:0
   OWD OverThresholds SD number:0        OWD OverThresholds DS number:0
   Min/Max/Avg/Sum RTT:40/70/56/3280     RTT Square Sum:186200
   NumOfRTT:59                           Drop operation number:0
   Operation sequence errors number:0    RTT Stats errors number:0
   System busy operation number:0        Operation timeout number:1
   Min Positive SD:10                    Min Positive DS:10
   Max Positive SD:20                    Max Positive DS:20
   Positive SD Number:12                 Positive DS Number:9
   Positive SD Sum:160                   Positive DS Sum:110
   Positive SD Square Sum:2400           Positive DS Square Sum:1500
   Min Negative SD:10                    Min Negative DS:10
   Max Negative SD:20                    Max Negative DS:20
   Negative SD Number:16                 Negative DS Number:10
   Negative SD Sum:170                   Negative DS Sum:120
   Negative SD Square Sum:1900           Negative DS Square Sum:1600
   Min Delay SD:20                       Min Delay DS:19
   Avg Delay SD:27                       Avg Delay DS:26
   Max Delay SD:35                       Max Delay DS:34
   Delay SD Square Sum:46550             Delay DS Square Sum:43329
   Packet Loss SD:0                      Packet Loss DS:0
   Packet Loss Unknown:0                 Average of Jitter:11
   Average of Jitter SD:11               Average of Jitter DS:12
   Jitter out value:4.0632744            Jitter in value:2.6011271
   NumberOfOWD:59                        Packet Loss Ratio: 1%
   OWD SD Sum:1640                       OWD DS Sum:1581
   ICPIF value: 0                        MOS-CQ value: 0
   TimeStamp unit: ms                    Packet Rewrite Number: 0
   Packet Rewrite Ratio: 0%              Packet Disorder Number: 0
   Packet Disorder Ratio: 0%             Fragment-disorder Number: 0
   Fragment-disorder Ratio: 0%           Jitter OverThresholds SD number:0
   Jitter OverThresholds DS number:0     OverallOverThresholds number:0
[AN1-nqa-ADMINISTRATOR-ICMPJITTER]
```

Figure 5.10: Sample average jitter output

As shown in Figure 5.11, the simulation results of average jitter for Seamless MPLS is smaller than that of multi-domain MPLS. Take data size of 8000 bytes for example, the jitter difference is about 3ms (i.e 12.5%). This performance difference between the two scenarios has significant impact on jitter sensitive real time traffic such as voice, video conference, live streaming, etc. Unlike BGP which maintain one best route to peers in

routing table, a route reflector in Seamless MPLS maintain (and advertise to its peers) more than one route to a given destination, as long as each such route has its own label. Use of these multiple routes reduce effects of congestion in the network and hence reduce the jitter.

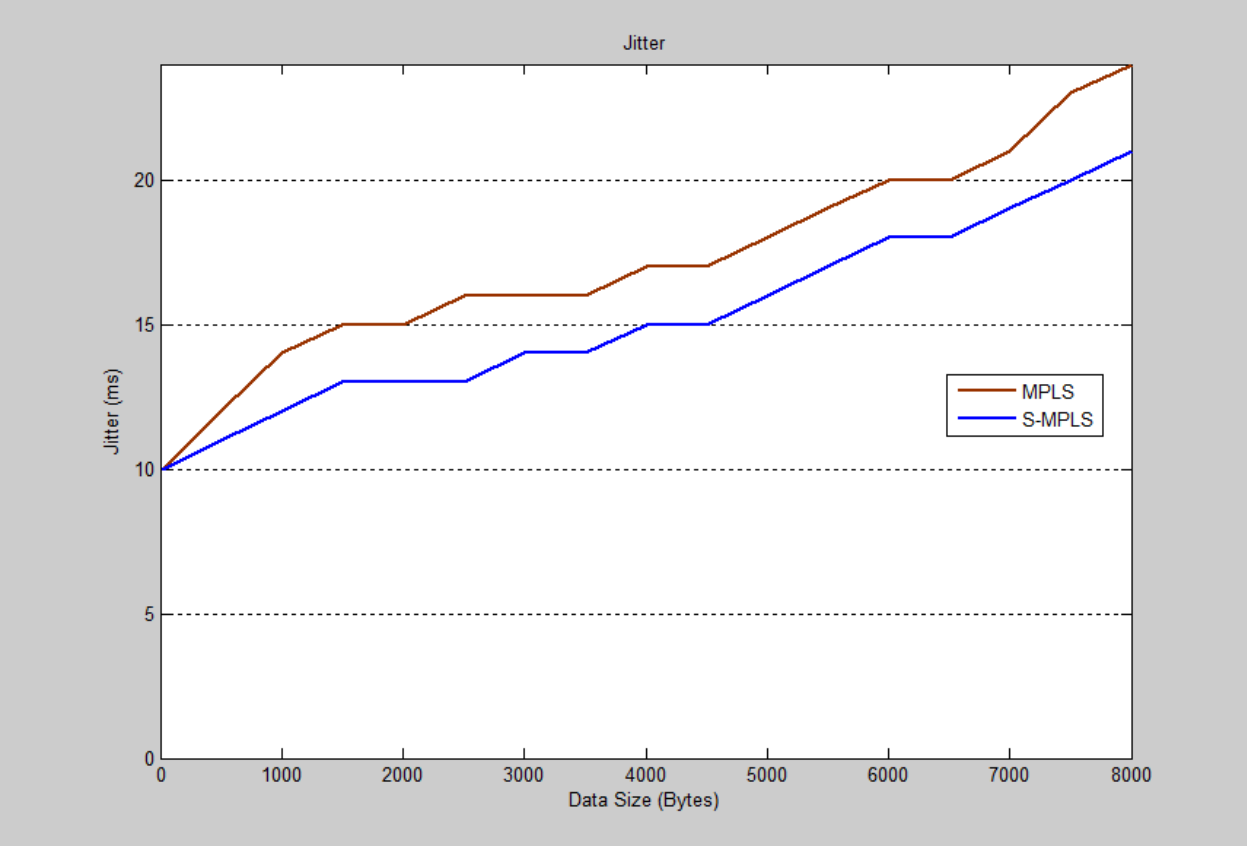

Figure 5.11: Graph of jitter for scenarios 1 & 2

6. Conclusion and Future Work

6.1. Conclusion

In this thesis impact analysis of an end-to-end MPLS architecture is done in comparison with classical MPLS using QoS parameters. This work has investigated the limitations in network architecture with multiple MPLS domains at various layers and explores the possibility of extending MPLS end-to-end by integrating access, aggregation and core network layers into single domain through the implementation of Seamless MPLS architecture.

To do the analysis and comparison of MPLS and Seamless MPLS, four QoS performance metrics such as throughput, latency, packet loss and jitter are used. First, two network scenarios are setup using eNSP with required configuration files. Next network traffic is generated using Ostinato and simulation data are collected using NQA and finally the results are presented using MATLAB.

From the study and simulation results the following conclusions can be drawn:

- Seamless MPLS can improve network performance using enabler technologies such as BGP-LU, LDP DoD, RR, etc.
- Implementation of QoS in intra-domain network alone does not guarantee end-to-end QoS.
- Compared to MPLS, Seamless MPLS improves the throughput of transferring file from one end of a network to another end of a network by up to 36.87% in the range of file size used in simulation.

- Seamless MPLS improves end-to-end packet delay up to 15.98% compared to MPLS.
- At the same congestion levels Seamless MPLS reduces packet loss by 20% and jitter by 12.5%.
- Any service provider, including Ethio telecom, can implement Seamless MPLS to integrate the separated network domains such as mobile backhaul and IP core networks into single MPLS domain with minimum cost and simplified management to enhance QoS requirement and hence improves customer satisfaction.

6.2. Future Work

Although the thesis has achieved all the objectives set in Chapter one, there are some issues to be addressed in the future. These issues are:

- Study and apply Seamless MPLS solution to Ethio telecom networks. First it is better to identify types of router (or devices) used in end-to-end IP network. The access devices must be tested if they can support MPLS and Seamless MPLS. Then the numbers of separate MPLS domains used in the company will be identified. By using test bed specific hardware, software, network configurations, etc. can be tested to verify the possibility of extending MPLS end-to-end.
- Implement and analyze traffic engineering on Seamless MPLS networks to further enhance QoS.
- Analyze the service per customer VPN and traffic classification for independent network traffic treatment

References

[1]. N. Kumar and G. Saraph, ”End-to-End QoS in Inter-domain Routing,” in *ICNS '06: Proceedings of the International conference on Networking and Services, IEEE Computer Society,* 2006, pp. 82.

[2]. Dr. A. S. Ahmad, Dr. T. Alatky, and M. Jafar, “Performance Analysis DiffServ based Quality of Service in MPLS Networks,” *International Journal of Scientific & Engineering Research,* vol. 6, no. 9, Sep. 2015.

[3]. B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure, “Inter-domain traffic engineering with BGP,” *IEEE Communications Magazine Internet Technology Series,* vol. 41, no. 5, May 2003.

[4]. B. Davie and Y. Rekhter, *MPLS Technology and Applications.* Morgan Kaufmann Publishers, 2000.

