Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Telecommunication Engineering Graduate Program

SDN Coupled Seamless MPLS Traffic Engineering Performance Analysis

By:

Medhin Worku

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, 2021

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Telecom Network Engineering Graduate Program

SDN Coupled Seamless MPLS Traffic Engineering Performance Analysis

By:

Medhin Worku

Approval by Board of Examiners

_____	_____	_____/_____/_____
School Dean	Signature	Date
Dr. Yalemzewd Negash	_____	_____/_____/_____
Advisor	Signature	Date
Dr. Beneyam Berehanu	_____	_____/_____/_____
Examiner	Signature	Date
Dr. Mesfin Kifle	_____	_____/_____/_____
Examiner	Signature	Date

November, 2021

Addis Ababa, Ethiopia

Declaration

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

Name: Medhin Worku Signature: _____

Place: Addis Ababa Date of Submission: _____

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

Advisor's Name: Dr. Yalemzewd Negash Signature_____

Acknowledgement

First and foremost, I am extremely grateful to God for his generosity and guidance in all my life. Secondly, I would like to express my sincere gratitude to my advisor Dr. Yalemzewd Negash for his insightful comments and suggestions, continuous follow-up, guidance, and patience during the course of this thesis. His observation, unreserved advice and support were very useful and constructive. I also would like to offer my special thanks to my family for their continuous support. Lastly, I would like to express my appreciation to AAiT - Ethio Telecom collaboration for their commitment and sponsorship of this postgraduate program.

Abstract

Internet Service Providers (ISPs), are requiring better performing MPLS networks. Seamless MPLS (S-MPLS) provides an end-to-end MPLS networks by integrating access, aggregation and core networks into a single MPLS domain, improving the performance of MPLS networks. The ISP network performance can also be further enhanced by implementing Traffic Engineering (TE). However, in S-MPLS Traffic Engineering (S-MPLS-TE), some of the tunnels fail to find a path and re-route traffic in the network when the best path goes down or gets congested. Software Defined Networking's (SDN's) Path Computation (PCE) feature can provide an approach to overcome these shortcomings of S-MPLS-TE. In this thesis, the performance of S-MPLS-TE coupled with SDN is analyzed and compared with S-MPLS-TE. In order to carry out the research, literatures relevant to S-MPLS-TE and SDN are reviewed. Based on the literatures reviewed, a simulation network, for S-MPLS-TE and SDN coupled S-MPLS-TE simulation, is setup using GNS-3, OpenDaylight controller, Ostinato, Postman and PRTG. The topology of the simulation network setup is based on Ethio telecom's access, aggregation and core networks. For the setup, ISIS, MPLS, BGP-LU, BGP-LS, PCEP, NETCONF and RESTCONF are utilized as networking protocols. In the simulation, latency and packet loss of both S-MPLS-TE and SDN coupled S-MPLS-TE simulations are measured, analyzed and compared for 250Kbps and 350Kbps TCP traffic scenarios. By coupling SDN with S-MPLS-TE, latency is enhanced by 12.26% in 250Kbps traffic scenario and by 16.07% in 350Kbps traffic scenario. Regarding packet loss, a 13.79% improvement in 250Kbps traffic scenario and a 22.667% improvement in 350Kbps traffic scenario are observed.

Keywords—MPLS, Seamless MPLS, Software Defined Networking, Path Computation Element, Simulation, Performance, Analysis

Contents

Acknowledgement iii
Abstract iv
Contents v
List of Acronyms vii
List of Figures x
List of Tables xi
1. Introduction 1
1.1. Background 1
1.2. Problem Statement 2
1.3. Objective 2
1.3.1. General Objective 2
1.3.2. Specific Objectives 3
1.4. Methodology 3
1.4.1. Literature Review 3
1.4.2. S-MPLS-TE and SDN Coupled S-MPLS-TE Simulation 3
1.5. Literature Review 4
1.6. Scope of the Research 6
1.7. Limitation of the Research 6
1.8. Thesis Layout 7
2. Seamless MPLS 8
2.1. Introduction to MPLS 8
2.1.1. MPLS Header 8
2.1.2. Forward Equivalence Class (FEC) 9
2.1.3. Label Switching Router (LSR) 10
2.1.4. Label Switched Path (LSP) 11
2.1.5. Label Distribution 11
2.1.6. MPLS Operation 12
2.1.7. MPLS Traffic Engineering (MPLS-TE) 13
2.2. MPLS Router Architecture 14
2.2.1. Control Plane 15
2.2.2. Data Plane 15
2.3. Seamless MPLS (S-MPLS) 16
2.3.1. Seamless MPLS Architectures 17

2.3.1.1. Labeled BGP Core and Aggregation 18
2.3.1.2. Labeled BGP Core, Access and Aggregation 20
2.3.1.3. Labeled BGP Core and Aggregation with Redistribution into Access Network IGP 21
2.3.2. Seamless MPLS Attributes 22
2.4. Inter-AS MPLS Traffic Engineering 23
2.4.1. Setting Up Inter-AS MPLS-TE Tunnels 23
2.4.2. Inter-AS MPLS-TE Limitations 24
3. Software Defined Networking (SDN) 25
3.1. SDN Architecture 26
3.1.1. Infrastructure Layer 26
3.1.2. Control Layer 27
3.1.3. Application Layer 29
3.2. Southbound Interface (SBI) Protocols 30
3.2.1. BGP-Link State (BGP-LS) 30
3.2.2. Path Computation Element Protocol (PCEP) 31
3.2.3. NETCONF 32
3.3. Northbound Interface (NBI) Protocol 32
3.3.1. RESTCONF 33
3.4. SDN Controller 33
3.4.1. OpenDaylight (ODL) Controller 34
3.4.1.1. The Southbound Interface and Protocol Plugins 38
3.5. Path Computation Element (PCE) 38
3.5.1. PCE Architecture 40
4. Simulation Network Setup 42
4.1. S-MPLS-TE Simulation 44
4.2. SDN Coupled S-MPLS-TE Simulation 49
5. Simulation Result and Analysis 58
5.1. Latency Analysis 61
5.2. Packet Loss Analysis 63
6. Conclusion 66
7. Future Works 67
References 68
Appendix 71

List of Acronyms

2G	Second Generation
3G	Third Generation
4G	Fourth Generation
AAA	Authentication, Authorization and Accounting
ABR	Area Border Router
AD-SAL	API-Driven SAL
AN	Access Network
API	Application Programming Interface
AS	Autonomous System
ASBR	AS Boundary Router
ASG	Aggregation Service Gateway
ATM	Asynchronous Transport Mode
BD	Bahir Dar
BGP	Boarder Gateway Protocol
BGP-LS	BGP-Link State
BGP-LU	BGP-Labeled Unicast
BL	Bole
BNSF	Base Network Service Functions
BR	Backbone Router
CAPEX	Capital Expenditures
CCP	Centralized Control Plane
CLI	Command Line Interface
CSG	Cell Service Gateway
DCP	Distributed Control Plane
eBGP	Exterior BGP
EGP	Exterior Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
ER	Edge Router
ERO	Explicit Route Object
FEC	Forwarding Equivalence Class
FRM	Forwarding Rules Manager
GE	Gigabit Ethernet
GNS3	Graphical Network Simulator-3
GUI	Graphical User Interface
HCP	Hierarchical Control Plane
HE	Head End
HTTP	Hypertext Transfer Protocol
iBGP	Interior BGP
IEEE	Institute of Electrical and Electronics Engineers

IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet Protocol
IS-IS	Intermediate System - Intermediate System
ISP	Internet Service Provider
JAR	Java ARchive
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
L2	Layer 2
L3	Layer 3
LAN	Local Area Network
LDP	Label Distribution Protocol
LER	Label Edge Router
LFIB	Label Forwarding Information Base
LIB	Label Information Base
LSP	Label Switched Path
LSPDB	LSP Database
LSR	Label Switching Router
LTE	Long Term Evolution
MAC	Media Access Control
MD-SAL	Model-Driven SAL
MPLS	Multi-Protocol Label Switching
MPLS-TE	MPLS Traffic Engineering
NAT	Network Address Translation
NB	Northbound
NBI	NB Interface
NETCONF	Network Configuration Protocol
NH	Next Hop
NHS	Next Hop-Self
NMS	Network Management System
NOS	Network Operating System
OAM	Operation, Administration and Maintenance
ODL	OpenDaylight
OF	OpenFlow
OPEX	Operational Expenditures
OSGi	Open Service Gateway Interface
OSPF	Open Shortest Path First
OVS	OpenvSwtich
OVSDB	OVS Database
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	Path Computation Element Protocol
PCErr	Path Computation Error

PCNtf	Path Computation Notification
PCRep	Path Computation Reply
PCReq	Path Computation Request
PE	Provider Edge
PNSF	Platform Network Service Functions
PRTG	Paessler Router Traffic Grapher
PW	Pseudo Wire
QoS	Quality of Service
RAM	Random Access Memory
RAN	Radio Access Network
REST	Representational State Transfer
RESTCONF	REST Configuration Protocol
RFC	Request for Comment
RIB	Routing Information Base
RIP	Routing Information Protocol
RPC	Remote Procedure Call
RR	Route Reflector
RSG	Radio Service Gateway
RSVP	Resource Reservation Protocol
SAL	Service Abstraction Layer
SB	Southbound
SBI	SB Interface
SDN	Software Defined Networking
SFC	Service Function Chaining
S-MPLS	Seamless MPLS
SNMP	Simple Network Management Protocol
SP	Service Provider
SR	Stable Release
SSH	Secure Shell
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TE	Traffic Engineering
TED	TE Database
TTL	Time to Live
TTP	Table Type Pattern
VLAN	Virtual LAN
VPLS	Virtual Private LAN Service
VPN	Virtual Private Network
VPWS	Virtual Private Wire Service
VRF	Virtual Routing and Forwarding
VTN	Virtual Tenant Network
WAN	Wide Area Network
XML	Extensible Markup Language

List of Figures

Figure 1: Position of MPLS in OSI Model 8
Figure 2: MPLS Packet Header 9
Figure 3: MPLS Network Architecture 10
Figure 4: MPLS Basic Operation 13
Figure 5: MPLS Router Conceptual Architecture 15
Figure 6: Labeled BGP Core and Aggregation 19
Figure 7: Labeled BGP Core, Access and Aggregation 20
Figure 8: Labeled BGP Core and Aggregation with Redistribution into AN IGP 22
Figure 9: SDN Architecture 26
Figure 10: An Overview of Popular SDN Control Plane Models 28
Figure 11: Components of Each Layer of SDN Architecture 30
Figure 12: The Complete View of OpenDaylight Architecture 35
Figure 13: The Evolution of SAL From API-Driven to Model-Driven 37
Figure 14: PCE Internal Architecture 41
Figure 15: Simulation Network Setup 43
Figure 16: IS-IS Neighborship in Bole Access and Aggregation Domain 45
Figure 17: MPLS LDP Neighborship in Bole Access and Aggregation Domain 45
Figure 18: Sample BGP-LU Configuration on BL-CSG 46
Figure 19: iBGP and eBGP Neighborship in Bole Access and Aggregation Domain 47
Figure 20: Sample ASBR Forced Flooding Configuration on BL-RSG-1 47
Figure 21: S-MPLS-TE Tunnel te-21 Configuration on BL-CSG 48
Figure 22: ERO of S-MPLS-TE Tunnel te-21 Configuration on BL-CSG 48
Figure 23: Status of S-MPLS-TE Tunnel te-21 on BL-CSG 48
Figure 24: ODL Controller CLI 49
Figure 25: Installed ODL Features (BGP, PCEP, NETCONF and RESTCONF) 50

Figure 26: ODL Controller Configuration Files in Karaf Directory 51
Figure 27: BGP RIB ID Configuration on example-bgp-rib module 51
Figure 28: BGP-LS Peer Configuration on example-bgp-peer module 52
Figure 29: BGP-LS Configuration on BL-CSG 52
Figure 30: BGP-LS Neighborship Between BL-CSG and ODL Controller 53
Figure 31: PCE Configuration on BL-CSG 53
Figure 32: PCE Peering Between ODL Controller and PCC on BL-CSG 53
Figure 33: Sample SSH and NETCONF configuration on BL-CSG 54
Figure 34: Mounting BL-CSG on ODL Controller Using Postman 54
Figure 35: SSH and NETCONF Sessions on BL-CSG 55
Figure 36: Postman REST POST Request 56
Figure 37: Status of ODL Controller Coupled S-MPLS-TE Tunnel te-100 on BL-CSG 57
Figure 38: SDN Coupled S-MPLS-TE Simulation Network Setup 57
Figure 39: Latency of S-MPLS-TE and ODL + S-MPLS-TE Tunnels at 250Kbps 61
Figure 40: Latency of S-MPLS-TE and ODL + S-MPLS-TE Tunnels at 350Kbps 62
Figure 41: Packet Loss of S-MPLS-TE and ODL + S-MPLS-TE Tunnels at 250Kbps 64
Figure 42: Packet Loss of S-MPLS-TE and ODL + S-MPLS-TE Tunnels at 350Kbps 64

List of Tables

Table 1: Latency of S-MPLS-TE and ODL coupled S-MPLS-TE Tunnels for 250Kbps 58
Table 2: Measured Packet Loss for S-MPLS-TE and ODL controller coupled S-MPLS-TE Tunnels at 250Kbps 59
Table 3: Latency of S-MPLS-TE and ODL coupled S-MPLS-TE Tunnels for 350Kbps 60
Table 4: Measured Packet Loss for S-MPLS-TE and ODL controller coupled S-MPLS-TE Tunnels at 350Kbps 60

1. Introduction

1.1. Background

Highly scalable, resilient, flexible and manageable network architectures are being demanded by the increased adoption of Multi-Protocol Label Switching (MPLS) with in operator networks. Seamless MPLS (S-MPLS) provides a better substitute for implementing end-to-end MPLS networks by unifying access, aggregation and core networks into a single MPLS domain. With S-MPLS architecture, the entire network uses unified IP/MPLS networking technology, with an end-to-end control plane [1].

IP/MPLS network has control and data plane. The control plane is accountable for maintaining connections and exchanging protocol information with other routers or network devices. The data plane or the forwarding plane is in charge of forwarding packets from incoming interface to outgoing interface through the router.

Software Defined Networking (SDN) is an architecture, which decouples control plane and data plane leaving only data plane on network devices. On the other hand control plane is retained on the SDN controller [2].

MPLS Traffic Engineering (MPLS-TE) faces the problem of creating backup path immediately when the best path goes down or gets congested. By using SDN, MPLS-TE performance is enhanced for the tunnel creation. It has been found that coupling SDN with MPLS offers better performance in terms of latency, response time and bandwidth utilization [3].

However, performance analysis of S-MPLS-TE coupled with SDN is left open for research. As the end-to-end control plane of access, aggregation and core networks are unified, coupling SDN to S-MPLS-TE will be feasible.

This research will analyze the performance of S-MPLS-TE coupled with SDN as compared to S-MPLS-TE. Its goals are to analyze latency and packet loss of SDN coupled S-MPLS-TE. It utilizes network emulating platform, virtual machine and network performance monitoring tool for the analysis.

1.2. Problem Statement

One of the most important application of IP/S-MPLS networks is traffic engineering, which has several advantages. In big Internet Service Provider (ISP) networks, however, traffic patterns can change frequently and these traffic variations can have number of undesirable effects on the network performance. In ISP networks, huge number of tunnels must find new substitute paths through the network, if a link that carries them fails.

With the implementation of S-MPLS-TE, each headend router, in individual Autonomous System (AS) associated with the tunnels, optimizes traffic routing individually for itself, competing for the same limited amount of bandwidth without the consideration of the other network routers' requirements. In such a condition, some of the tunnels fail to find a path and re-route traffic in the network.

To address these problems, S-MPLS-TE can be coupled with SDN. SDN controller will have complete view of network topology, can monitor link conditions and create or delete tunnels [3]. In this study, latency and packet loss of the SDN coupled S-MPLS-TE will be analyzed. For the analysis experimental network setup will be simulated and evaluated.

1.3. Objective

1.3.1. General Objective

The general objective of this study is to analyze the performance of SDN coupled S-MPLS-TE in comparison with S-MPLS-TE.

1.3.2. Specific Objectives

Specific objective of this thesis are the following:

- To investigate S-MPLS-TE capable technologies and protocols
- To investigate SDN architecture and technologies
- To setup experimental network based on Ethio telecom's access, aggregation and core network
- To simulate S-MPLS-TE
- To simulate SDN coupled S-MPLS-TE
- To analyze performance of S-MPLS-TE and SDN coupled S-MPLS-TE
- To compare analysis results

1.4. Methodology

In this section, methodology employed to conduct the study will be discussed in detail. Procedures followed to carry out the research are literature review, S-MPLS-TE simulation, SDN coupled S-MPLS-TE simulation and simulation result analysis.

1.4.1. Literature Review

State of the art literatures relevant to S-MPLS-TE and SDN are collected, compared and analyzed to identify the most relevant among them. Most literatures that are related to S-MPLS-TE and SDN will be reviewed. IEEE Explore, ResearchGate and Google books are main sources of the literatures selected. Related potential literatures are selected, examined, compared and evaluated for the validity of this study.

1.4.2. S-MPLS-TE and SDN Coupled S-MPLS-TE Simulation

S-MPLS-TE and SDN coupled S-MPLS-TE will be simulated on experimental network setup based on Ethio telecom's access, aggregation and core network. It is composed of 13 routers, 1 hub,1 SDN controller and Ostinato traffic generators. The simulation will be performed by utilizing the following tools.