[5]. C. Labovitz, A. Ahuja, R. Wattenhfore, and S. Venkatachar, “The impact of internet policy and topology on delayed routing convergence,” in *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Proc. IEEE,* vol. 1, 2001, pp. 537-546.

[6]. T. G. Griffin, F. B. Shepherd, and G. Wilfong, “The stable path problem and inter-domain routing,” *IEEE/ACM Trans. Networking.,* vol. 10, pp. 232-243, Apr. 2002.

[7]. S. H. Zwayen and M. B. Ibrahim, "Evaluating the Performance of MPLS and Frame - Relay using OPNET Modeler," *International Journal of Computer Applications, ISSN: 0975–8887*, vol. 108, no. 12, Dec. 2014.

[8]. E. Rosen, "Using BGP to Bind MPLS Labels to Address Prefixes," RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[9]. N. Leymann, “Seamless MPLS Architecture,” draft-leymann-mpls-seamless-mpls-07(work in progress), June 2014.

[10]. Nokia, “Evolving to end-to-end MPLS architectures,” Finland, White paper, SR1610000967EN, Oct. 2016.

[11]. Huawei Technologies Co., Ltd., “Seamless MPLS Networking,” Shenzhen 518219, China, White paper, 2010.

[12]. Juniper networks, Inc., “Building Multi-Generation Scalable Networks with End-to-End MPLS,” CA 94089 USA, White paper, 2000452-001-EN, Jan. 2012.

[13]. D. Griffin, J. Grien, J. Spencer, P. Georgatsos, and P. Morand, “Interdomain routing through QoS-class planes,”in *Communication Magazine*, vol. 45, no. 2, pp. 88-95, Feb.2007.

[14]. L. Xiao, J. Wang, K. Lui, and K. Nahrstedt, “Advertising inter-domain QoS routing information,” *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1949 – 1964, Dec. 2004.

[15]. M. Boucadair and P. Morand, “A solution for providing inter-as mpls-based qos tunnels,” Draft-boucadair-pce-interas-01.txt (work in progress), October 2005.

[16]. R. Atkinson, E. S. Floyd, and Internet Architecture Board, "IAB Concerns and Recommendations Regarding Internet Research and Evolution," RFC 3869, DOI 10.17487/RFC3869, August 2004, <https://www.rfc-editor.org/info/rfc3869>.

[17]. T. Griffin and B. Presmore, "An Experimental Analysis of BGP Convergence Time," in *Proc. IEEE ICNP*, 2001.

[18]. A. Bremler-Barr, Y. Afek, and S. Schwarz, "Improved BGP Convergence via Ghost Flushing," in *Proc. IEEE INFOCOM*, 2003.

[19]. C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," in *Proc. ACM SIGCOMM*, 2000.

[20]. T. Bu, L. Gao, and D. Towsley, "On Routing Table Growth," in *Proc. IEEE Global Internet Symp.*, 2002.

[21]. T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The Stable Paths Problem and Inter-domain Routing," *IEEE/ACM Trans. Net.*, vol. 10, no. 2, pp. 232–43, Apr. 2002.

[22]. L. Xiao and K. Nahrstedt, "Reliability Models and Evaluation of Internal BGP Networks," in *Proc. IEEE INFOCOM 2004*, Hong Kong, China, 2004.

[23]. O. Nordstrom and C. Dovrolis, "Beware of BGP attacks," *SIGCOMM Computer Communications Review*, vol. 34, no. 2, pp. 1-8, Apr. 2004.

[24]. T. Beckhaus, B. Decraene, K. Tiruveedhula, M. Konstantynowicz, and L. Martini, "LDP Downstream-on-Demand in Seamless MPLS," RFC 7032, DOI 10.17487/RFC7032, October 2013, <https://www.rfc-editor.org/info/rfc7032>.

[25]. J. Bhalla, "Multiprotocol Label Switching," *International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)*, vol. 1, no. 4, Nov. 2015.

[26]. M. Tanvir and A. Said, "Decreasing Packet Loss for QoS Sensitive IP Traffic in DiffServ Enabled Network Using MPLS TE," in *International Symposium in Information Technology*, Kuala Lumpur, Malaysia, 2010, pp. 789-793.

[27]. B. Miller and E. Stewart, "Multi-Protocol Label Switching (MPLS) Conformance and Performance Testing," Ixia, White paper, Jan. 2004.

[28]. S. Gurung, "Implementation of MPLS VPN," BSc. thesis, Turku University of Applied Science, Southwest Finland, 2015.

[29]. H3C Technologies Co., Ltd., "MPLS Basics Introduction," [Online]. Available: http://www.h3c.com.hk/Products___Solutions/Technology/MPLS/Technology_Introduction/200702/201197_57_0.htm (Accessed on June 5, 2018)

[30]. Z. Li, K. Lu, "Inter-SDN (SDNi) in Seamless MPLS for Mobile Backhaul Network," draft-li-rtgwg-sdni-seamless-mpls-mbh-00 (work in progress), March 2017.

[31]. L. Andersson, I. Minei, and B. Thomas, "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <https://www.rfc-editor.org/info/rfc5036>.

[32]. T. Bates, E. Chen, and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)," RFC 4456, DOI 10.17487/RFC4456, April 2006, <https://www.rfc-editor.org/info/rfc4456>.

[33]. Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[34]. C. Feerick, "Quality of Service (QoS),"Juniper networks, Jan. 2015. [Online]. Available: https://www.juniper.net/documentation/en_US/learn-about/LA_QoS.pdf. [Accessed July 10, 2018].

[35]. W. Sugeng, J. Istiyantp, K. Mustofa, and A. Ashari, "The Impact of QoS Changes towards Network Performance," *International Journal of Computer Networks and Communications Security*, vol. 3, no. 2, pp. 48-53, Feb. 2015.

[36]. M. Hasib, "Analysis of Packet Loss Probing in Packet Networks,"Ph.D. dissertation, Department of Electronic Engineering, Queen Mary, University of London, 2006.

[37]. H. J. Lee, M. S. Kim, J. W. Hong, and, G. H. Lee, "QoS parameters to network performance metrics mapping for SLA monitoring," *KNOM Review,* vol. 5, no. 2, pp. 42-53, 2002.

[38]. V. H. Shukla, S. B. Deshmukh, "Implementing QOS Policy in MPLS Network," *International Conference on Computer Technology (ICCT)*, Vile Parle (west), Mumbai-56, 2015.

[39]. J. Evans and C. Filsfils, *Deploying IP and MPLS QOS for Multiservice Networks: Theory and practice*. USA: Morgan Kaufmann Publishers is an imprint of Elsevier, 2007.

[40]. Huawei Technologies Co., Ltd., "eNSP Global Shock Release," Dec. 7, 2012. [Online]. Available: http://support.huawei.com/enterprise/en/bulletins-service/ENEWS1000001140 (Accessed July 11, 2018).

[41]. Huawei Technologies Co., Ltd., "NQA Technology," Shenzhen 518129, China, White paper, July 1, 2014.

[42]. S. Mishra, S. Sonavane, and A. Gupta, "Study of traffic generation tools," *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE),* vol. 4, no. 6, pp. 159–162, Jun. 2015.

Appendix

Appendix A1

Scripts for MPLS Configuration

AN1

```
 sysname AN1
#
 clock timezone Nairobi add 03:00:00
#
mpls lsr-id 1.1.1.1
mpls
#
mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.2.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet0/0/0
 ip address 10.10.1.1 255.255.255.0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  import-route ospf 1
  peer 2.2.2.2 enable
  peer 2.2.2.2 route-policy HK_POLICY export
  peer 2.2.2.2 next-hop-local
  peer 2.2.2.2 label-route-capability
#
ospf 1
```

```
 area 0.0.0.1
  network 1.1.1.1 0.0.0.0
  network 10.10.1.0 0.0.0.255
  network 10.10.2.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
nqa test-instance ADMINISTRATOR ftp
 test-type ftp
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 source-port 1500
 records result 10
 description FTP_TEST
 ftp-username tne1
 ftp-password cipher %$%$PH8G%s^.N)$*Cu-Vzj=HsfDA%$%$
 ftp-filename tne.txt
nqa test-instance ADMINISTRATOR tcp
 test-type tcp
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 destination-port 1700
 records result 10
 description TCP_TEST
 probe-count 5
nqa test-instance ADMINISTRATOR udp
 test-type udp
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 destination-port 1800
 records result 10
 description UDP_TEST
nqa test-instance ADMINISTRATOR icmp
 test-type icmp
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 records result 10
 description ICMP_TEST
 probe-count 5
 source-interface LoopBack0
nqa test-instance ADMINISTRATOR lspping
 test-type lspping
 destination-address ipv4 9.9.9.9 lsp-masklen 32
 records result 10
 description LSP_TEST
nqa test-instance ADMINISTRATOR ICMPJITTER
```