- **Graphical Network Simulator-3 (GNS3):** GNS3 is an open source software that emulates routers and switches hardware to simulate complex networks. GNS3 can be used on any computer to experiment with various router configurations and build networks, all without plugging in a single physical network cable. The experimental network setup for both S-MPLS-TE and SDN coupled S-MPLS-TE will be emulated using GNS3.
- **OpenDaylight (ODL):** OpenDaylight project is an open source SDN project aimed at enhancing SDN by offering a community-led and industry-supported framework for the OpenDaylight Controller. It is both multiprotocol and modular.
- **Ostinato:** Ostinato is a packet crafter, network traffic generator and analyzer with a friendly graphical user interface. It can craft and send packets of several streams with different protocols at different rates. Ostinato will be used to generate traffic for the simulation.
- **Postman:** Postman is a collaboration platform for Application Programming Interface (API) development. Postman will be used to send REST API requests to directly configure the SDN controller.
- **Paessler Router Traffic Grapher (PRTG):** PRTG is network-monitoring software that collects statistics from designated hosts such as routers, servers, switches and other important devices or applications**.** PRTG network analyzer will collect and analyze latency and packet loss data from the simulation on GNS3.

1.5. Literature Review

IP/MPLS networks have evolved and have become critically important for ISPs. They are built around equipment such as routers, switches, and firewalls. Both the control plane and the data plane are parched into these hardware devices. The control plane determines where the traffic is sent. The data plane forwards that traffic based on how the control plane is configured. MPLS is used to perform traffic engineering in IP/MPLS networks,

providing greater reliability and more efficient usage of network resources. But, MPLS-TE suffers from the failure to create back up tunnel during congestion and link failure. When MPLS-TE is coupled with SDN, the performance of MPLS-TE has improved.

S-MPLS merges multiple MPLS domains of ISP networks into single MPLS domain. The core, aggregation and access networks of ISP are unified with a single control plane in S-MPLS. S-MPLS-TE also suffers from the incapability to generate a backup tunnel at the time of congestion and link failure. SDN separates the control plane from the data plane. It effectively decouples the hardware from the software layer. So, coupling SDN with S-MPLS-TE will be feasible. However, the performance of SDN coupled S-MPLS-TE, to the best of our knowledge, is not yet analyzed.

Several researchers have analyzed the performance of SDN coupled with IP/MPLS networks. The authors in [3] analyzed the performance of SDN when coupled with MPLS. They found that, by using some of the core features of SDN, MPLS performance is enhanced for the tunnel creation. They have also found that coupling SDN with MPLS offers better performance in terms of latency, response time and bandwidth utilization. The gap of the research is that it did not analyzed the performance of S-MPLS-TE when coupled with SDN.

On the other hand, the authors in [4] presented a class based hybrid SDN model for MPLS networks. It used the standard MPLS data-plane together with a control plane based on OpenFlow to come up with a systematic incremental deployment methodology as well as a hybrid operation model. The research presented a hybrid network model which encompasses both SDN and MPLS based functionality and experimentally evaluated it. They found that the hybrid networking model gives the proof of concept for the co-existence of MPLS and SDN based network elements in the network. Their research has proven that a centralized controller is able to provision traffic flows better as compared

to decentralized based approaches using MPLS. The hybrid model provided a better alternative for the smooth transition of a legacy network to SDN enabled network. The inadequacy of the study was it did not analyze the performance of the proposed SDN model.

Authors in [5] proposed a new traffic engineering architecture for the hybrid SDN-MPLS network. It applied OpenFlow-enabled switches over the edge of the network to improve flow-level management flexibility, while MPLS routers are considered as the core of the network to make the scheme applicable for existing MPLS networks. The proposed scheme increased the network throughput and reduced the total packet loss significantly. The limitation of this research is that it was restricted to MPLS and it did not consider S-MPLS-TE.

1.6. Scope of the Research

This research analyzes the performance of SDN coupled seamless MPLS TE. The research is carried out by simulating, measuring and analyzing performance of two TE tunnels on seamless MPLS TE and SDN coupled seamless MPLS TE network setups. The simulation is performed on network setup based on part of Ethio telecom's network with three autonomous systems. The research uses latency and packet loss as performance measurement parameters.

1.7. Limitation of the Research

Though Ethio telecom has vast network with multiple autonomous systems, this thesis is based only on portion of Ethio telecom's access, aggregation and core networks with three autonomous systems. The scope of this research is also limited to two the performance measurement parameters, namely latency and packet loss, to measure the performance of SDN coupled seamless MPLS TE and seamless MPLS TE. Additionally, the simulation is also limited two traffic scenarios only.

1.8. Thesis Layout

This thesis is organized in seven chapters. The first chapter deals with the background, problem statement, objectives, literature review, scope and limitation of the study. The second chapter introduces MPLS, seamless MPLS architecture and its attributes. The third chapter discusses SDN and its protocols, OpenDaylight controller and Path Computation Element. The fourth chapter explains the simulation network setup used in this thesis, S-MPLS-TE simulation and SDN coupled S-MPLS-TE simulation. The fifth chapter presents latency and packet loss simulation results and their respective analysis. The six chapter draws a conclusion for the thesis. The seventh chapter points to future areas of research, which can be studied further.

2. Seamless MPLS

2.1. Introduction to MPLS

Multi-Protocol Label Switching (MPLS) is a vendor independent forwarding mechanism used to interconnect networks [6]. MPLS forwards packets among nodes on the basis of labels attached in packets instead of looking long IP addresses at every router. This reduces core routers' time by eliminating the need for complex routing table lookups. The label determines virtual switching paths between distant nodes. Hence it increases data transfer speed significantly [7]. MPLS is known as "multi-protocol" because it works with Internet Protocol (IP), Asynchronous Transport Mode (ATM) and Frame Relay network protocols [8]. The primary MPLS addressing mechanism is a label. Label is a 20-bit field found in an MPLS header lodged between the Layer 3 and Layer 2 protocol headers as shown in figure 1. For this reason, MPLS is also referred to as a Layer 2.5 protocol [6] [7] [8]. Basic MPLS concepts will be discussed below.

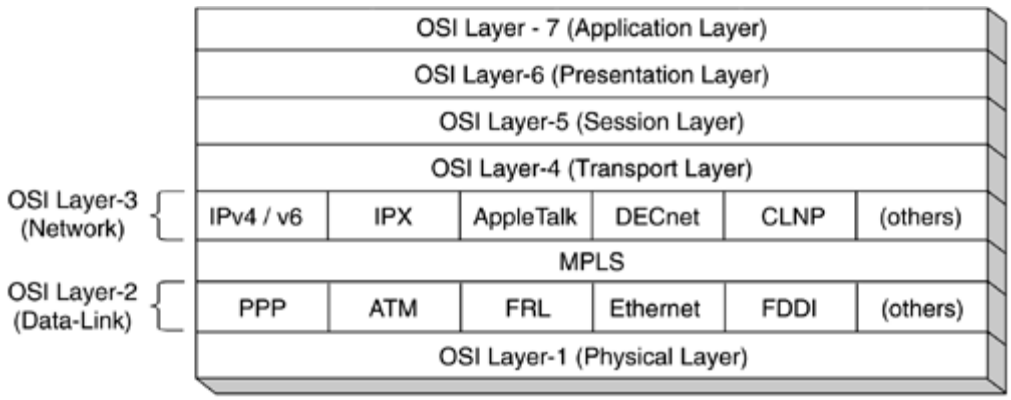


Figure 1: Position of MPLS in OSI Model [7]

2.1.1. MPLS Header

MPLS header, shown in figure 2, is structured in four parts with a total length of 32 bits, 20 bits for label, 3 bits for experiment, 1 bit for bottom of stack and 8 bits for Time to Live (TTL). MPLS Header comprises of an identifier called label. A label is a short, fixed length and locally significant identifier used for uniquely identifying Forwarding Equivalence

Class (FEC). Followed by label is an experimental field (exp) that is reserved for experimental use or often used for quality of service purpose. Stack field is used to indicate that the label is in the lower level of the stack. The label nearest to the Layer 2 header is called the top label, and the label nearest to the Layer 3 header is called the bottom label. The stack field is set to 1 if the label is the bottom label and set to 0 if not. TTL is used for MPLS loop prevention. The value of TTL is decreases by 1 at every hop when it goes through the Label Switching Routers (LSRs). When TTL value reaches to 0 the packet will be dropped [9] [10].

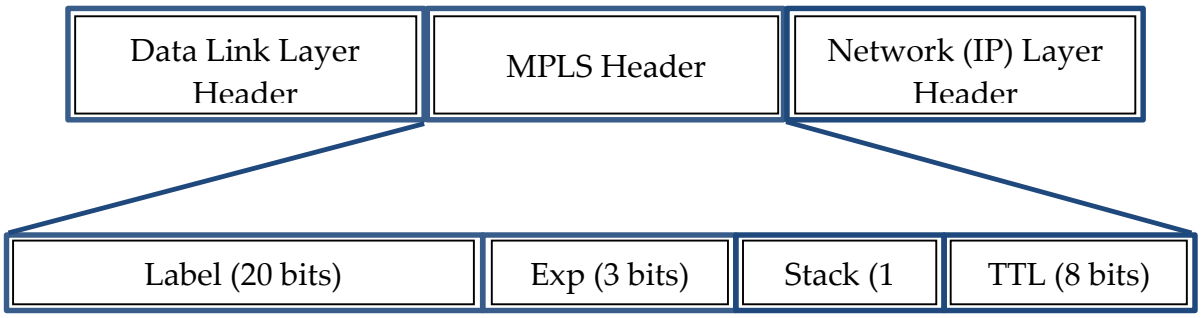


Figure 2: MPLS Packet Header [9]

2.1.2. Forward Equivalence Class (FEC)

As a forwarding technology, MPLS groups packets to be forwarded in the same manner into a class called a forwarding equivalence class (FEC). Packets of the identical FEC are handled similarly on an MPLS network. FECs classification is very flexible. Any combination of source address, destination address, source port, destination port, protocol type and VPN can be the basis for the classification [10].

2.1.3. Label Switching Router (LSR)

Label advertising and switching can be performed by Label Switching Router (LSR). The fundamental component on an MPLS network is LSR. An MPLS domain is a network consisting of LSRs. As shown in figure 3, MPLS domain has the following types of LSRs [10]:

- **Ingress LSR**—Also called Label Edge Router (LER), adds label to packets entering into the MPLS network.
- **Transit LSR**—An intermediate LSR in the MPLS network that forward packets to the egress LSR according to labels on LSP.
- **Egress LSR**—Also called Label Edge Router (LER), removes label from packets and forwards them to their destination network. Transit LSRs forward packets according to the label of the packets in the MPLS network and edge LSRs deal with the conversion between MPLS and IP [10].

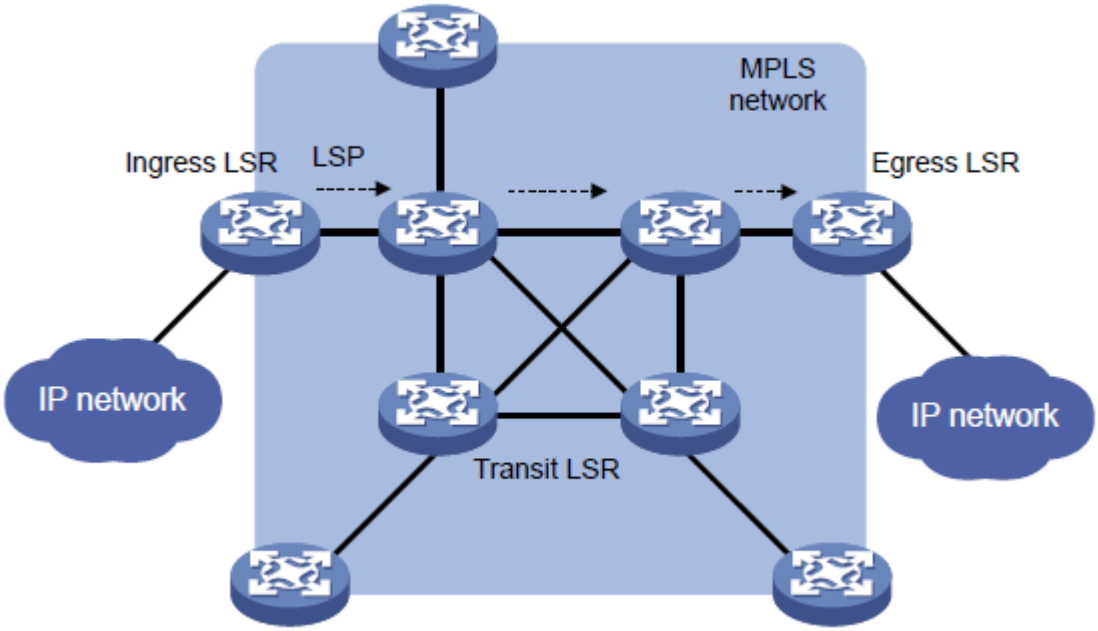


Figure 3: MPLS Network Architecture [10]

LSR does one or more of the following actions, when it receives a packet [11]:

Push: Adds a label. This is typically performed by the ingress router.

Swap: Replaces a label. This is usually performed by intermediate LSRs between the ingress and egress routers.

Pop: Removes a label. This is most often done by the egress router [11].

2.1.4. Label Switched Path (LSP)

A Label Switched Path (LSP) is the path along which packets of an FEC travel through an MPLS network [10]. It is a path through a network from an ingress to an egress router that has been established by the distribution of labels which defines hop-by-hop forwarding treatment. An LSP is defined by a sequence of labels assigned to routers on the packet's path from source to destination [12].

2.1.5. Label Distribution

When the packet enters the MPLS topology, ingress LSR receives the packet and pushes the MPLS label on to the packet and forwards it to the next hop via the LSP. 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 pops the packet label and forwards it to the destination router [13].

All the LSRs in MPLS network have Interior Gateway Protocol (IGP) (ISIS, EIGRP, RIP, OSPF, etc.) running throughout the network. 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 locally significant, that do not have global meaning across the network, a mechanism to instruct the routers on which label should they use while forwarding the packets is needed. For this purpose, label distribution protocol is necessary [13].

Label distribution protocols create and maintain the label-to-IGP prefix bindings along an LSP. A set of procedures by which one LSR informs a peer LSR about the meaning of labels used to forward traffic between them is called label distribution protocol. It enables each peer to learn about the neighboring peer's label mappings. The label distribution protocol provides the information MPLS uses to create the forwarding tables in each LSR in the MPLS domain [14].

The following protocols are currently used for label distribution [14]:

- BGP—Border Gateway Protocol
- LDP—Label Distribution Protocol
- RSVP-TE—Resource Reservation Protocol with traffic-engineering extensions.

2.1.6. MPLS Operation

All the traffic coming into the MPLS domain enters at the ingress router (LER) and leaves at the egress router (LER). At the ingress router, each packet is assigned a label and the packet is forwarded throughout the MPLS domain based on this label. At each router, the label gets swapped with another label, which represents the next router. Finally, when the packet reaches the egress router, it discards the label from the packet and forwards the packet based on the network layer header. Basic operations of MPLS are shown in figure 4 where all routers are enabled with MPLS. The steps involved are as follows [17]:

Step1: Routing protocols exchange routing information to destination networks.

Step2: Label Distribution Protocol (LDP) establishes label mappings to destination network.

Step3: Ingress LER receives packet and assigns a label to the packets based on FEC.

Step4: LSR forwards packets using label swapping.

Step5: LER at egress removes label and delivers packet based on the network layer header.

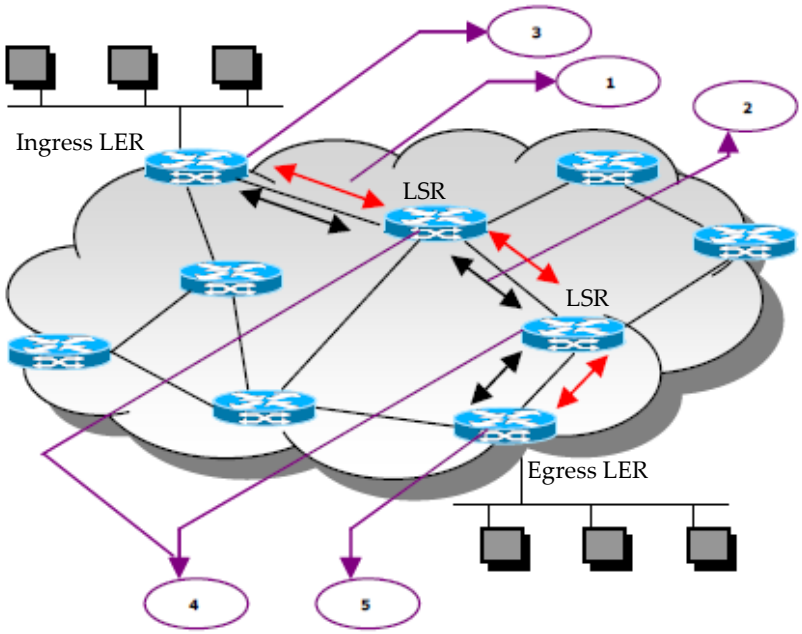


Figure 4: MPLS Basic Operation [17]

2.1.7. MPLS Traffic Engineering (MPLS-TE)

Resiliency remains an issue in telecommunication networks. Network failures can be caused by major disasters, human errors, network security threats, environmental issues etc. With the upsurge of customer demands and high traffic flows, having a telecom network, with high Quality of Service (QoS) and Quality of Resilience is of extreme importance [15] [20].

Traffic engineering (TE) is one of the fastest growing technology implemented on an MPLS backbone network. It is the process of deviating traffic from the natural path to provide better QoS among routers within the ISP. MPLS-TE can greatly enhance the packet performance of a plain MPLS network. TE allows ISPs to route network traffic to provide the best QoS to their customers in terms of throughput and delay. TE is implemented in a network infrastructure for the purpose of avoiding network congestion and improving QoS [15] [20].

MPLS-TE operates by learning about the network topology and its available resources. It provides a way to integrate TE capabilities (such as those used on Layer 2 protocols like ATM) into Layer 3 protocol (IP). MPLS TE uses an extension to existing protocols like Resource Reservation Protocol (RSVP), Intermediate System - Intermediate (IS-IS) and Open Shortest Path First (OSPF) to calculate and establish unidirectional tunnels that are created according to the network constraint. Depending on their destination, traffic flows are mapped on the different tunnels [15] [20].

The path is specified at the LSP head end in the Explicit Route Object (ERO). ERO contains the addresses of nodes through which the LSP must pass. If required, the ERO can contain the entire path that the LSP must follow from the ingress to the egress router. The path is provided by creating a tunnel originating from the ingress router to the egress router. So, the tunnels are unidirectional. If a tunnel is created for carrying traffic from A to B, then the return tunnel from B to A is not automatically created [15] [20].

2.2. MPLS Router Architecture

In Figure 5, a conceptual architecture of an MPLS label switched router with IP functionality is shown. As can be seen, there are two planes: a control plane and a data plane. At the control plane, IP routing protocols can exchange routing information that is used to forward the incoming packets.

Data plane decides how to switch the incoming packets after being received at inbound interface and manages label distribution and binding. It also maintains a Label Information Base (LIB) where remote and local bindings are stored, and creates a Label Forwarding Information Base (LFIB). The packet arriving on the data plane consults the LFIB for proper forwarding in an outgoing direction [13] [16].

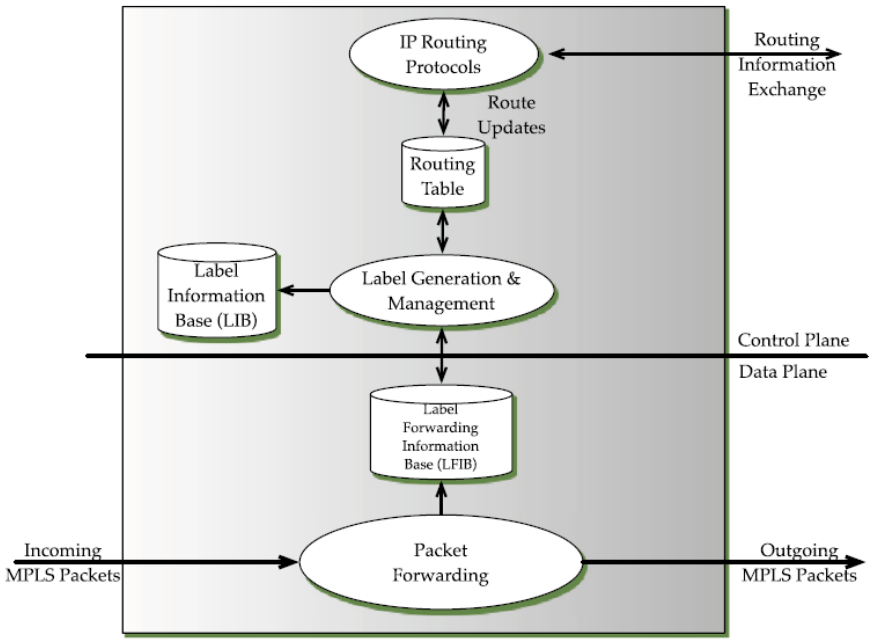


Figure 5: MPLS Router Conceptual Architecture [16]

2.2.1. 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 that include Intermediate System - Intermediate System (ISIS), Open Shortest Path First (OSPF) and Boarder Gateway Protocol (BGP), and label information exchange protocols such as Label Distribution Protocol (LDP) [13].

2.2.2. Data Plane

Data plane has a forwarding plane that is based on the information attached to labels. There are two types of tables in MPLS, namely LIB and LFIB. Label Forwarding Information Base (LFIB) is used by the data plane to forward labeled packets. The Local Information Base (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 [13].

2.3. Seamless MPLS (S-MPLS)

Crucial change in the mobile services happened on account of the shift from mobile voice calls to mobile data. As mobile standards have progressed from 2G to 3G and presently to 4G, mobile operators like Ethio telecom are feeling the squeeze to enhance mobile backhaul speed, scale, and QoS, while controlling and diminishing the expense per bit in the transport network which is an undeniably challenging assignment. While Layer 3 (L3) packet transport has been principal in the core and generally in the aggregation domain of the network, the access network has largely been based on Layer 2 (L2) transport. Therefore, industry answers for RAN backhaul across the access, aggregation and core layers of the network have been founded on mixed packet transport technologies [18]. These solutions regularly include a blend of L2 and L3 transport, with a L2 access and aggregation interfacing with a L3 core, or a L2 access interfacing with a L3 aggregation and core. Such models inherently couple the transport and service layers of the network and are subsequently tormented with both control and management plane translations at the edge of network domains. Interworking functions like VLAN into Pseud wire (PW), VLAN in to Virtual Private LAN Service (VPLS), VLAN into Virtual Private Wire Service (VPWS) and PW into MPLS Virtual Private Network (VPN) Virtual Routing and Forwarding (VRF) are regularly needed at the edge of these domains to provide the end-to-end services [18].

These interworking elements are essentially touchpoints for the service that burn-through resources on the network device deployed at this boundary. These intermediate touchpoints lead to management intricacy and require repeated provisioning as new services are added to the network. The middle touchpoints additionally complicate the operation and management of the services, since they don't allow straightforward end-to-end service monitoring and require separate L2 and L3 Operation and Maintenance technologies to be utilized in the respective segments of the service. The transport

innovation utilized in the IP RAN backhaul and the network architecture should be painstakingly designed to be versatile and scalable enough to meet the necessities of different services being transported across a huge number of areas in the network [18].

Seamless MPLS gives the design standard to making a scalable, resilient, and manageable IP backhaul framework that is optimized to seamlessly interwork with different telecom technologies. It is intended to simultaneously support numerous telecom services on a single unified network infrastructure. Seamless MPLS in backhaul networks unifies the MPLS transport across the access, aggregation, and core domains [18]. The key aspects of seamless MPLS are described in this section.

2.3.1. Seamless MPLS Architectures

Seamless MPLS employs hierarchical technique to overcome the scaling challenge in MPLS-based networks. Seamless MPLS utilizes a divide-and-conquer method where the core, aggregation, and access networks are partitioned in various MPLS/IP domains. A single-AS multi-area design or inter-AS organization could be used to segment the network between the core and aggregation domains. The seamless transport concept involves splitting the network's core, aggregation, and access layers into discrete IGP/LDP domains, regardless of the kind of segmentation [18].

By dividing these network layers into such distinct and isolated IGP domains, the size of routing and forwarding tables on individual routers in these domains is reduced, resulting in improved stability and convergence. Within each independent IGP domain, LDP is utilized for label distribution and the creation of LSPs. This allows a device in an access, aggregation, or core domain to have reachability via intra domain LDP LSPs to any other device in the same domain [18].

RFC 3107 procedures, which use BGP-labeled unicast as an inter-domain label distribution protocol to establish hierarchical LSPs across domains, are used to achieve reachability across domains. This allows the IGP in each isolated domain to keep its link state database as small as possible, while all external reachability information is sent via BGP, which is designed to scale to millions of routes. Interior Border Gateway Protocol (iBGP) labeled unicast is used to construct intra-domain LSPs in single-AS multi-area designs [18].

In inter-AS designs, iBGP-labeled unicast is used to create intra-domain LSPs inside the AS. To extend the end-to-end LSP across the AS boundary, exterior Border Gateway Protocol (eBGP) labeled unicast is utilized. In both cases, seamless MPLS transport across domains is accomplished through hierarchical LSPs that rely on a BGP-distributed label for transiting isolated MPLS domains and an LDP distributed label for reaching the intra-domain Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) corresponding to the labeled BGP next-hop within the AS. The radio access network, the aggregation network and the core network are the most common layers in SP networks, and seamless MPLS allows for a comprehensive and versatile transport framework. The type of access and the size of the network are taken into account while structuring the transport architecture [18].

2.3.1.1. Labeled BGP Core and Aggregation

Networks deployed in medium-to-large geographies apply the labeled BGP core and aggregation architecture. As shown in Figure 6, this architecture presupposes that non-MPLS IP/Ethernet or TDM access is aggregated in a comparatively large scale network. The network infrastructure is organized by segmenting the core and aggregation networks into independent IGP/LDP domains. A single-AS multi-area design or an inter-AS organization could be used to segment the core and aggregation domains [18].

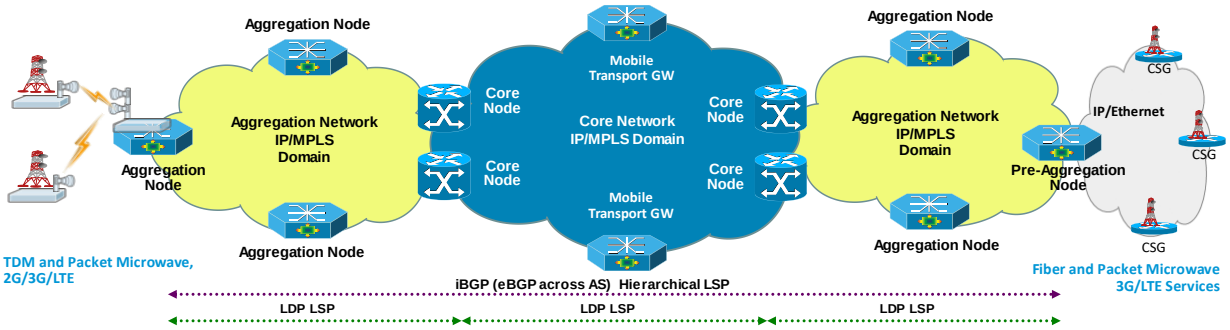


Figure 6: Labeled BGP Core and Aggregation [18]

Separation can be enabled with the single-AS multi-area option by making the aggregation network a separate IGP area from the core network or by running a different IGP process on the core ABR nodes corresponding to the aggregation and core networks. Native IP or Ethernet point-to-point or ring topologies over fiber, Ethernet microwave-based access, or point-to-point TDM+Ethernet links over hybrid microwave can be used for RAN access. The aggregation nodes are responsible for enabling all services. Within each segmented domain, LDP is used to create intra-area LSP. Labeled BGP LSPs are used to integrate the aggregation and core domains. The core ABRs implement BGP Next-Hop-Self (NHS) function to extend the iBGP hierarchical LSP across the aggregation and core domains in the single-AS multi-area option. iBGP is used to establish the hierarchical LSP from the pre-aggregation node to the ASBRs, and eBGP is used to extend the end-to-end LSP across the AS boundary when the core and aggregation networks are arranged in separate autonomous systems. At the core ABRs/ASBRs, BGP community-based egress filtering is applied to the aggregation networks, ensuring that the pre-aggregation and aggregation nodes only learn required remote destinations and drop all undesired prefixes. This reduces the size of these nodes' BGP tables while preventing unnecessary updates [18].

2.3.1.2. Labeled BGP Core, Access and Aggregation

Figure 7 shows a labeled BGP core, access and aggregation architecture for networks deployed across large geographic areas. An MPLS-enabled access network with fiber and packet microwave links aggregated in a large-scale network is assumed in this architecture. The network infrastructure is arranged by segmenting the core, aggregation, and access networks into independent IGP/LDP domains. The segmentation between the core, aggregation, and access domains could be based on a single-AS multi-area design or an inter-AS organization [18].

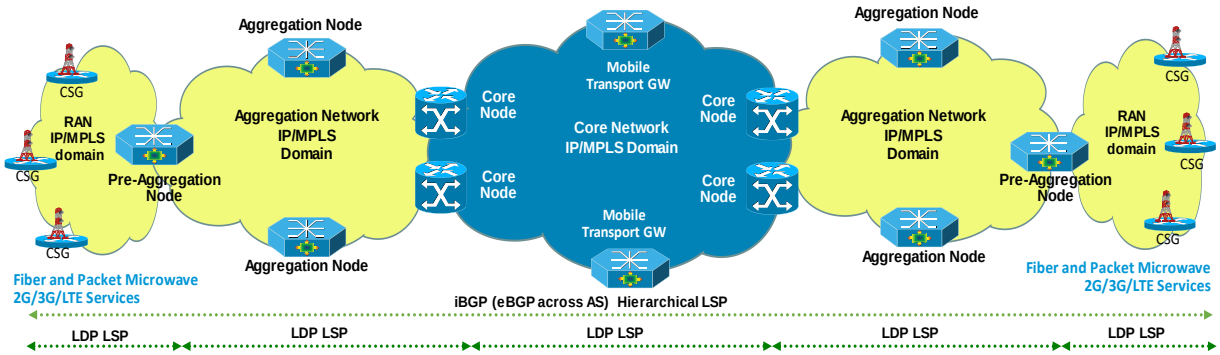


Figure 7: Labeled BGP Core, Access and Aggregation [18]

Separation between core and aggregation networks can be enabled, in the single-AS multi-area option, by making the aggregation network a separate IGP area from the core network, or by running a different IGP process on the core ABR nodes corresponding to the aggregation and core networks. Running a different IGP process on the pre-aggregation nodes, corresponding to the aggregation and RAN access networks, allows for the separation of aggregation and access networks [18].

While the core and aggregation networks are in separate autonomous systems in the inter-AS option, the separation between aggregation and access networks is enabled either by making the access network part of a different IGP area than the aggregation network, or by running a different IGP process on the pre-aggregation nodes

corresponding to the aggregation and RAN access networks. The CSGs in the access as well as the pre-aggregation/aggregation nodes enable mobile 2G/3G/LTE and wireline services. Within each segmented domain, LDP is used to create intra-area LSP. Labeled BGP LSPs integrate the access, aggregation, and core networks. The pre-aggregation nodes and core ABRs function as ABRs for their respective domains and extend the labeled iBGP hierarchical LSP across the access, aggregation, and core domains in the single-AS multi-area option. When the core and aggregation networks are organized in separate autonomous systems, the pre-aggregation nodes act as ABRs performing BGP NHS function to extend the iBGP hierarchical LSP across the access and aggregation domains. At the ASBRs, eBGP is used to extend the end-to-end LSP across the AS boundary [18].

The core ABRs perform BGP community-based egress filtering toward the aggregation networks, and the pre-aggregation node ABRs perform BGP community-based egress filtering toward the RAN access, so that the corresponding nodes in each domain only learn required remote destinations and all unwanted prefixes are dropped. This reduces the size of the BGP tables on these nodes and also prevents unnecessary updates [18].

2.3.1.3. Labeled BGP Core and Aggregation with Redistribution into Access Network IGP

Redistribution into the Access Network (AN) IGP is part of this labeled BGP core and aggregation architecture. This architecture is appropriate for networks that span large geographic areas. It assumes an MPLS-enabled access network with fiber and packet microwave links being aggregated in a large-scale network. The network infrastructure organization in this architecture model is the same as the one described in the section 2.3.1.1 with options for both single-AS multi-area and inter-AS designs, as shown in figure 8. The service options in this architecture are also the same. The CSGs in the access as well as the pre-aggregation/aggregation nodes enable mobile 2G/3G/LTE and wireline

services. This architecture differs from the previous one in that the hierarchical-labeled BGP LSP only spans the core and aggregation networks, not through the RAN access network. [18].

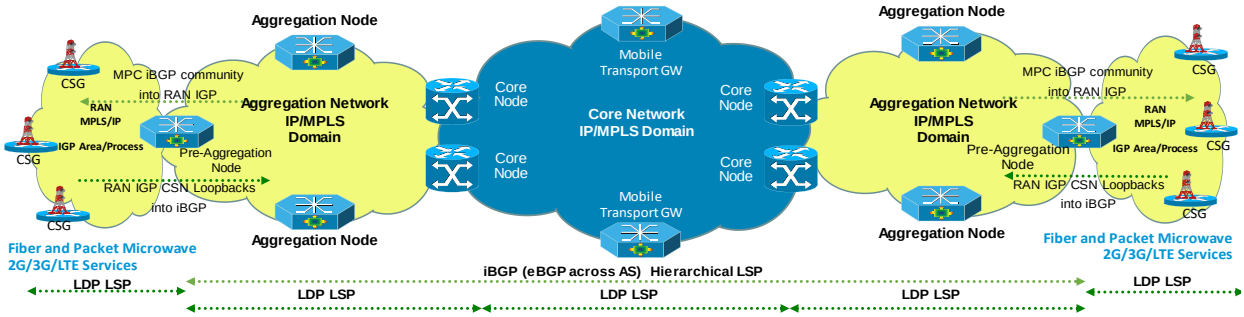


Figure 8: Labeled BGP Core and Aggregation with Redistribution into AN IGP [18]

Instead of using BGP for inter-domain label distribution in the RAN, the end-to-end seamless MPLS LSP is extended into the RAN access using LDP with redistribution in to the IGP. The IGP scale in the RAN access is kept small by selective redistribution of required remote prefixes from iBGP based on communities [18].

2.3.2. Seamless MPLS Attributes

The following are some of the primary benefits of seamless MPLS [19]:

- Scalability and resiliency (recognizing that certain nodes have limited capabilities): Access nodes can scale up to thousands of nodes and are often designed to be simple and low-cost. Seamless MPLS enables the end-to-end network to scale to over 100,000 MPLS devices while taking into account the fact that some nodes (such as access) have restricted capabilities [19].
- A single end-to-end MPLS domain: Seamless MPLS combines the aggregation and access domains into a single MPLS domain and extends the core domain. This unified domain simplifies the management and troubleshooting of the transport and services layers [19].

- A network without boundaries (seamless): MPLS-based services can be established between any two endpoints using seamless MPLS, which eliminates the need for per-service configuration in intermediary nodes [19].
- Rapid fault detection and recovery: End-to-end fault detection, fast protection, and end-to-end operations, administration, and maintenance (OAM) functions are all supported by seamless MPLS [19].
- Decoupling of transport layer from services: No matter how the underlying transport is put out, seamless MPLS allows services to be provisioned wherever they are needed [19].

2.4. Inter-AS MPLS Traffic Engineering

In MPLS-TE, information regarding all the links in the network is acquired by the distribution of link state by the IGP and is therefore confined within the same boundaries as the link state advertisements. Because the visibility of the topology and of the constraints is limited to a single IGP area or AS, TE LSPs dynamically computed by the head end are also limited in the same way. This presents a problem in big networks that deploy several IGP areas and ASs for scalability, or when services span across several service providers. Inter-AS LSPs exist both within the same provider and across different providers. Multiple ASs can be present within a single service provider's network. Separate ASs are maintained for a variety of reasons, including administrative authority and the desire to preserve routing protocol isolation between geographical domains and prevent the entire network from collapsing in the case of a local IGP failure [20].