```
 test-type jitter
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 destination-port 1600
 description JITTER_TEST
#
user-interface con 0
 authentication-mode password
#
return
```

ABR1

```
 sysname ABR1
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 2.2.2.2
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.2.2 255.255.255.0
 mpls
 mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.3.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet0/0/0
 ip address 10.10.12.1 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 100
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack0
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack0
 #
```

```
 ipv4-family unicast
  undo synchronization
  import-route ospf 1
  peer 1.1.1.1 enable
  peer 1.1.1.1 route-policy HK_POLICY export
  peer 1.1.1.1 reflect-client
  peer 1.1.1.1 next-hop-local
  peer 1.1.1.1 label-route-capability
  peer 3.3.3.3 enable
  peer 3.3.3.3 route-policy HK_POLICY export
  peer 3.3.3.3 reflect-client
  peer 3.3.3.3 next-hop-local
  peer 3.3.3.3 label-route-capability
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 10.10.3.0 0.0.0.255
  network 10.10.12.0 0.0.0.255
 area 0.0.0.1
  network 10.10.2.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
user-interface con 0
 authentication-mode password
#
Return
```

ASBR1

```
 sysname ASBR1
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 3.3.3.3
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.3.2 255.255.255.0
 mpls
 mpls ldp
#
```

```
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.4.1 255.255.255.0
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
bgp 100
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack0
 peer 10.10.4.2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
  import-route ospf 1
  peer 2.2.2.2 enable
  peer 2.2.2.2 route-policy HK_POLICY export
  peer 2.2.2.2 next-hop-local
  peer 2.2.2.2 label-route-capability
  peer 10.10.4.2 enable
  peer 10.10.4.2 route-policy HK_POLICY export
  peer 10.10.4.2 next-hop-local
  peer 10.10.4.2 label-route-capability check-tunnel-reachable
#
ospf 1
 area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 10.10.3.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
user-interface con 0
 authentication-mode password
#
return
```

ASBR2

```
 sysname ASBR2
#
 clock timezone Nairobi add 03:00:00
#
mpls lsr-id 4.4.4.4
mpls
#
mpls ldp
```

```
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.4.2 255.255.255.0
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.5.1 255.255.255.0
 mpls
 mpls ldp
#
interface Serial2/0/0
 link-protocol ppp
 ip address 10.10.13.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
bgp 200
 peer 5.5.5.5 as-number 200
 peer 5.5.5.5 connect-interface LoopBack0
 peer 10.10.4.1 as-number 100
 #
 ipv4-family unicast
  undo synchronization
  import-route ospf 2
  peer 5.5.5.5 enable
  peer 5.5.5.5 route-policy HK_POLICY export
  peer 5.5.5.5 next-hop-local
  peer 5.5.5.5 label-route-capability
  peer 10.10.4.1 enable
  peer 10.10.4.1 route-policy HK_POLICY export
  peer 10.10.4.1 label-route-capability check-tunnel-reachable
#
ospf 2
 area 0.0.0.0
  network 4.4.4.4 0.0.0.0
  network 10.10.5.0 0.0.0.255
  network 10.10.13.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
user-interface con 0
 authentication-mode password
```

```
#
return
```

CR1

```
 sysname CR1
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 5.5.5.5
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.5.2 255.255.255.0
 mpls
 mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.6.1 255.255.255.0
 mpls
 mpls ldp
#
interface Serial2/0/0
 link-protocol ppp
 ip address 10.10.14.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 5.5.5.5 255.255.255.255
#
bgp 200
 peer 4.4.4.4 as-number 200
 peer 4.4.4.4 connect-interface LoopBack0
 peer 6.6.6.6 as-number 200
 peer 6.6.6.6 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  import-route ospf 2
  peer 4.4.4.4 enable
  peer 4.4.4.4 route-policy HK_POLICY export
```

```
  peer 4.4.4.4 reflect-client
  peer 4.4.4.4 next-hop-local
  peer 4.4.4.4 label-route-capability
  peer 6.6.6.6 enable
  peer 6.6.6.6 route-policy HK_POLICY export
  peer 6.6.6.6 reflect-client
  peer 6.6.6.6 next-hop-local
  peer 6.6.6.6 label-route-capability
#
ospf 2
 area 0.0.0.0
  network 5.5.5.5 0.0.0.0
  network 10.10.5.0 0.0.0.255
  network 10.10.6.0 0.0.0.255
  network 10.10.14.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
user-interface con 0
 authentication-mode password
#
Return
```

CR2

```
 sysname CR2
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 6.6.6.6
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.6.2 255.255.255.0
 mpls
 mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.7.1 255.255.255.0
 mpls
 mpls ldp
#
```

```
interface Serial2/0/0
 link-protocol ppp
 ip address 10.10.13.2 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 6.6.6.6 255.255.255.255
#
bgp 200
 peer 5.5.5.5 as-number 200
 peer 5.5.5.5 connect-interface LoopBack0
 peer 7.7.7.7 as-number 200
 peer 7.7.7.7 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  import-route ospf 2
  peer 5.5.5.5 enable
  peer 5.5.5.5 route-policy HK_POLICY export
  peer 5.5.5.5 reflect-client
  peer 5.5.5.5 next-hop-local
  peer 5.5.5.5 label-route-capability
  peer 7.7.7.7 enable
  peer 7.7.7.7 route-policy HK_POLICY export
  peer 7.7.7.7 reflect-client
  peer 7.7.7.7 next-hop-local
  peer 7.7.7.7 label-route-capability
#
ospf 2
 area 0.0.0.0
  network 6.6.6.6 0.0.0.0
  network 10.10.6.0 0.0.0.255
  network 10.10.7.0 0.0.0.255
  network 10.10.13.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
user-interface con 0
 authentication-mode password
#
Return
```

ASBR3

```
 sysname ASBR3
```

```
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 7.7.7.7
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.7.2 255.255.255.0
 mpls
 mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.8.1 255.255.255.0
#
interface Serial2/0/0
 link-protocol ppp
 ip address 10.10.14.2 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 7.7.7.7 255.255.255.255
#
bgp 200
 peer 6.6.6.6 as-number 200
 peer 6.6.6.6 connect-interface LoopBack0
 peer 10.10.8.2 as-number 300
 #
 ipv4-family unicast
  undo synchronization
  import-route ospf 2
  peer 6.6.6.6 enable
  peer 6.6.6.6 route-policy HK_POLICY export
  peer 6.6.6.6 next-hop-local
  peer 6.6.6.6 label-route-capability
  peer 10.10.8.2 enable
  peer 10.10.8.2 route-policy HK_POLICY export
#
ospf 3
 area 0.0.0.0
  network 7.7.7.7 0.0.0.0
  network 10.10.7.0 0.0.0.255
```

```
  network 10.10.14.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
user-interface con 0
 authentication-mode password
#
Return
```

ASBR4

```
sysname ASBR4
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 8.8.8.8
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.8.2 255.255.255.0
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.9.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 8.8.8.8 255.255.255.255
#
bgp 300
 peer 9.9.9.9 as-number 300
 peer 9.9.9.9 connect-interface LoopBack0
 peer 10.10.8.1 as-number 200
 #
 ipv4-family unicast
  undo synchronization
  import-route ospf 3
  peer 9.9.9.9 enable
  peer 9.9.9.9 route-policy HK_POLICY export
  peer 9.9.9.9 next-hop-local
  peer 10.10.8.1 enable
  peer 10.10.8.1 route-policy HK_POLICY export
```

```
#
ospf 3
 area 0.0.0.0
  network 8.8.8.8 0.0.0.0
  network 10.10.9.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
user-interface con 0
 authentication-mode password
#
Return
```

ABR2

```
 sysname ABR2
 ftp server enable
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 9.9.9.9
mpls
#
mpls ldp
#
aaa
 authentication-scheme default
 authorization-scheme default
 accounting-scheme default
 domain default
 domain default_admin
 local-user tne1 password cipher $%$,3Z5X'IfZ9^{7rOeW+Q(s`E\%$%$
 local-user tne1 privilege level 15
 local-user tne1 ftp-directory flash:/
 local-user tne1 service-type ftp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.9.2 255.255.255.0
 mpls
 mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.10.1 255.255.255.0
 mpls
```