2.4.1. Setting Up Inter-AS MPLS-TE Tunnels

Setup of inter-AS tunnels is possible without any extensions, by computing the path offline and specifying the hops at the headend using ERO (IP explicit path). A path will be signaled across ASs once it is specified. Using hop-by-hop signaling between adjacent

neighbors, an end-to-end LSP between ingress and egress LERs is created across AS boundaries. It is identical to the setup of a TE LSP within a single domain. A combination of strict and loose next-hop addresses are used to build the ERO, as fully strict explicit paths are not applicable in building inter-AS TE LSPs. Strict or loose next-hops can be used in intra-domain TE LSP building and only loose next-hops are used to traverse the TE LSP across domains through ASBR [20] [21].

In [21] ASBR forced link flooding is used to advertise links to all inter-AS nodes. When forced link flooding is configured on an interface, the MPLS TE link management module advertises the link to all nodes. As a result of this advertisement, the TE topology database on all the nodes within the inter-AS is updated with this information. ASBR forced link flooding allows the links to be advertised even if IGP adjacencies are not running over these links. TE LSPs can traverse these links at the edge of a network between two nodes running BGP even if the exit ASBR is not listed in the IP explicit path. Therefore, a headend LSR can consider that link when it computes its TE LSP path.

2.4.2. Inter-AS MPLS-TE Limitations

Setting up inter AS LSPs by computing the path offline and specifying hops at the headend has the drawback of forcing a service provider to adopt an operational model that relies on offline computation for both primary and secondary paths. Furthermore, unless the bypass tunnels protecting the inter AS links are also computed offline, the issue of fast re-route is also not addressed. In addition, strict EROs are also not applicable in inter-AS tunnel building [20] [21].

3. Software Defined Networking (SDN)

In traditional networks, the logic that determines how packets should be forwarded through the network and the mechanism that actually forwards the packets are tightly coupled. These two structures are commonly referred to as the control-plane and the data-plane (or forwarding-plane), respectively. Furthermore, the entire network structure is very decentralized. The configuration of a single device can be completely independent of all others. These characteristics are the outcome of networking technology's natural evolution. They make the network more resilient to failure and provide certain performance guarantees. In practice, however, these characteristics make managing, debugging, and reconfiguring huge networks extremely challenging [22] [23].

Software Defined Networking (SDN), aims to address these issues by radically rethinking network management. In a nutshell, SDN is an emerging networking architecture in which network control is detached from the forwarding mechanism and is directly programmable. The decoupling of data and control planes, as well as centralized control of the entire network, are the two defining characteristics of SDN. A third feature emerges from these two: programmability. New functionality can be added as needed by exposing a centralized view and control over an entire network infrastructure. The SDN controller, sometimes known as the network operating system (NOS), is the most important component of a software-defined network. To facilitate the central control logic, SDN discovers and updates a global abstract network view [22] [23].

The network OS controls the entire network state via the forwarding plane in software defined networking. SDN, as compared to traditional networks, allows network operators to define functionality after the networks have been installed [22] [23].

3.1. SDN Architecture

In SDN architecture, the control plane or network plane is separated from the forwarding hardware, and a control program or controller that makes forwarding decisions and installs rules on switches or routers is logically centralized. These rules cause packets to be switched across ports by the forwarding hardware. Figure 9 shows how SDN can be logically represented as a three-layered architecture [24].

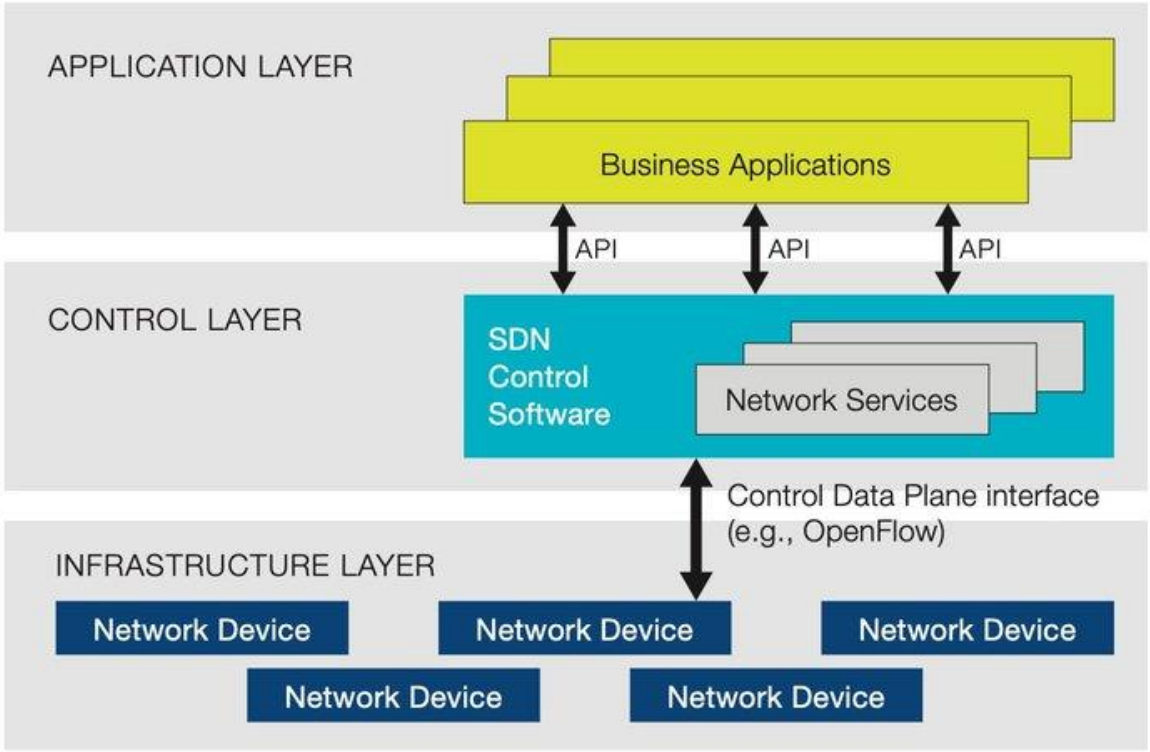


Figure 9: SDN Architecture [24]

3.1.1. Infrastructure Layer

Data plane is a term used to describe this layer. It entails forwarding hardware, such as switches and routers, as well as forwarding components and Application Programming Interfaces (APIs). Network devices perform set of elementary forwarding operations. They are programmable devices that respond to the instructions sent by the SDN

controller. The communication between SDN controller and network devices is enabled by southbound APIs. Southbound APIs enable SDN controllers to make dynamic changes in the forwarding plane in real time, allowing for more efficient network control [23] [24].

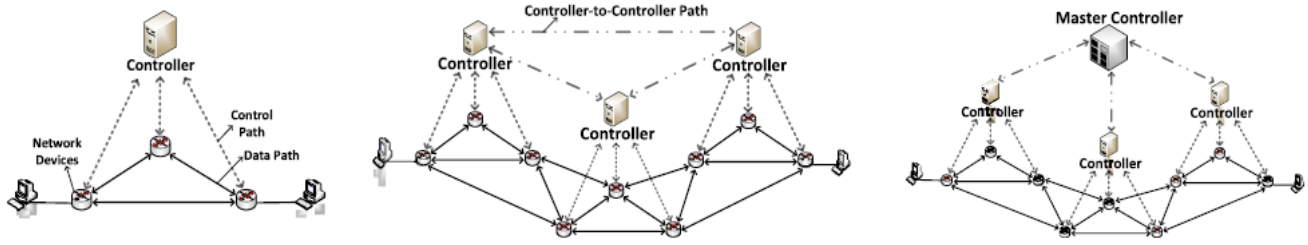
3.1.2. Control Layer

A control plane is another name for this layer. This layer contains all of the complexity that traditional network devices have as a result of numerous protocols, algorithms, and other factors. An SDN control plane is composed of a set of software called SDN controllers that provide control capabilities for network forwarding behavior supervision. The control layer manages forwarding hardware and installs forwarding rules via APIs. SDN controller is the brain of the network. It is centralized control point which manages flow control to the network devices below and the applications logic above [23] [24] [25].

SDN controller creates a network abstraction view, comprising statistics and network state, and provides it to the application level. The controller could be used to control the data plane from the application level. When an instruction is sent from the upper level, the controller receives it and passes it on to the lower level devices. The northbound API provides applications, on top of the SDN stack, with a network abstraction interface. This interface offers application-level network programmability [23] [24] [25].

The northbound API is certainly the most critical part of SDN architecture. SDN controller is valued by innovative applications it can support, and northbound API must follow application requirements. SDN controllers are also connected to automation stacks and orchestration platforms via northbound APIs [23] [24] [25].

Depending on the architecture, such as centralized, distributed, hierarchical, and so on, an SDN control plane may be made up of one or more controllers. Figure 10 illustrates the most common control plane models used in SDN technology: Centralized Control Plane (CCP), Distributed Control Plane (DCP) and Hierarchical Control Plane (HCP). These models have characteristics related to a variety of concepts, including control plane scalability, resiliency, and better manageability [25].



(a) Centralized (Single) Controller Design

(b) Distributed (Flat) Controller Design

(b) Hierarchical Controller Design

Figure 10: An Overview of Popular SDN Control Plane Models [25]

In the figure 10 above, the two-sided solid, dashed, and dashed-dotted arrows represent two-way data path among network devices, control path between controller and data plane devices, and controller-to-controller path among controllers, respectively. In figure 10a (CCP), there is one main controller with global network view. In figure 10b (DCP), every controller is in charge of separate sites/parts of the network(s), with partially shared network view. In figure 10c (HCP), there are levels where controllers reside and are accountable for distinct sites (sub-domains) as well as a master controller on top with global network view for global applications like routing [25].

- **Centralized (Single) Controller Plane Model (CCP)**: A single central controller with a global network view is used in this control plane model. As a result, the model is easy and simple to operate. This setup may be suitable for small to medium-sized networks [25].

- **Distributed Controller Plane Model (DCP)**: This approach is made up of distributed controllers that are connected to devices. Each controller is in charge of a sub-network/domain of the entire network and has its own local network view, which is abstracted as a logical node to the controllers next to it. When these controllers receive a packet bound for a destination outside of their domain, they communicate with each other (i.e., connected neighbors) to establish an end-to-end path [25].
- **Hierarchical Controller Plane Model (HCP)**: A minimum of two vertical layers make up this control plane design. While local domain controllers constitute the lower tier, additional controllers (typically referred to as master or broker) reside in the upper layer(s). The domain controllers have complete control over their domains and are not connected to one another, only to the master controller. However, a local controller does not maintain a global view of the whole network. Instead, by abstracting all domains as logical nodes, the master controller has a complete global view of the entire network [25].

3.1.3. Application Layer

In this layer, applications and services, as shown in figure 11, take over control and infra-structure layer via northbound APIs such as Representational State Transfer (REST) APIs. The application layer contains a collection of applications that leverage the northbound API's functionalities to implement operational logic and network control. One can monitor the physical network and control routing, firewalls, and load balancers from the application level. All application layer commands are translated into southbound instructions, that program the behavior of forwarding devices. This plane contains applications that are needed to provide end-users with network services. These applications communicate with a controller(s) in order to make internal decisions based on an abstract view of the network [23] [24] [25].

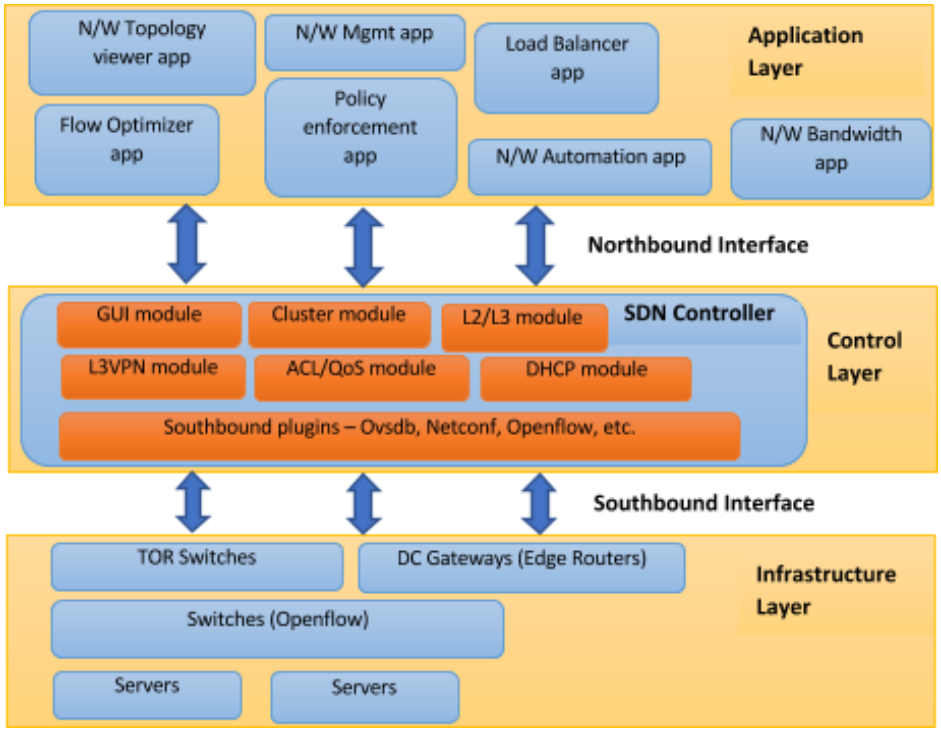


Figure 11: Components of Each Layer of SDN Architecture [29]

3.2. Southbound Interface (SBI) Protocols

The southbound interface is located between the controller and the devices on the infrastructure plane, as shown in Figure 11. SBI manages the statistical information feedback in one way and passes the configuration commands in the other. The SBI APIs enable an SDN controller to control a networking infrastructure efficiently and make dynamic changes in accordance with the real-time traffic demands and needs. The infrastructure layer is made up of network devices that use SBI protocols to expose their capabilities to the control layer. SBI protocols include BGP-LS, PCEP and NETCONF/YANG [25] [26] [27].

3.2.1. BGP-Link State (BGP-LS)

BGP Link-State (BGP-LS) is a new BGP address family that is also known as BGP for Traffic Engineering (BGP-TE). BGP-LS is a BGP extension that allows link-state information to be carried by BGP. This link state information is obtained from the IGP in

the same way as other state information is acquired, but in this case it is from the area's Traffic Engineering Database (TED). Using the BGP routing protocol, a mechanism for collecting link-state and TE information from networks and sharing it with external components has been presented [27].

This is achieved using a new BGP Network Layer Reachability Information (NLRI) encoding format. Entries in the TED are converted into routes in order to advertise them in to BGP-LS. The TED then installs these converted routes into a user-visible routing table on behalf of the corresponding IGP. Following that, BGP-speakers or Route Reflectors with all of the network's routes initiate a BGP-peer connection with an external component (such as an SDN controller) and forward all necessary data [27].

3.2.2. Path Computation Element Protocol (PCEP)

Path computation element protocol (PCEP) is a TCP based protocol that enables the use of path computation element (PCE). BGP-LS and PCEP work together as a SBI protocols in SDN. On one hand, BGP-LS provides TED information on topology and link conditions, while PCEP communicates between Path Computation Element (PCE) and Path Computation Client (PCC). PCE is an element that can be applied to a central controller (SDN controller) and calculates an optimal path, and PCCs are the clients or infrastructure devices that apply these paths [27].

Interaction is achieved through the exchange of PCEP messages running over TCP/IP. Messages are defined to initiate, maintain and terminate a PCEP session. To perform path computations, PCC and PCE first open a PCEP session within a TCP session. The PCEP session establishment includes exchanging Open and Keep Alive messages in order to agree on session parameters, such as timers and session refresh messages. The exchange of two PCEP messages: The Path Computation Request (PCReq) message and the Path Computation Reply (PCRep) message, is at the core of the protocol interaction realization.

Additional messages are also defined to handle specific events and communication errors (e.g., Notification (PCNtf) and Error (PCErr) messages). The PCEP session terminates upon the reception of a Close message [31]. In this way, the TED database is used by the PCE. Thus, PCE with information from TED, LSPDB and PCC reports has a centralized view of the entire network, and is able to calculate optimal paths, create additional LSPs, and perform load balancing [27] [31].

3.2.3. NETCONF

NETCONF defines an extendible set of operations that a NETCONF client can send to a NETCONF server. A NETCONF client typically has the role of network manager, and the NETCONF server has a role of managed network device. In many cases, there are also intermediate systems that control a particular aspect or domain. They are often referred to as "controllers." A controller acts as a server to its managers and as a client to its devices [28].

NETCONF outlines monitoring the operational status of the device and updating the device configuration. In addition, NETCONF defines a mechanism for NETCONF server implementers to define additional operations, actions, and notifications. Using this mechanism, network managers typically perform invoking management actions and subscribe to notifications from the managed device. Normally, all the data to be retrieved or updated when monitoring or configuring the device, as well as the management actions and device notifications, are described by YANG data model [28].

3.3. Northbound Interface (NBI) Protocol

The northbound interface lies between the application plane and control plane (SDN controller), and passes information pertaining to the applications' requirements so that the controller can create the best possible software network with the appropriate service quality, adequate security and the essential management for the operations to be able to

be carried out without problems. The basic protocol for these transmissions is based on the REST (Representative State Transfer) API. This application interface facilitates the transfer of necessary information for the configuration, the operations and the measurement of statistics [26].

3.3.1. RESTCONF

RESTCONF follows the REST principles. REST is a design pattern specified by Roy Fielding in his doctoral dissertation. REST in itself uses the basic HTTP methods, basically GET, PUT, POST, PATCH, and DELETE. It is based on core principles such as client-server, statelessness, cache ability, layered system and uniform interface [28].