```
 mpls ldp
#
interface GigabitEthernet0/0/0
 ip address 10.10.15.1 255.255.255.0
#
interface LoopBack0
 ip address 9.9.9.9 255.255.255.255
#
bgp 300
 peer 8.8.8.8 as-number 300
 peer 8.8.8.8 connect-interface LoopBack0
 peer 10.10.10.10 as-number 300
 peer 10.10.10.10 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  import-route direct
  import-route ospf 3
  peer 8.8.8.8 enable
  peer 8.8.8.8 route-policy HK_POLICY export
  peer 8.8.8.8 reflect-client
  peer 8.8.8.8 next-hop-local
  peer 10.10.10.10 enable
  peer 10.10.10.10 route-policy HK_POLICY export
  peer 10.10.10.10 reflect-client
  peer 10.10.10.10 next-hop-local
#
ospf 3
 area 0.0.0.0
  network 9.9.9.9 0.0.0.0
  network 10.10.9.0 0.0.0.255
  network 10.10.15.0 0.0.0.255
 area 1.1.1.1
  network 10.10.10.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
nqa-server tcpconnect 9.9.9.9 1700
nqa-server udpecho 9.9.9.9 1600
nqa-server udpecho 9.9.9.9 1800
#
user-interface con 0
 authentication-mode password
#
Return
```

AN2

```
sysname AN2
#
mpls lsr-id 10.10.10.10
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.10.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet0/0/0
 ip address 10.10.11.1 255.255.255.0
#
interface LoopBack0
 ip address 10.10.10.10 255.255.255.255
#
bgp 300
 peer 9.9.9.9 as-number 300
 peer 9.9.9.9 connect-interface LoopBack0
#
 ipv4-family unicast
  undo synchronization
  import-route ospf 3
  import-route direct
  peer 9.9.9.9 enable
  peer 9.9.9.9 next-hop-local
  peer 9.9.9.9 route-policy HK_POLICY export
#
ospf 3
 area 0.0.0.1
 network 10.10.10.10 0.0.0.0
 network 10.10.10.0 0.0.0.255
 network 10.10.11.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
#
user-interface con 0
 authentication-mode password
#
Return
```

Appendix A2

Scripts for Seamless MPLS Configuration

AN1

```
sysname AN1
#
 clock timezone Nairobi add 03:00:00
#
mpls lsr-id 1.1.1.1
mpls
#
mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.2.1 255.255.255.0
 mpls
 mpls ldp
 mpls ldp advertisement dod
#
interface GigabitEthernet0/0/0
 ip address 10.10.1.1 255.255.255.0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  network 1.1.1.1 255.255.255.255
  import-route ospf 1
  peer 2.2.2.2 enable
  peer 2.2.2.2 route-policy HK_POLICY export
  peer 2.2.2.2 next-hop-local
  peer 2.2.2.2 label-route-capability
#
ospf 1
 area 0.0.0.1
  network 1.1.1.1 0.0.0.0
  network 10.10.1.0 0.0.0.255
```

```
  network 10.10.2.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
nqa test-instance ADMINISTRATOR ftp
 test-type ftp
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 source-port 1500
 records result 10
 description FTP_TEST
 ftp-username tne1
 ftp-password cipher %$%$TRC",eknrD{*AA;}kHnNyo.z%$%$
 ftp-filename tne.txt
nqa test-instance ADMINISTRATOR tcp
 test-type tcp
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 destination-port 1700
 records result 10
 description TCP_TEST
 probe-count 5
nqa test-instance ADMINISTRATOR udp
 test-type udp
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 destination-port 1800
 records result 10
 description UDP_TEST
nqa test-instance ADMINISTRATOR icmp
 test-type icmp
 destination-address ipv4 9.9.9.9
 source-address ipv4 1.1.1.1
 records result 10
 description ICMP_TEST
 probe-count 5
 source-interface LoopBack0
nqa test-instance ADMINISTRATOR lspping
 test-type lspping
 destination-address ipv4 9.9.9.9 lsp-masklen 32
 records result 10
 description LSP_TEST
nqa test-instance ADMINISTRATOR ICMPJITTER
 test-type jitter
 destination-address ipv4 9.9.9.9
```

```
 source-address ipv4 1.1.1.1
 destination-port 1600
 description JITTER_TEST
#
user-interface con 0
 authentication-mode password
#
Return
```

ABR1

```
sysname ABR1
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 2.2.2.2
mpls
#
mpls ldp
#
mpls ldp remote-peer abr2
 remote-ip 9.9.9.9
#
mpls ldp remote-peer an1
 remote-ip auto-dod-request
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.2.2 255.255.255.0
 mpls
 mpls ldp
 mpls ldp advertisement dod
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.3.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet0/0/0
 ip address 10.10.12.1 255.255.255.0
#
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
bgp 100
```

```
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack0
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  peer 1.1.1.1 enable
  peer 1.1.1.1 route-policy HK_POLICY export
  peer 1.1.1.1 reflect-client
  peer 1.1.1.1 next-hop-local
  peer 1.1.1.1 label-route-capability
  peer 3.3.3.3 enable
  peer 3.3.3.3 route-policy HK_POLICY export
  peer 3.3.3.3 reflect-client
  peer 3.3.3.3 next-hop-local
  peer 3.3.3.3 label-route-capability
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 10.10.3.0 0.0.0.255
  network 10.10.12.0 0.0.0.255
 area 0.0.0.1
  network 10.10.2.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
user-interface con 0
 authentication-mode password
#
Return
```

ASBR1

```
 sysname ASBR1
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 3.3.3.3
mpls
#
mpls ldp
#
interface Serial1/0/0
```

```
 link-protocol ppp
 ip address 10.10.3.2 255.255.255.0
 mpls
 mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.4.1 255.255.255.0
 mpls
#
interface LoopBack0
 ip address 3.3.3.3 255.255.255.255
#
bgp 100
 peer 2.2.2.2 as-number 100
 peer 2.2.2.2 connect-interface LoopBack0
 peer 10.10.4.2 as-number 200
 #
 ipv4-family unicast
  undo synchronization
  peer 2.2.2.2 enable
  peer 2.2.2.2 route-policy HK_POLICY export
  peer 2.2.2.2 next-hop-local
  peer 2.2.2.2 label-route-capability
  peer 10.10.4.2 enable
  peer 10.10.4.2 route-policy HK_POLICY export
  peer 10.10.4.2 next-hop-local
  peer 10.10.4.2 label-route-capability check-tunnel-reachable
#
ospf 1
 area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 10.10.3.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
user-interface con 0
 authentication-mode password
#
Return
```

ASBR2

```
 sysname ASBR2
#
```

```
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 4.4.4.4
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.4.2 255.255.255.0
 mpls
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.5.1 255.255.255.0
 mpls
 mpls ldp
#
interface Serial2/0/0
 link-protocol ppp
 ip address 10.10.13.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
bgp 200
 peer 5.5.5.5 as-number 200
 peer 5.5.5.5 connect-interface LoopBack0
 peer 10.10.4.1 as-number 100
 #
 ipv4-family unicast
  undo synchronization
  peer 5.5.5.5 enable
  peer 5.5.5.5 route-policy HK_POLICY export
  peer 5.5.5.5 next-hop-local
  peer 5.5.5.5 label-route-capability
  peer 10.10.4.1 enable
  peer 10.10.4.1 route-policy HK_POLICY export
  peer 10.10.4.1 label-route-capability check-tunnel-reachable
#
ospf 2
 area 0.0.0.0
  network 4.4.4.4 0.0.0.0
  network 10.10.5.0 0.0.0.255
```

```
  network 10.10.13.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
user-interface con 0
 authentication-mode password
#
Return
```

CR1

```
 sysname CR1
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 5.5.5.5
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.5.2 255.255.255.0
 mpls
 mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.6.1 255.255.255.0
 mpls
 mpls ldp
#
interface Serial2/0/0
 link-protocol ppp
 ip address 10.10.14.1 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 5.5.5.5 255.255.255.255
#
bgp 200
 peer 4.4.4.4 as-number 200
 peer 4.4.4.4 connect-interface LoopBack0
 peer 6.6.6.6 as-number 200
```

```
 peer 6.6.6.6 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  peer 4.4.4.4 enable
  peer 4.4.4.4 route-policy HK_POLICY export
  peer 4.4.4.4 reflect-client
  peer 4.4.4.4 next-hop-local
  peer 4.4.4.4 label-route-capability
  peer 6.6.6.6 enable
  peer 6.6.6.6 route-policy HK_POLICY export
  peer 6.6.6.6 reflect-client
  peer 6.6.6.6 next-hop-local
  peer 6.6.6.6 label-route-capability
#
ospf 2
 area 0.0.0.0
  network 5.5.5.5 0.0.0.0
  network 10.10.5.0 0.0.0.255
  network 10.10.6.0 0.0.0.255
  network 10.10.14.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
user-interface con 0
 authentication-mode password
#
Return
```

CR2

```
 sysname CR2
#
 clock timezone Nairobi add 03:00:00
#
mpls lsr-id 6.6.6.6
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.6.2 255.255.255.0
 mpls
 mpls ldp
```

```
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.7.1 255.255.255.0
 mpls
 mpls ldp
#
interface Serial2/0/0
 link-protocol ppp
 ip address 10.10.13.2 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 6.6.6.6 255.255.255.255
#
bgp 200
 peer 5.5.5.5 as-number 200
 peer 5.5.5.5 connect-interface LoopBack0
 peer 7.7.7.7 as-number 200
 peer 7.7.7.7 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  peer 5.5.5.5 enable
  peer 5.5.5.5 route-policy HK_POLICY export
  peer 5.5.5.5 reflect-client
  peer 5.5.5.5 next-hop-local
  peer 5.5.5.5 label-route-capability
  peer 7.7.7.7 enable
  peer 7.7.7.7 route-policy HK_POLICY export
  peer 7.7.7.7 reflect-client
  peer 7.7.7.7 next-hop-local
  peer 7.7.7.7 label-route-capability
#
ospf 2
 area 0.0.0.0
  network 6.6.6.6 0.0.0.0
  network 10.10.6.0 0.0.0.255
  network 10.10.7.0 0.0.0.255
  network 10.10.13.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
user-interface con 0
```

```
 authentication-mode password
#
Return
```

ASBR3

```
 sysname ASBR3
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 7.7.7.7
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.7.2 255.255.255.0
 mpls
 mpls ldp
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.8.1 255.255.255.0
 mpls
#
interface Serial2/0/0
 link-protocol ppp
 ip address 10.10.14.2 255.255.255.0
 mpls
 mpls ldp
#
interface LoopBack0
 ip address 7.7.7.7 255.255.255.255
#
bgp 200
 peer 6.6.6.6 as-number 200
 peer 6.6.6.6 connect-interface LoopBack0
 peer 10.10.8.2 as-number 300
 #
 ipv4-family unicast
  undo synchronization
  peer 6.6.6.6 enable
  peer 6.6.6.6 route-policy HK_POLICY export
  peer 6.6.6.6 next-hop-local
  peer 6.6.6.6 label-route-capability
```