3.4. SDN Controller

The controller is the most important component in the SDN architecture. It implements the network's rules, executes the instructions provided by the applications, and distributes them among the network's various physical layer devices. SDN controllers differ from one another in terms of programming language and platform, but they all eventually execute the same tasks: communicate with network devices utilizing SBI protocols. The SDN controller platform often includes a number of "plug - and - play" modules that can be used to perform various network functions. Inventorying what devices are on the network and their capabilities, collecting network statistics, and so on are some of the basic responsibilities. Extensions can be added to improve functionality and support more advanced features like running algorithms for analytics and composing new rules across the network [29].

OpenFlow and OpenvSwitch Database (OVSDB) are two of the most well-known protocols used by SDN controllers to interact with switches/routers. NETCONF/YANG, BGP-LS, PCEP etc. are some of additional protocols that could also be used by an SDN controller. The SDN controller contributes to the network by managing and monitoring

its state and coordinating all of its installed services. It implements the service over the network and builds the device topology and discovery mechanism that interconnects all elements. It can also calculate routes and is intelligent enough to detect a faulty link and reroute connections, ensuring the network's high availability. TCP sessions are used to connect all devices to the controller. It also provides a set of APIs that allow managing applications to access the controller's services. OpenDaylight, NOX, POX, Beacon, Floodlight, and ONOS are some of the SDN controller platforms that have been developed. OpenFlow is used as the only Southbound (SB) interface protocol by the majority of these controllers. ONOS is still in in development stage, and supports OpenFlow and NETCONF protocols [29] [30]. For this research, OpenDaylight controller is utilized because it supports BGP-LS, PCEP, NETCONF and RESTCONF.

3.4.1. OpenDaylight (ODL) Controller

OpenDaylight is a Linux Foundation-led open source project that aims to accelerate SDN adoption. Its goal is to become an open platform that all organizations can use, preventing private applications from stifling market growth, while lowering development costs. The key benefit of this concept is that it removes barriers by allowing organizations to avoid committing to specific manufacturers which may obstruct their future progress. Because it is a common platform, organizations can use technology from a variety of manufacturers. This project's platform can be deployed directly without the need for any other components. This is due to the platform's architecture, which provides a set of core functions for the apps, as well as the diverse group of collaborators that contribute to the project. The micro service architecture is one of the primary aspects that introduces this controller in an SDN. In this context, a micro service is a network protocol or service that a user wants to enable in their OpenDaylight installation, such as the BGP protocol or a AAA service (Authentication, Authorization and Accounting). By employing this architecture, the controller only implements the protocols that the user requires, and thus

lowers costs and implementations. Furthermore, this architecture enables the company to scale easily in accordance with business requirements. It allows custom development of new capabilities composed of protocols and additional network services due to its Software-Defined focus. It abstracts its capabilities at a high level so that network engineers and developers can create new applications to customize network configuration and administration. The major function of the ODL controller, therefore, is to provide centralized control of physical and virtual network devices using standards and open source protocols. It supports a wide range of protocols, including OpenFlow as well as well-known protocols like SNMP, RESTCONF, NETCONF, OVSDB, BGP-LS, and PCEP among others [29].

Figure 12 highlights an abstract architecture of ODL Helium release which is represented by the first 3 layers: the network applications and services, the controller platform, as well as the SB interfaces and protocols. All of these layers form the control plane of SDN [30].

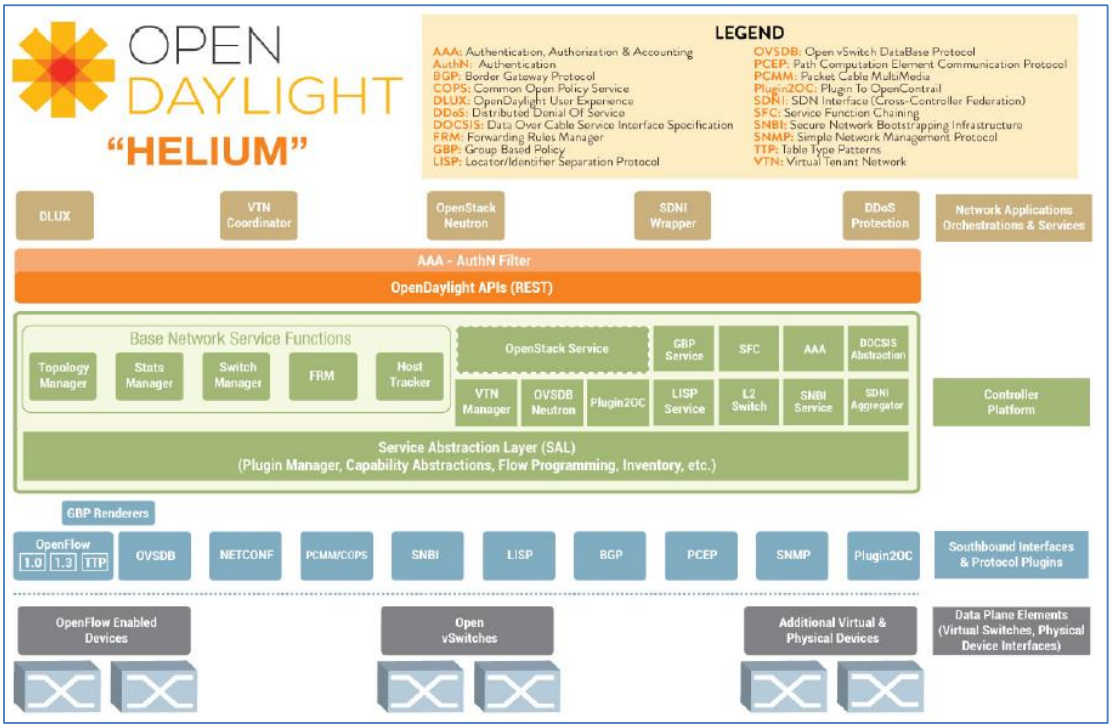
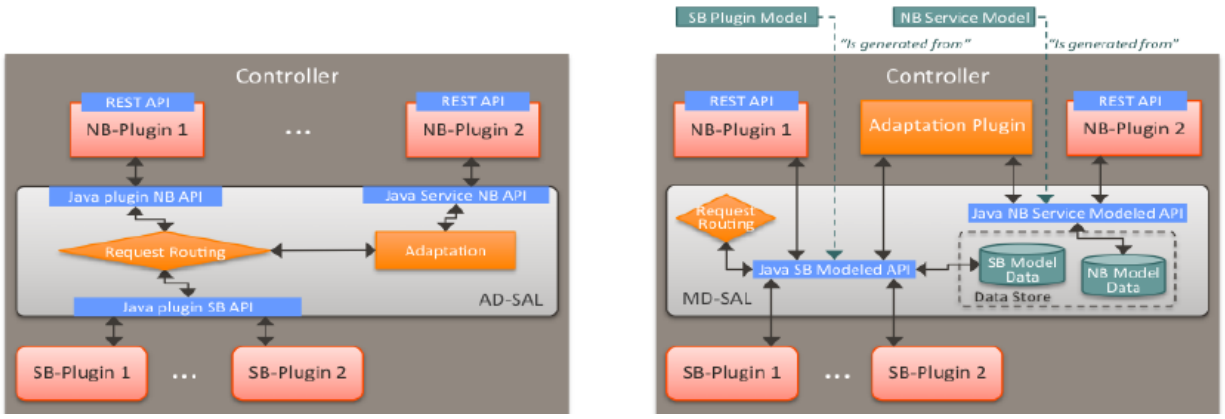


Figure 12: The Complete View of OpenDaylight Architecture [30]

The ODL is a modular, pluggable and flexible controller. SDN abstraction is enabled by the controller platform, which is the key layer in its architecture. This layer provides network applications with open Northbound (NB) APIs for controlling and managing physical and virtual network elements. It also contains the Base Network Service Functions (BNSFs), the Platform Network Service Functions (PNSF), and the Service Abstraction Layer (SAL). BNSFs are responsible for collecting statistics, collecting information about the elements within the entire network and their capabilities. They enable the collected information and statistics to be accessed by NB APIs of the applications. They include topology manager and statistics manager. PNSFs contain plug-in oriented services which perform specific networking tasks and other extensions to enhance the SDN functionality. AAA Service and L2 switch are some of the services [30].

The SAL is the heart of ODL, allowing it to support multiple SB protocols (through SB plugins) and provide uniform set of services to other modules and network applications. Device discovery is a SAL service that topology manager/exporter uses to create the network topology and build element capabilities. Most of SAL services are built based on the SB plugins features. NB and SB plugins can be service producers or service consumers or even both. The SAL functions as a massive services registry in which producers advertise their services using their APIs. When a consumer requests an advertised service by a generic API, the SAL connects and binds both the producer and the consumer. The developers of ODL started coding the original SAL with an API-Driven SAL architecture [30].

Figure 13(a) highlights the API-Driven SAL or AD-SAL. The developers had to code the SAL APIs to route service requests between consumers and providers, as well as the adaptation functionality if NB API is not similar to its corresponding SB API [30].



(a) API-Driven SAL (AD-SAL) (b) Model-Driven SAL (MD-SAL)

Figure 13: The Evolution of SAL From API-Driven to Model-Driven [30]

Regardless of AD-SAL hiding the element-level complexity, developing the SAL APIs as well as the adaption functionality for every new plugin can limit ODL's scalability. A new model-driven architecture of SAL called Model-Driven SAL (MD-SAL) is implemented as a result of the demand for a scalable architecture. As it is shown in figure 13(b), the complexity of SAL APIs is hidden by MD-SAL architecture. In addition, data is routed, utilizing java uniform APIs generated from YANG models, between consumers and producers by this architecture. XML and JSON encodings are used to represent YANG based data [30].

Mostly, SB plugin provides services. When it is complied, APIs such as Remote Procedure Calls (RPCs) and notifications are built by YANG compiler for the customers that are parts of the plugin. Additionally, data generated by models can be saved using MD-SAL data stores. MD-SAL does not constitute the adaptation functionality. Plugins that perform model-to-model mapping between couple of APIs provide the adaptation functionality. Therefore, there are no any plugin-specific APIs in MD-SAL, which shows its capability to adapt to any SB or NB plugins installed on the ODL [30].

3.4.1.1. The Southbound Interface and Protocol Plugins

To enable a secure communication between ODL controller and the network elements, SB protocols are used. These protocols can be used to manage, configure and monitor network elements. Several SB protocols (through SB plugins) are supported by ODL. ODL's support for heterogeneous networks and ensured interoperability with other technologies and between other vendors is enabled by these SB protocols. Some of the supported SB plugins that implement SB protocols are OpenFlow (OF), OpenvSwtich Database (OVSDB), BGP-LS/PCEP and NETCONF [30].

3.5. Path Computation Element (PCE)

Network operators are being pressured to adopt MPLS control plane in core networks because of the rising demand for applications and services that require flexible and assured QoS. TE capability to route traffic flows through LSPs along explicit routes is provided by MPLS. Such TE capability enables source nodes to perform path computation, subjected to additional QoS constraints, using available resource and topology information gathered through IGP. The QoS constraint can be guaranteed bandwidth in MPLS networks [31].

In LSP provisioning, one of the crucial steps to achieve TE solutions and adequate resource utilization is the path computation process. Path computation, in single domain network, is usually determined by the routing process at the source node. Resources, reserved during the signaling phase, are exploited through distributed protocol such as the reservation protocol with TE extensions (RSVP-TE) [31].

The separation between the two operations, in dynamic network scenarios, could lead to sub-optimal TE solutions, inducing waste of network resources in general. Furthermore, heavy processing at each control plane node could be required to compute path in distributed manner, particularly when it is based on multiple constraints [31].

Additional issues, such as restricted topology visibility due to scalability, arise when there is a move from single-layer single domain to multi-layer multi domain networks. Furthermore, when multiple carriers are involved in a path computation, where source and destination of a traffic request belong to different administrative domains, open advertisement of detailed intra-domain network resources is not permitted due to the need to preserve information confidentiality across domains [31].

Inter-domain TE performance, in terms of the overall network resource utilization, will be affected due to the considerable complication of path computation by such limitations. Manual provision of QoS guaranteed applications across multiple domains typically relies on sub-optimal solutions and usually need several weeks [31].

Significant research is being done in the context of core networks' control plane as a result of these drawbacks on the path computation for provisioning operations. A set of techniques, defined under the parasol of the Path Computation Element (PCE) architecture, are proposed by the Internet Engineering Task Force (IETF). Path computation by dedicated network entities (PCEs) is the basis for these techniques [31].

Fundamentally PCE is a path computation engine capable of computing a path for a TE LSP. Computed TE LSP will fulfill several constraints such as TE LSP bandwidth, destination, priority and other attributes. The set of available network resources for a given topology or a set of network topologies, is the basis for these constraints. No assumption is made on the nature of the algorithm used for path computation. The PCE gathers link-state information and carry out path computation on behalf of network nodes [31] [32].

The Path Computation Element Protocol (PCEP) facilitates the communication between the network devices, referred to as Path Computation Client (PCC), and the PCE. The PCE, in its own domain, has full visibility and updated information on available network

resources and is responsible for the path computation. In a multi domain scenarios, cooperation between PCEs takes place by sharing the result of each intra-domain path computation expressed as, for instance, border node(s) to traverse, encrypted intra-domain routes and metric values. The whole source-destination path will be provided by the combination of these results without additional information exchange among different domains. Specific techniques and procedures have been proposed and defined for each specific scenario [31] [32].

There are two modes of operation for PCE [32]:

- **Stateless:** In this mode of operation, PCEs process path computation requests independently without considering previous path computation requests and established TE LSPs. The network topology's available network resources (i.e. TED) and the requested TE LSP attributes are used by a stateless PCE. As a result, a PCE may allocate the same resources to two non-synchronized path computation requests. This could lead to LSP set-up failure, if the available network resources are not sufficient to satisfy both requests [32].
- **State full:** In this mode, another database referred to as the LSP database, in addition to the TED, is maintained by the PCEs to augment their knowledge of the network state. All established LSPs will be included in the LSP database. During processing path computation requests, Statefull PCE performs efficient path computation by utilizing both databases [32].

3.5.1. PCE Architecture

The PCE architecture relies on two functional elements: the PCE and the PCC. The PCE is in charge of carrying out constraint-based path computation for path requesting PCCs. It can be implemented on a dedicated server. On the other hand, PCC is implemented on a Network Management System (NMS) or on network device (e.g. LSP requesting source

node). A PCE may also request path computation to other (peer or hierarchically higher) PCEs, behaving as a PCC. The two functional elements communicate through PCEP. Figure 14 illustrates the typical PCE internal architecture [31].

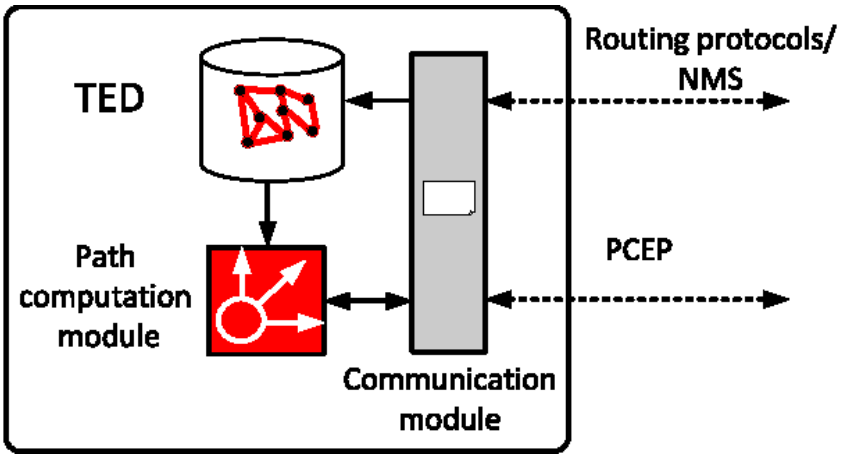


Figure 14: PCE Internal Architecture [31]

Traffic Engineering Database (TED), the path computation module and the communication module are the main component modules of the architecture. The TED collects information about network topology and current TE information (e.g. nodes, prefixes, link bandwidth utilization), populated by routing protocols (e.g. OSPF-TE, BGP-LS) or other mechanisms (e.g. NMS). Computation of requested paths based on selected algorithms and specific policies is the responsibility of the path computation module. The communication module acts as interface to handle communication protocols (e.g. routing protocols and PCEP). The PCE performs path computation based on its internal TED, that contains the network topology and the current TE information [31].

4. Simulation Network Setup

In this chapter, the simulation network setup for the performance analysis of S-MPLS-TE and SDN coupled S-MPLS-TE will be presented. The simulation network setup, depicted in figure 15, for this thesis is based on Ethio telecom's core, aggregation and access networks. GNS-3 is used as emulation software to simulate S-MPLS-TE and SDN coupled S-MPLS-TE. Lenovo L480 laptop computer with 16GB of RAM and 500GB storage is used as a host for the simulation. Both S-MPLS-TE and SDN coupled S-MPLS-TE simulation network setup have three independent network domains each with its own AS.

The first access and aggregation domain is based on Bole area aggregation and access networks with an AS umber of 65109. This domain is composed of one access (Bole Cell Service Gateway (BL-CSG)) and two aggregation (Bole Radio Service Gateways (BL-RSG-1 and BL-RSG-2)) routers with a single ostinato traffic generator connected to the access router. SDN capable Cisco IOS XRv routers are used for the BL-CSG emulation. BL-SCG is connected to the two Cisco IOS C7200 aggregation routers (BL-RSG-1 and BL-RSG-2). BL-RSG-1 and BL-RSG-2 are also connected to one other to increase reliability for the access network traffic, and to the Bole core domain router (BL-NE40E-A) in the core domain. All connection among routers in this domain and with the core domain is based on Gigabit Ethernet (GE) interfaces.

The second core domain is composed of five provider routers (Back Bone Routers (BR-1, BR-2, BR-3, BR-4 and BR-5)) and two provider edge routers (Bole PE (BL-NE40E-A) and Bahir Dar PE (BD-ER-A)) with an AS number of 65108. All routers in this domain are Cisco IOS C7200. Provider routers in core domain are connected in mesh fashion (i.e. each router is connected to every router). All connections among routers in this domain and with the Bole and Bahir Dar access and aggregation domains are also based on Gigabit Ethernet (GE) interfaces.

The third access and aggregation domain is based on Bahir Dar access and aggregation networks with an AS number of 65113. This domain is composed of one access (Bahir Dar Cell Service Gateway (BD-CSG)) and two aggregation (Bahir Dar Radio Service Gateways (BD-RSG-1 and BD-RSG-2)) routers with a one ostinato traffic generator connected to the access router. SDN capable Cisco IOS XRv routers are used for the BD-CSG emulation. BD-SCG is connected to the two Cisco IOS C7200 aggregation routers (BD-RSG-1 and BD-RSG-2). BD-RSG-1 and BD-RSG-2 are also connected to one other to increase reliability for the access network traffic, and to the Bahir Dar core domain router (BD-ER-A) in the core domain. All connection among routers in this domain and with the core domain is based on Gigabit Ethernet (GE) interfaces.

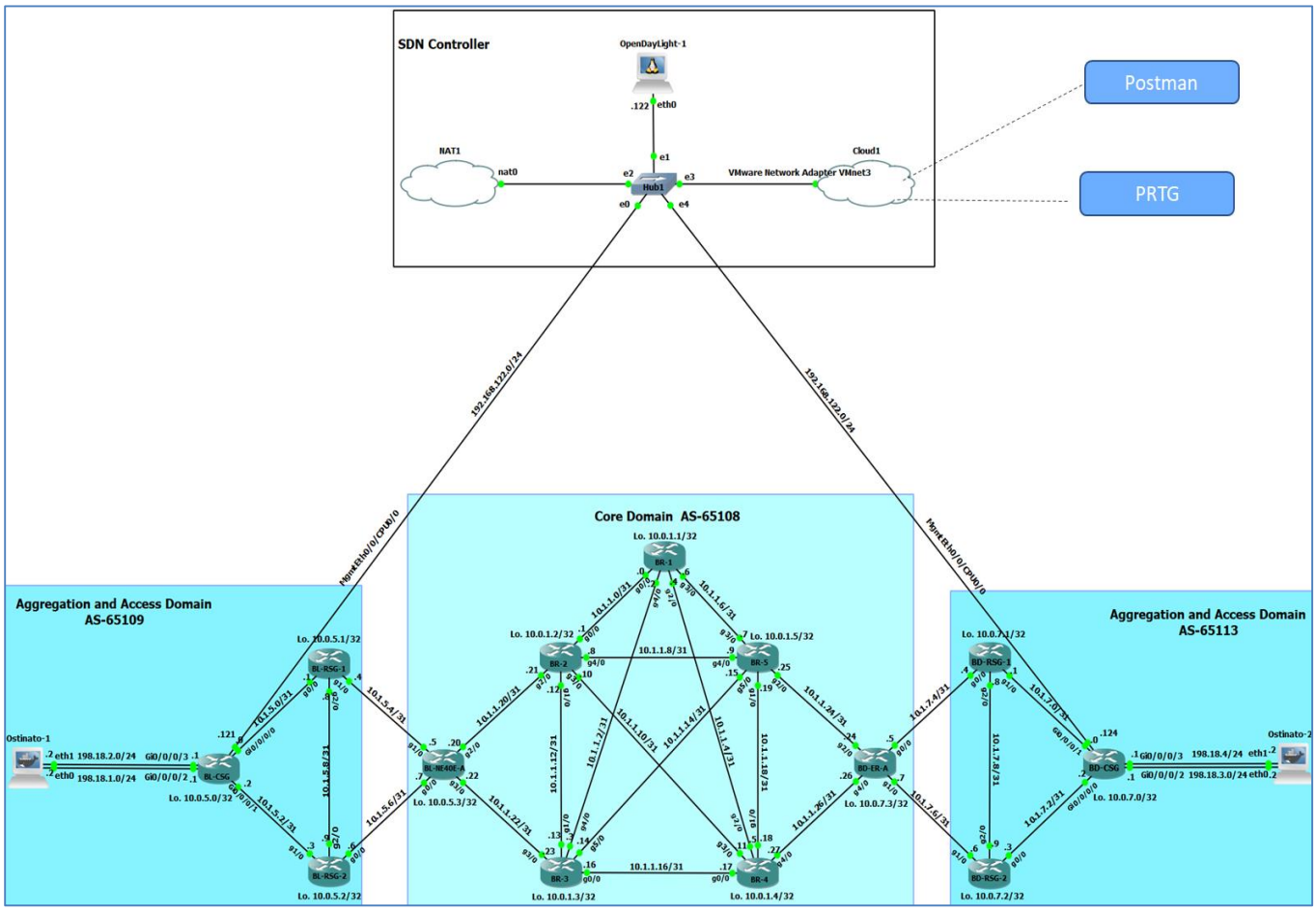


Figure 15: Simulation Network Setup

For the SDN part of the simulation, OpenDaylight (ODL) controller is used. The ODL controller deployed version is Boron Stable Release 0.5.3 (Boron SR-3). To host ODL controller GNS-3 Ubuntu Docker/container is utilized. The controller is connected to the S-MPLS-TE simulation setup through a hub. This hub also offers connectivity to the NAT cloud which provides internet connectivity for the Ubuntu Docker/container and to the Cloud-1 that provides host connectivity for the simulation network setup. Cloud-1 provides the simulation network setup access to Postman and PRTG. Postman is used to program the ODL controller. To measure the performance of both simulations PRTG network monitoring tool is employed.

4.1. S-MPLS-TE Simulation

S-MPLS-TE is simulated on the three domains of the network setup. This is achieved by configuring the loopback and link IP addresses, IGP (ISIS), MPLS/LDP, BGP-LU and TE on each router in the network setup. The IP address planning is as follows:

- In Bole access and aggregation domain (AS-65109), 10.0.5.0/32, 10.0.5.1/32 and 10.0.5.2/32 are allocated for loopback addresses of BL-CSG, BL-RSG-1 and BL-RSG-2 respectively. In this domain, point-to-point link connectivity is achieved by dividing 10.1.5.0/28 in to /31 segments.
- In Core domain (AS-65108) 10.0.1.0/32, 10.0.1.1/32, 10.0.1.2/32, 10.0.1.3/32, 10.0.1.4/32, 10.0.1.5/32, 10.0.5.3/32 and 10.0.7.3/32 are allocated for loopback addresses of BR-1, BR-2, BR-3, BR-4, BR-5, BL-NE40E-A and BD-ER-A respectively. With regard to point-to-point link connectivity, 10.1.1.0/27 is divided in to /31 segments for IP address of each link in the domain.
- In Bahir Dar access and aggregation domain (AS-65113), 10.0.7.0/32, 10.0.7.1/32 are 10.0.7.2/32 allocated for loopback addresses of BD-CSG, BD-RSG-1 and BD-RSG-2 respectively. Point-to-point link connectivity, in this domain, is achieved by dividing 10.1.7.0/28 in to /31 segments.

Since the network setup is segmented in to three independent AS domains, each domain will run its own IGP. For this simulation IS-IS is used as an IGP. It is the protocol used in Ethio telecom network. IS-IS provides loopback to loopback reachability among the routers in the respective AS domain. Figures 16, shows sample IS-IS neighborship in Bole access and aggregation domain.

```
RP/0/0/CPU0:BL-CSG#show isis neighbors
Thu Jun 10 10:38:26.182 UTC

IS-IS BLIPRAN neighbors:
System Id      Interface        SNPA           State Holdtime Type IETF-NSF
BL-RSG-1       Gi0/0/0/0        ca01.4674.0008 Up    8        L2   Capable
BL-RSG-2       Gi0/0/0/1        ca02.2b38.001c Up    7        L2   Capable
```

Figure 16: IS-IS Neighborship in Bole Access and Aggregation Domain

Next, LDP is used to build intra-AS LSPs. MPLS LDP is configured on all intra-AS routers of all domains. Figures 17 shows sample MPLS LDP neighborship Bole access and aggregation domain.

```
RP/0/0/CPU0:BL-CSG#show mpls ldp neighbor brief
Thu Jun 10 11:08:21.119 UTC

Peer                 GR  NSR  Up Time     Discovery        Addresses       Labels
                                          ipv4  ipv6       ipv4  ipv6      ipv4  ipv6
-------------------- --- ---  ---------- ----------       ----------      ------------
10.0.5.1:0           N   N    00:47:57   1     0          4     0         12    0
10.0.5.2:0           N   N    00:47:56   1     0          4     0         12    0
```

BL-RSG-1

BL-RSG-2

Figure 17: MPLS LDP Neighborship in Bole Access and Aggregation Domain

In order to traverse these intra-AS LSPs in to the neighboring AS domain, BGP Labeled Unicast (BGP-LU) is used. BGP-LU unifies the access, aggregation and core domains as if they are a single MPLS domain (i.e. the MPLS is seamless (S-MPLS)). This helps to create end-to-end hierarchical LSPs. S-MPLS is realized by configuring BGP-LU on all intra-AS routers and Autonomous System Border Routers (ASBRs). Figure 18 below shows sample BGP-LU configuration on BL-CSG.

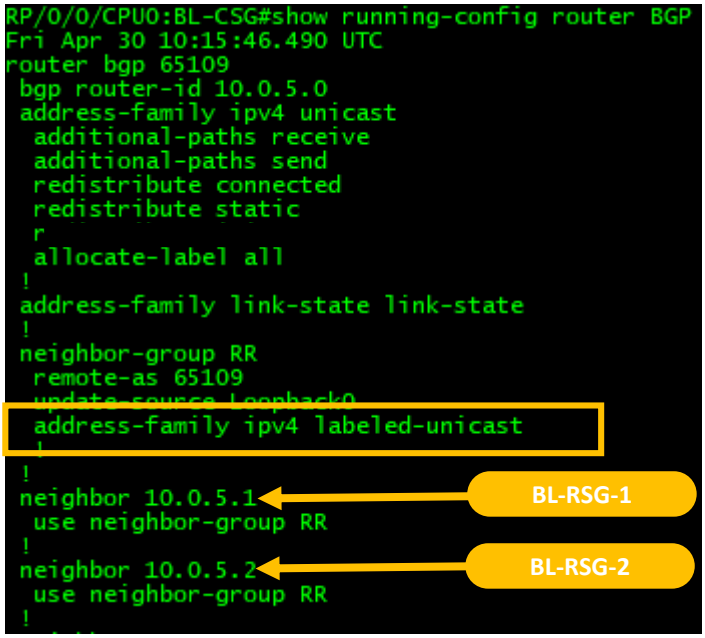


Figure 18: Sample BGP-LU Configuration on BL-CSG

In this this simulation, the ASBRs are BL-RSG-1, BL-RSG-2, BL-NE40E-a, BD-ER-A, BD-RSG-2 and BD-RSG-1. For Bole Access and Aggregation domain BL-RSG-1 and BL-RSG-2 are the ASBRs. The ASBRs act as Route Reflectors (RRs) for BL-CSG; and form iBGP-LU neighborship with BL-CSG and with one another. Similarly, Bahir Dar Access and Aggregation domain's ASBRs are BD-RSG-1 and BD-RSG-2, that are configured as RRs for BD-CSG; and form iBGP-LU neighborship with BD-CSG and with one another. In the same way, Core domain ASBRs, BL-NE40E-A and BD-ER-A, create iBGP-LU neighborship with all routers in the domain. They also act as RRs for core domain routers.

To extend end-to-end LSP across AS boundaries, eBGP-LU is used at the ASBRs. Routers BL-RSG-1 and BL-RSG-2 form an eBGP-LU neighborship with BL-NE40E-A. In similar fashion, routers BD-RSG-1 and BD-RSG-2 form an eBGP-LU neighborship with BD-ER-A. Figures 19 shows sample BGP-LU neighborship in Bole access and aggregation domain.

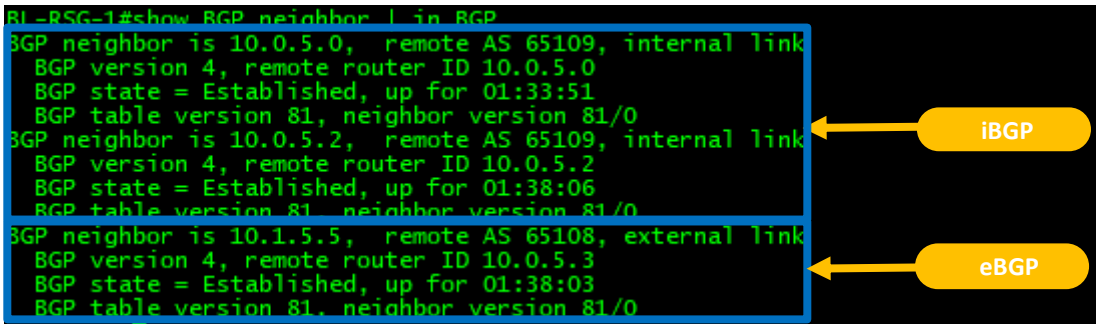


Figure 19: iBGP and eBGP Neighborship in Bole Access and Aggregation Domain

As such, all loopbacks from all domains are labeled with MPLS labels. This illustrates the realization of S-MPLS across all domains.

The last step in S-MPLS-TE simulation network setup is TE configuration. MPLS TE is configured globally on each domain's S-MPLS router and specifically on interfaces that are members of the IGP (IS-IS). Since MPLS TE tunnels are restricted to single IGP domain, they cannot cross to other AS domains. So as to traverse TE tunnels over AS boundaries, ASBR forced flooding is configured on all ASBR interfaces without IGP adjacencies. ASBR forced flooding advertises links in to the neighboring AS domains. There by updating TE topology databases of all the nodes within the inter-AS domains. Figure 20 below shows sample ASBR forced flooding configuration on BL-RSG-1 interface GE1/0, which links BL-RSG-1 to BL-NE40E-A (Core domain's ASBR).

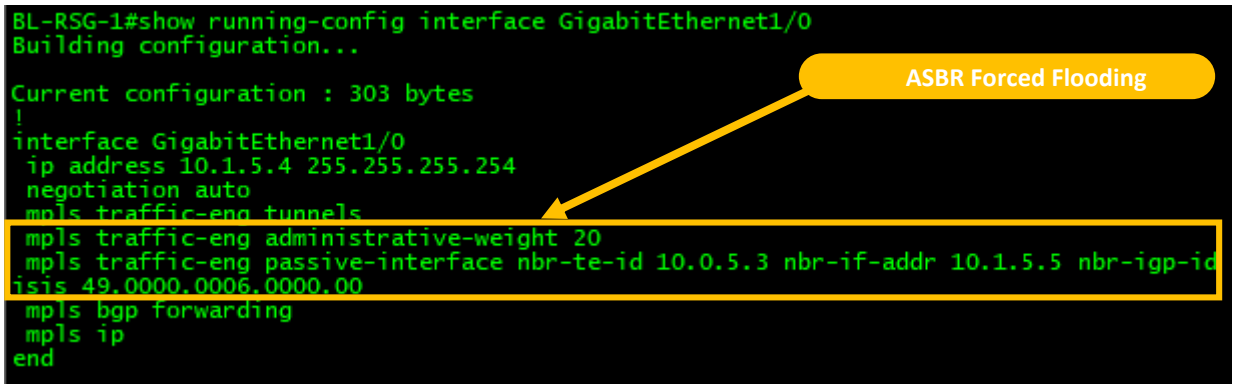


Figure 20: Sample ASBR Forced Flooding Configuration on BL-RSG-1

To simulate TE tunnel on S-MPLS-TE network setup, ERO is configured on Head End (HE) S-MPLS router. For this simulation TE tunnel te-21 is configured on the HE router, i.e. BL-CSG. The source of the S-MPLS-TE tunnel te-21 is BL-CSG and the destination of the tunnel is BD-CSG. The ERO for tunnel te-21 contains next-hop (NH) addresses spanning the three AS domains. The ERO is configured as a mix of strict and loose hops, as dynamic EROs are not possible in inter-AS TE tunnels. In addition, strict only EROs are also not possible in inter-AS TE tunnels. This is because path computation in inter-AS TE is limited to single IGP domain. Figure 21 shows TE tunnel te-21 configuration on BL-CSG destined to BD-CSG. Figures 22 and 22 show tunnel te-21 ERO configuration and tunnel status.

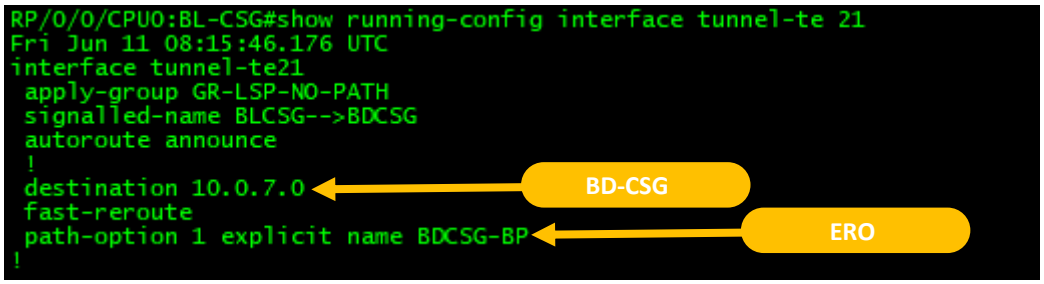


Figure 21: S-MPLS-TE Tunnel te-21 Configuration on BL-CSG

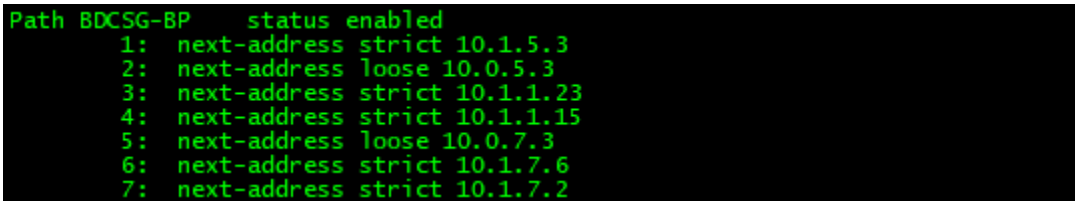


Figure 22: ERO of S-MPLS-TE Tunnel te-21 Configuration on BL-CSG

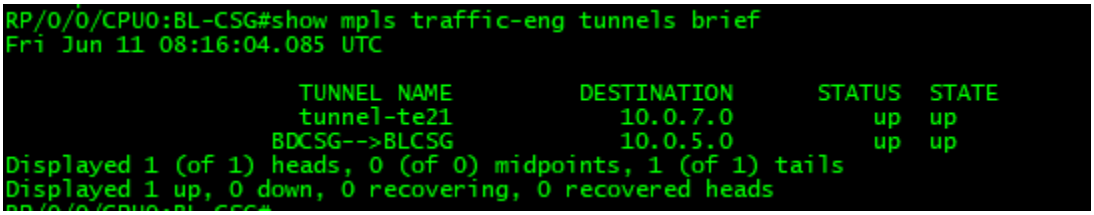


Figure 23: Status of S-MPLS-TE Tunnel te-21 on BL-CSG

Now, two traffic scenarios are deployed to measure the performance of S-MPLS-TE tunnel te-21. In the first scenario, 250kbps TCP traffic is generated by Ostinato-1 traffic generator through BL-CSG. In the second scenario, 350kbps of TCP traffic is generated by Ostinato-1 traffic generator through BL-CSG. In both scenarios, generated traffic is forwarded through BL-CSG tunnel te-21 destined to BD-CSG. The performance of the tunnel te-21 is measured in both scenarios independently for one hour using PRTG network monitoring platform. Latency and packet loss are used as performance parameters. The result of the performance measurement is presented in the next chapter.