```
  peer 10.10.8.2 enable
  peer 10.10.8.2 route-policy HK_POLICY export
  peer 10.10.8.2 label-route-capability check-tunnel-reachable
#
ospf 2
 area 0.0.0.0
  network 7.7.7.7 0.0.0.0
  network 10.10.7.0 0.0.0.255
  network 10.10.14.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
user-interface con 0
 authentication-mode password
#
Return
```

ASBR4

```
 sysname ASBR4
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 8.8.8.8
mpls
#
mpls ldp
#
mpls ldp remote-peer abr2
 remote-ip auto-dod-request
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.8.2 255.255.255.0
 mpls
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.9.1 255.255.255.0
 mpls
 mpls ldp
 mpls ldp advertisement dod
#
interface LoopBack0
 ip address 8.8.8.8 255.255.255.255
```

```
#
bgp 300
 peer 9.9.9.9 as-number 300
 peer 9.9.9.9 connect-interface LoopBack0
 peer 10.10.8.1 as-number 200
 #
 ipv4-family unicast
  undo synchronization
  import-route ospf 3
  peer 9.9.9.9 enable
  peer 9.9.9.9 route-policy HK_POLICY export
  peer 9.9.9.9 next-hop-local
  peer 9.9.9.9 label-route-capability
  peer 10.10.8.1 enable
  peer 10.10.8.1 route-policy HK_POLICY export
  peer 10.10.8.1 label-route-capability check-tunnel-reachable
#
ospf 3
 area 0.0.0.0
  network 8.8.8.8 0.0.0.0
  network 10.10.9.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
user-interface con 0
 authentication-mode password
#
Return
```

ABR2

```
 sysname ABR2
 ftp server enable
#
clock timezone Nairobi add 03:00:00
#
mpls lsr-id 9.9.9.9
mpls
#
mpls ldp
#
mpls ldp remote-peer an1
 remote-ip 1.1.1.1
#
aaa
```

```
 authentication-scheme default
 authorization-scheme default
 accounting-scheme default
 domain default
 domain default_admin
 local-user tne1 password cipher %$%$N:)ZSilL_3)zH2261(tPyqS=%$%$
 local-user tne1 privilege level 15
 local-user tne1 ftp-directory flash:/
 local-user tne1 service-type ftp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.9.2 255.255.255.0
 mpls
 mpls ldp
 mpls ldp advertisement dod
#
interface Serial1/0/1
 link-protocol ppp
 ip address 10.10.10.1 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet0/0/0
 ip address 10.10.15.1 255.255.255.0
#
interface LoopBack0
 ip address 9.9.9.9 255.255.255.255
#
bgp 300
 peer 8.8.8.8 as-number 300
 peer 8.8.8.8 connect-interface LoopBack0
 peer 10.10.10.10 as-number 300
 peer 10.10.10.10 connect-interface LoopBack0
 #
 ipv4-family unicast
  undo synchronization
  network 9.9.9.9 255.255.255.255
  import-route ospf 3
  peer 8.8.8.8 enable
  peer 8.8.8.8 route-policy HK_POLICY export
  peer 8.8.8.8 reflect-client
  peer 8.8.8.8 next-hop-local
  peer 8.8.8.8 label-route-capability
  peer 10.10.10.10 enable
  peer 10.10.10.10 route-policy HK_POLICY export
```

```
  peer 10.10.10.10 reflect-client
  peer 10.10.10.10 next-hop-local
  peer 10.10.10.10 label-route-capability
#
ospf 3
 area 0.0.0.0
  network 9.9.9.9 0.0.0.0
  network 10.10.9.0 0.0.0.255
  network 10.10.15.0 0.0.0.255
 area 0.0.0.1
  network 10.10.10.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
nqa-server tcpconnect 9.9.9.9 1700
nqa-server udpecho 9.9.9.9 1600
nqa-server udpecho 9.9.9.9 1800
#
user-interface con 0
 authentication-mode password
```

AN2

```
sysname AN2
#
mpls lsr-id 10.10.10.10
mpls
#
mpls ldp
#
interface Serial1/0/0
 link-protocol ppp
 ip address 10.10.10.2 255.255.255.0
 mpls
 mpls ldp
#
interface GigabitEthernet0/0/0
 ip address 10.10.11.1 255.255.255.0
#
interface LoopBack0
 ip address 10.10.10.10 255.255.255.255
#
bgp 300
 peer 9.9.9.9 as-number 300
 peer 9.9.9.9 connect-interface LoopBack0
```

```
#
 ipv4-family unicast
  undo synchronization
  import-route ospf 3
  import-route direct
  peer 9.9.9.9 enable
  peer 9.9.9.9 next-hop-local
  peer 9.9.9.9 route-policy HK_POLICY export
#
ospf 3
 area 0.0.0.1
 network 10.10.10.10 0.0.0.0
 network 10.10.10.0 0.0.0.255
#
route-policy HK_POLICY permit node 1
 apply mpls-label
#
user-interface con 0
 authentication-mode password
#
Return
```