4.2. SDN Coupled S-MPLS-TE Simulation

To simulate SDN coupled S-MPLS-TE, ODL controller is installed on Ubuntu Docker/Container and coupled with the S-MPLS-TE network setup emulated in the previous section. ODL Boron release, version SR-3 Karaf distribution 0.5.3, is downloaded and installed on Ubuntu Docker on GNS3. Figure 24 shows the CLI of ODL controller.

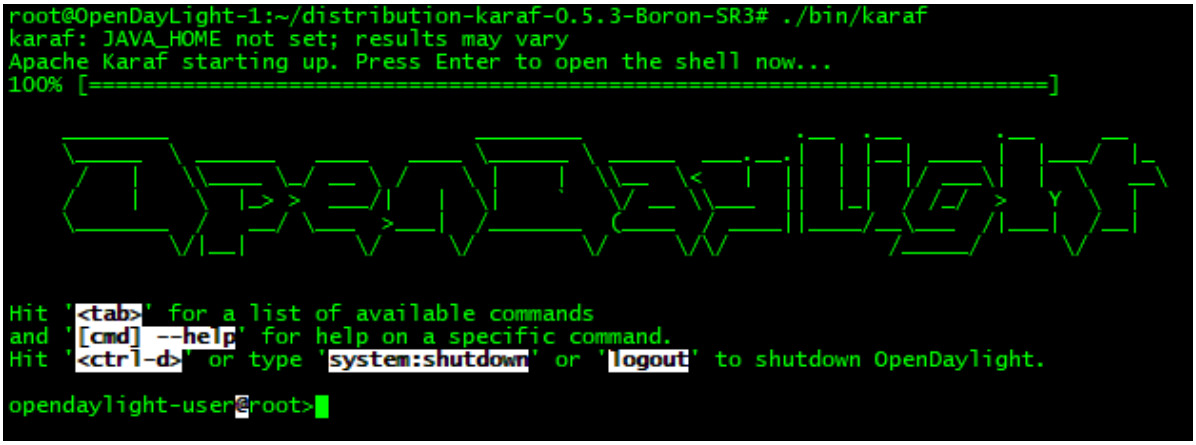


Figure 24: ODL Controller CLI

To couple ODL controller with the S-MPLS-TE network setup, 192.168.122.0/24 IP address segment is used. 192.168.122.122/24 is assigned for the ODL controller, configured on Ubuntu Docker. For HE routers BL-CSG and BD-CSG, 192.168.122.121/24

and 192.168.122.124/24 are assigned respectively. These IP addresses are configured on the management interfaces of the respective routers. In this simulation, ODL controller is used as PCE and the HE routers, i.e. BL-CSG and BD-CSG, are PCCs. The ODL controller gathers TE topology on the HE routers by establishing BGP-LS neighborship with the PCCs. To create, modify and remove TE tunnels on S-MPLS-TE network setup, PCEP peer connection between the ODL controller and PCCs is required. In order to perform these operations, the following configurations are done on ODL controller and HE routers. On ODL controller, prior to any configuration, various ODL controller features are installed. These features include odl-bgpcep-pcep, odl-bgpcep-bgp, odl-mdsal, odl-yangtools, odl-netconf, and odl-restconf. odl-bgpcep-pcep and odl-bgpcep-bgp features provide BGP-LS and PCEP plugin for ODL controller. odl-mdsal feature enables model-driven service abstraction for the controller. odl-yangtools feature enables yang data model based abstraction on the controller. odl-netconf feature provides SB plugin that enables the controller to configure the PCCs. odl-restconf provides NB API plugin which facilitates the programming of the controller by Postman. Figure 25 shows some of the features installed on ODL controller.

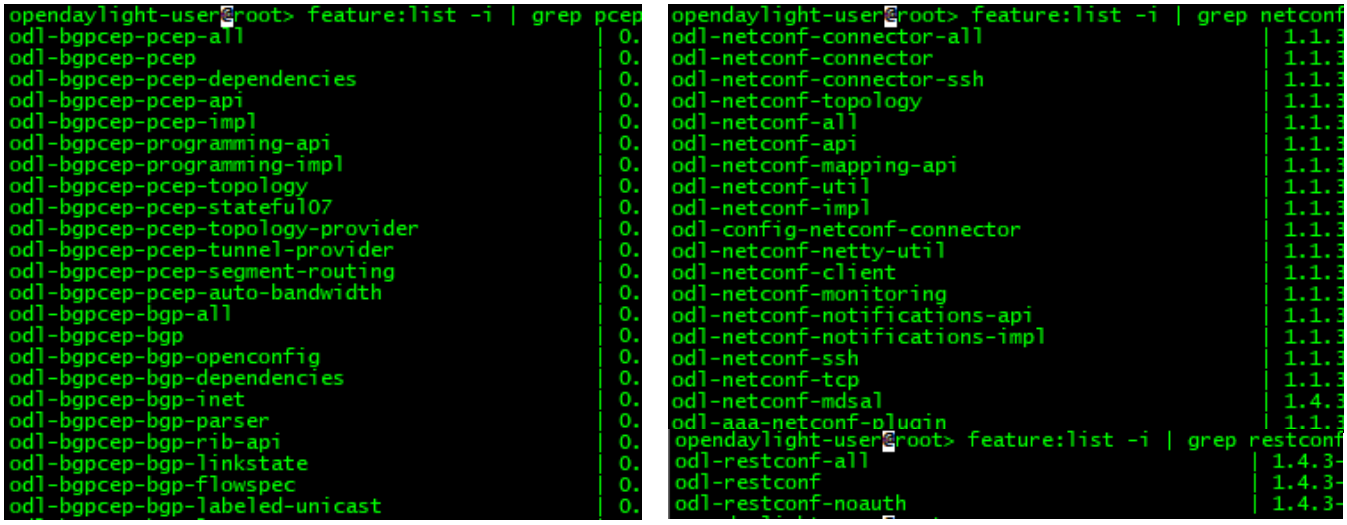


Figure 25: Installed ODL Features (BGP, PCEP, NETCONF and RESTCONF)

To establish, from the controller side, BGP-LS neighborship and PCEP peering with the PCCs, the ODL controller is configured by updating configuration files in karaf directory. Karaf directory files are shown in figure 26.

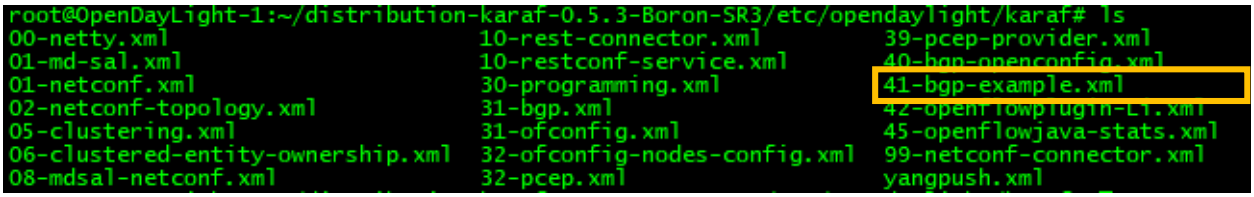


Figure 26: ODL Controller Configuration Files in Karaf Directory

The modules in 41-bgp-example.xml configuration file are example-bgp-rib, which sets BGP Routing Information Base (RIB) of ODL controller, and example-bgp-peer which connects ODL controller to BGP-LS peer (PCCs). example-bgp-rib module is updated with RIB ID 192.168.122.122 (IP address of ODL controller) and local AS 65109 (AS number of Bole Access and Aggregation domain). example-bgp-peer module is updated with IP addresses of BGP-LS speakers to which the ODL controller connect. BGP-LS speakers (PCCs), in this simulation, are S-MPLS-TE HE routers (BL-CSG and BD-CSG). BL-CSG's management interface IP (192.168.122.121) and BD-CSG's management interface IP (192.168.122.121) are configured as BGP-LS peers on example-bgp-peer module. BL-CSG is configured as iBGP-LS peer, since both the ODL controller and BL-CSG are in identical AS. On the other hand, BD-CSG is configured as eBGP-LS peer with remote AS 65113. Figures 27 and 28 show BGP RIB ID and BGP-LS peer configuration on example-bgp-rib and example-bgp-peer modules respectively.

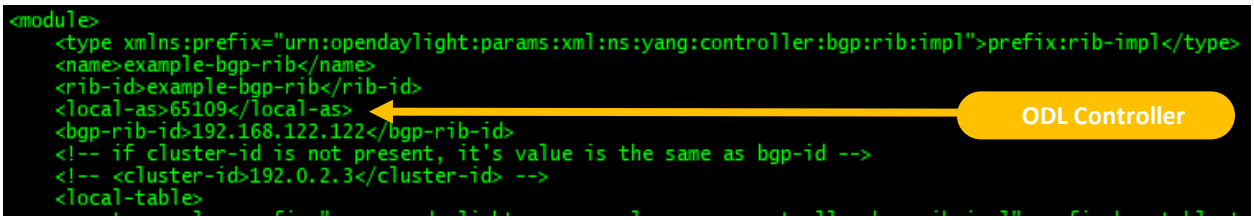


Figure 27: BGP RIB ID Configuration on example-bgp-rib module

```
<module>
    <type xmlns:prefix="urn:opendaylight:params:xml:ns:yang:controller:bgp:rib:impl">prefix:bgp-peer</type>
    <name>example-bgp-peer</name>
    <host>192.168.122.121</host>
    <holdtimer>180</holdtimer>
    <retrytimer>10</retrytimer>
    <peer-role>ibgp</peer-role>
    <rib>
<module>
    <type xmlns:prefix="urn:opendaylight:params:xml:ns:yang:controller:bgp:rib:impl">prefix:bgp-peer</type>
    <name>example-bgp-peer</name>
    <host>192.168.122.124</host>
    <holdtimer>180</holdtimer>
    <retrytimer>10</retrytimer>
    <remote-as>65113</remote-as>
    <peer-role>ebgp</peer-role>
```

BL-CSG (PCC)

BD-CSG (PCC)

Figure 28: BGP-LS Peer Configuration on example-bgp-peer module

To realize the coupling, the PCCs also need additional BGP-LS, PCE, NETCONF and SSH configurations. BGP-LS configuration on both S-MPLS-TE HE routers is achieved by adding ODL controller (192.168.122.122) as BGP neighbor to the global BGP configuration with Link-State address family. Figure 29 shows BGP-LS configuration on BL-CSG.

```
RP/0/0/CPU0:BL-CSG#show running-config router BGP
Fri Apr 30 10:15:46.490 UTC
router bgp 65109
 bgp router-id 10.0.5.0
 address-family ipv4 unicast
  additional-paths receive
  additional-paths send
  redistribute connected
  redistribute static

  allocate-label all
 !
 address-family link-state link-state
 !
 neighbor-group RR
  remote-as 65109
  update-source Loopback0
  address-family ipv4 labeled-unicast
  !
 !
 neighbor 10.0.5.1
  use neighbor-group RR
 !
 neighbor 10.0.5.2
  use neighbor-group RR
 !
 neighbor 192.168.122.122
  remote-as 65109
  update-source MgmtEth0/0/CPU0/0
  address-family ipv4 unicast
  !
  address-family link-state link-state
```

ODL Controller

BGP LS

Figure 29: BGP-LS Configuration on BL-CSG

Figure 30 below shows BGP-LS neighborship between BL-CSG and ODL controller.

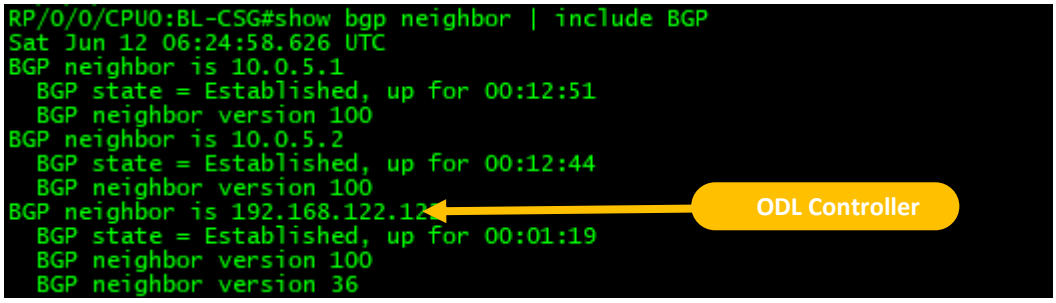


Figure 30: BGP-LS Neighborship Between BL-CSG and ODL Controller

PCEP peering between ODL controller (PCE) and S-MPLS-TE HE routers (PCCs) is attained by adding PCE configuration on the traffic engineering configuration of PCCs. Figures 31 and 32 show PCE sample configuration and PCE peering on BL-CSG (PCC).

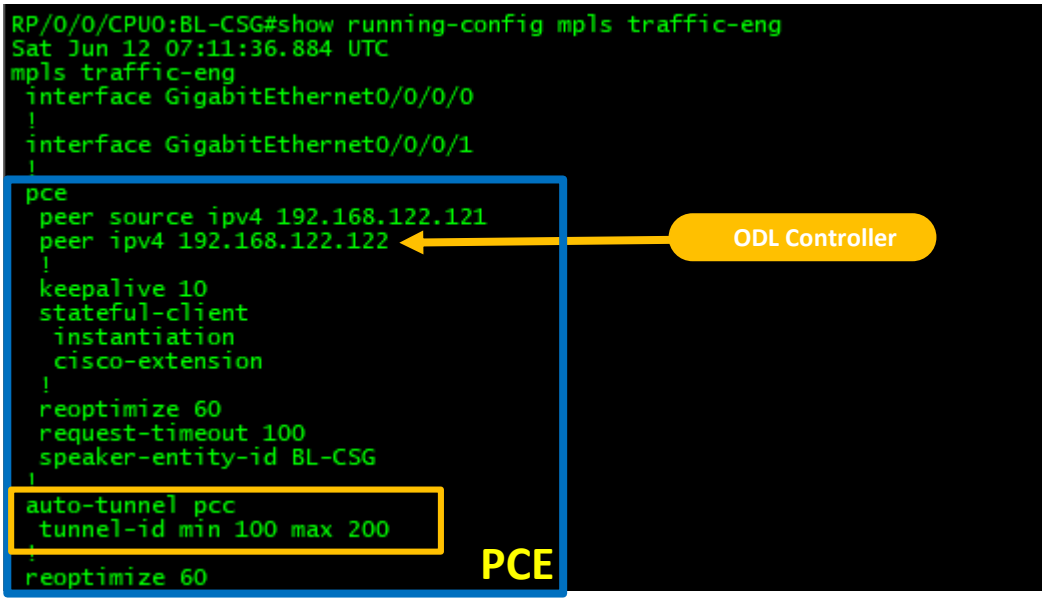


Figure 31: PCE Configuration on BL-CSG

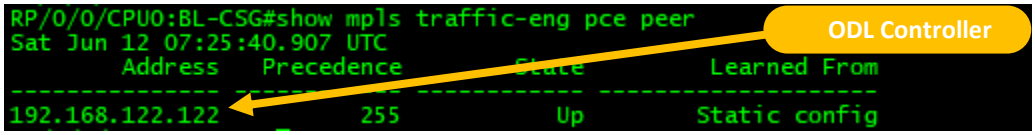


Figure 32: PCE Peering Between ODL Controller and PCC on BL-CSG

Now, the ODL controller has BGP-LS neighborship and PCEP peering with the PCCs. But the ODL controller needs to be able to configure the PCCs. This is acomplished by first allowing NETCONF and SSH at the control planes of the PCCs. And then

configuring NETCONF and SSH on the PCCs, and mounting the PCCs on the ODL controller. Figure 33 shows sample SSH and NETCONF configuration on BL-CSG.

```
control-plane
 management-plane
  out-of-band
   interface all
    allow SSH
    allow NETCONF
   !
```

```
netconf-yang agent
 ssh
!
ssh server v2
ssh server vrf default
ssh server netconf vrf default
ssh timeout 120
```

Figure 33: Sample SSH and NETCONF configuration on BL-CSG

NETCONF allows the ODL controller to configure the PCCs. SSH enables the PCCs to be mounted on the ODL controller. This mounting provides secure communication between the ODL controller and the PCCs. Postman is used to mount the PCCs on the ODL controller. Postman uses REST POST request to mount PCCs through RESTCONF, a NB API plugin installed feature on the controller. The request is in XML format. Figure 34 shows Postman mounting of BL-CSG on to ODL controller. Figure 35 shows NETCONF and SSH sessions created on BL-CSG, after it is mounted on ODL controller.

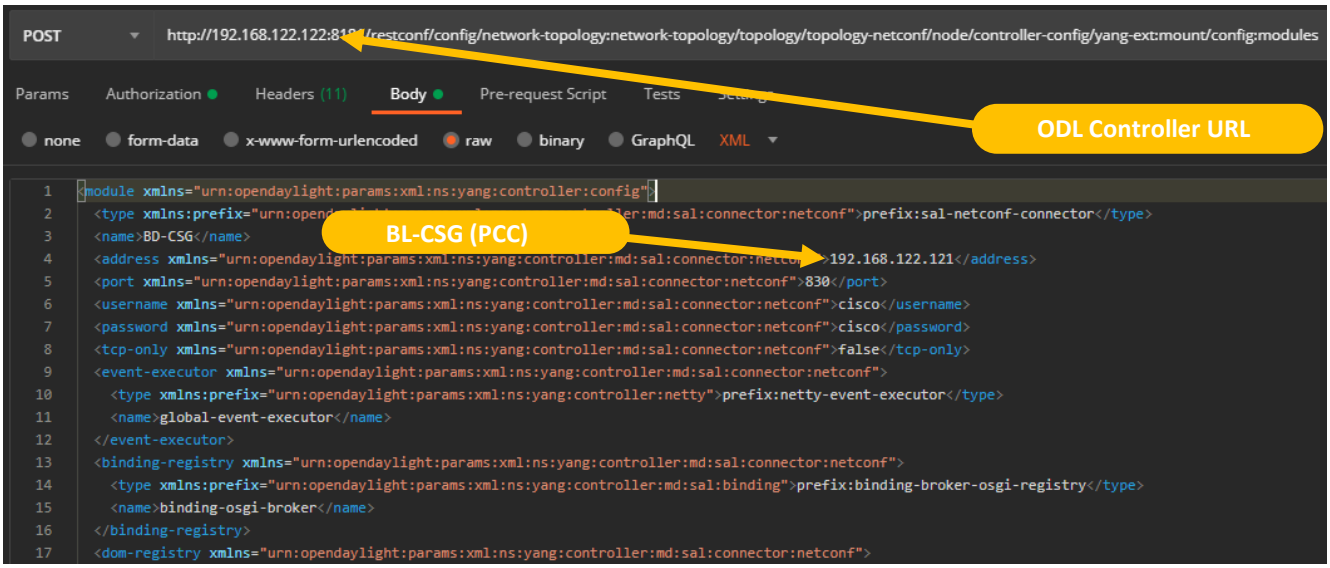


Figure 34: Mounting BL-CSG on ODL Controller Using Postman

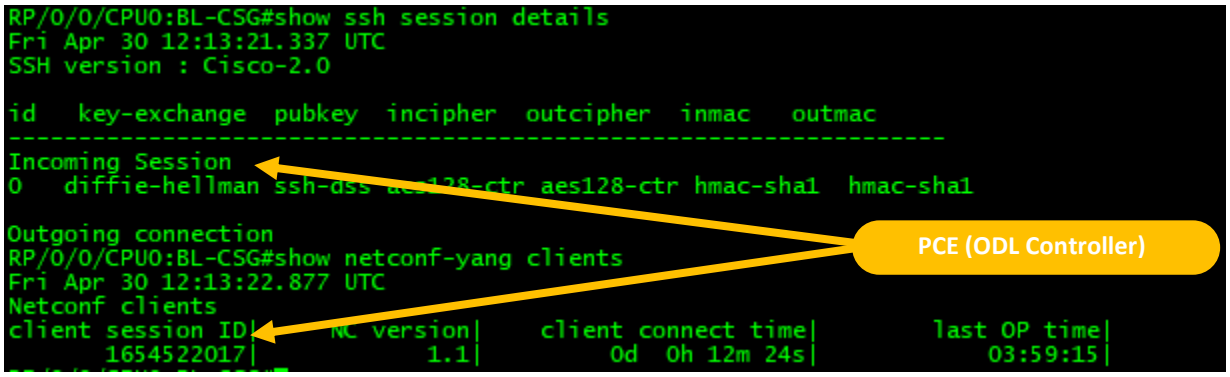


Figure 35: SSH and NETCONF Sessions on BL-CSG

After coupling ODL controller with S-MPLS-TE network setup, the next step is to simulate ODL controller coupled S-MPLS-TE tunnel by programming the ODL controller. The ODL controller is programed using Postman. It uses XML and JSON formats to exchange data between Postman and itself. The ODL controller in turn configures the PCC according to the program, through NETCONF. To simulate TE tunnel on ODL coupled S-MPLS-TE network setup, REST POST request is sent by Postman in XML format. The REST POST request's body has the PCC (BL-CSG and BD-CSG) IP address, signaled name of the TE tunnel, source IP address of the TE tunnel, destination IP address of the TE tunnel and the ERO through which the tunnel traverses the three domains. In this simulation TE tunnel te-100 is programmed on the ODL controller. The TE tunnel has the name tunnel te-100 because of the PCE auto tunnel range configuration on BL-CSG, shown in figure 31. The source of the ODL controller coupled S-MPLS-TE tunnel te-100 is BL-CSG and the destination of the tunnel is BD-CSG. The ERO for tunnel te-100 contains next-hop (NH) addresses spanning the three AS domains. The ERO only has strict hops, unlike TE tunnel te-21 simulated in S-MPLS-TE simulation. Figure 36 shows the REST POST request sent by Postman to program tunnel te-100 on the ODL controller. TE tunnel te-100 is programmed to span from BL-CSG to BD-CSG. Figure 37 shows the status of ODL controller coupled S-MPLS-TE tunnel te-100 on BL-CSG.

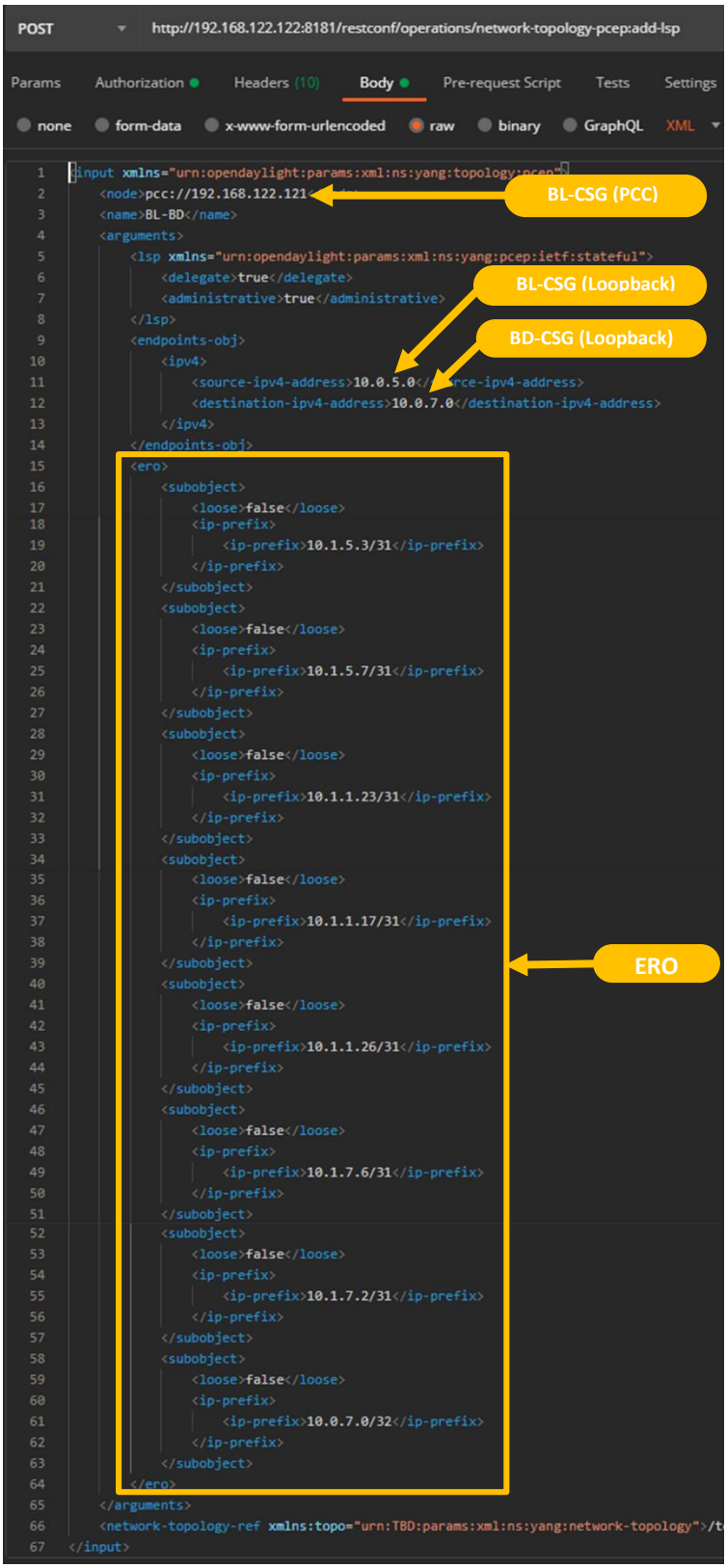


Figure 36: Postman REST POST Request

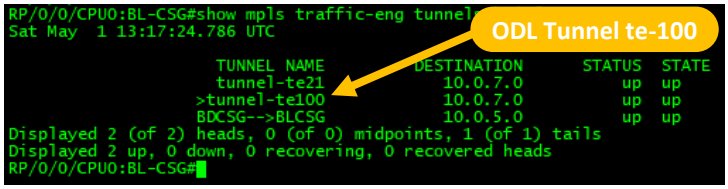


Figure 37: Status of ODL Controller Coupled S-MPLS-TE Tunnel te-100 on BL-CSG

Finalized SDN coupled S-MPLS-TE simulation network setup is depicted in figure 38.

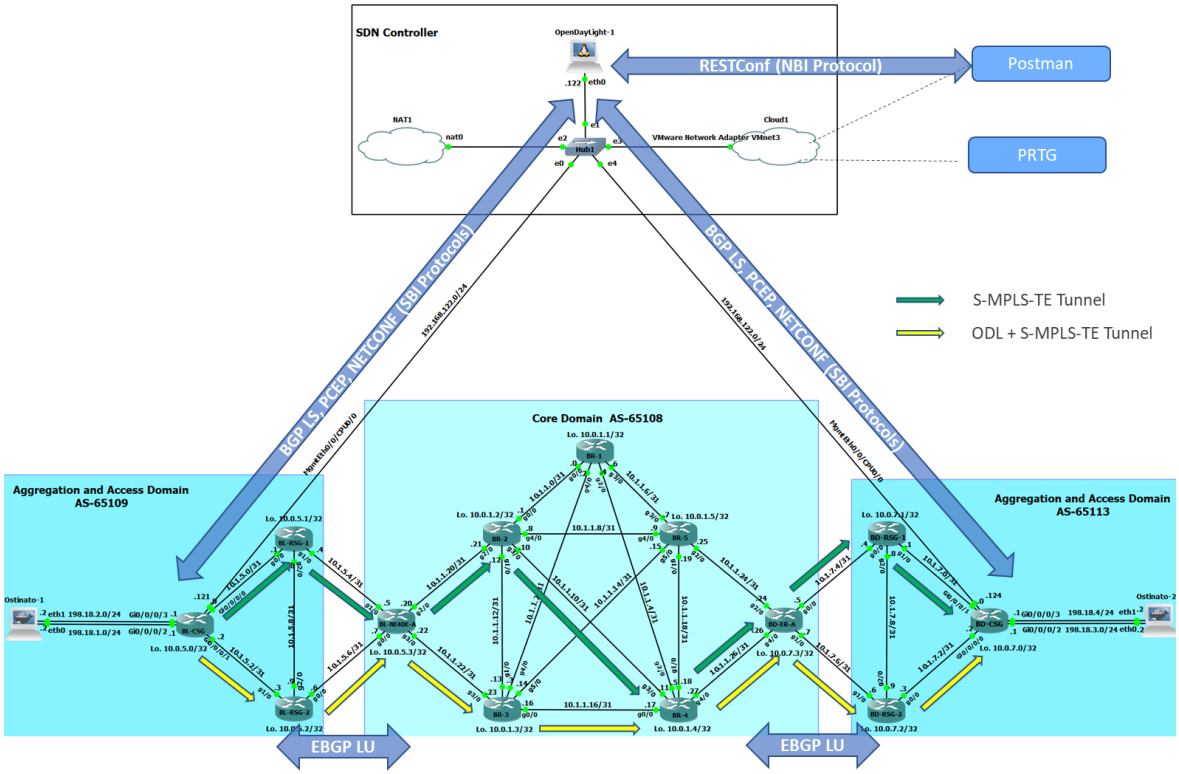


Figure 38: SDN Coupled S-MPLS-TE Simulation Network Setup

To measure the performance of ODL controller coupled S-MPLS-TE tunnel te-100, two traffic scenarios are deployed. In the first scenario, 250kbps TCP traffic is generated by Ostinato-1 traffic generator through BL-CSG. In the second scenario, 350kbps of TCP traffic is generated by Ostinato-1 traffic generator through BL-CSG. In both scenarios, generated traffic is forwarded through BL-CSG ODL configured tunnel te-100 destined to BD-CSG. The performance (i.e. latency and packet loss) of tunnel te-100 is measured in both scenarios independently for a duration of one hour using PRTG network monitoring platform. The result of the performance measurement is presented in the next chapter.

5. Simulation Result and Analysis

In this chapter, the results of S-MPLS-TE and SDN coupled S-MPLS-TE simulations carried out in chapter 4 are presented and analyzed. As detailed in chapter 4, two scenarios are considered during the simulations. The first scenario is implemented by generating 250Kbps of TCP traffic using Ostinato and forwarding it through the tunnels for a period of one hour. This traffic is forwarded through S-MPLS-TE tunnel te-21 and ODL controller coupled S-MPLS-TE tunnel te-100 independently. Then, the performance of both tunnels is measured. For this thesis, latency and packet loss are used to measure the performance of both tunnels. These performance parameters are measured, for both tunnels, using PRTG network monitoring platform, for the duration of both scenarios. The measurement is carried out in 5-minute intervals. Tables 1 and 2 show instantaneous values of latency and packet loss for S-MPLS-TE tunnel and ODL controller coupled S-MPLS-TE tunnels respectively, during the 250Kbps traffic scenario.

Table 1: Latency of S-MPLS-TE and ODL coupled S-MPLS-TE Tunnels for 250Kbps

Sampling Time	S-MPLS-TE Tunnel Latency (in msec.)	ODL+S-MPLS-TE Tunnel Latency (in msec.)
5 min	134.8	104.4
10 min	128.4	105.6
15 min	125.2	113
20 min	123	110.2
25 min	104.4	111.2
30 min	128.6	98.4
35 min	128	116.8
40 min	124.8	111.2
45 min	121.2	94.6
50 min	108	109.4
55 min	118	106.4
60 min	129.2	111.8

Table 2: Measured Packet Loss for S-MPLS-TE and ODL controller coupled S-MPLS-TE Tunnels at 250Kbps

Sampling Time	S-MPLS-TE Tunnel Packet Loss (%)	ODL+S-MPLS-TE Tunnel Packet Loss (%)
5 min	0%	0%
10 min	0%	0%
15 min	0%	0%
20 min	0%	0%
25 min	0%	2%
30 min	0%	0%
35 min	0%	0%
40 min	0%	0%
45 min	6%	0%
50 min	0%	2%
55 min	0%	0%
60 min	1%	2%

Similarly, in the second scenario, 350Kbps of TCP traffic generated is by Ostinato-1 and is forwarded through both tunnels, for a duration of one hour. This traffic is forwarded through both S-MPLS-TE tunnel te-21 and ODL controller coupled S-MPLS-TE tunnel te-100, independently. After the generation and forwarding of the 350Kbps TCP traffic, the performance of both tunnels is measured.

In the same way as in the first scenario, latency and packet loss are also used to measure the performance of both tunnels in the second scenario. These performance parameters are measured, for both tunnels, using PRTG network monitoring platform, for the duration of the second scenario. The measurement is performed in 5-minute intervals. Tables 3 and 4 show instantaneous values of latency and packet loss for S-MPLS-TE and ODL controller coupled S-MPLS-TE tunnels respectively, during the 350Kbps traffic scenario.

Table 3: Latency of S-MPLS-TE and ODL coupled S-MPLS-TE Tunnels for 350Kbps

Sampling Time	S-MPLS-TE Tunnel Latency (in msec.)	ODL+S-MPLS-TE Tunnel Latency (in msec.)
5 min	143.6	127.6
10 min	139.2	133.4
15 min	136.8	104.4
20 min	141	105.6
25 min	132.6	113
30 min	114	110.2
35 min	131.2	111.2
40 min	127.8	98.4
45 min	132.2	116.8
50 min	134.2	111.2
55 min	128.8	94.6
60 min	130.2	109.4

Table 4: Measured Packet Loss for S-MPLS-TE and ODL controller coupled S-MPLS-TE Tunnels at 350Kbps

Sampling Time	S-MPLS-TE Tunnel Packet Loss (%)	ODL+S-MPLS-TE Tunnel Packet Loss (%)
5 min	2%	0%
10 min	0%	0%
15 min	0%	0%
20 min	0%	4%
25 min	0%	0%
30 min	0%	0%
35 min	0%	0%
40 min	0%	0%
45 min	3%	2%
50 min	0%	1%
55 min	4%	0%
60 min	0%	0%

5.1. Latency Analysis

In this section, latency of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels presented in tables 1 and 3 are analyzed. Figure 39 shows latency of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels in the first scenario (i.e. at 250Kbps of traffic), plotted against the sampling time. As shown in figure 39, latency of S-MPLS-TE tunnel has a maximum of 134.8 msec. measured at the 5th minute and a minimum of 104.4 msec. measured at the 25th minute. The average latency of S-MPLS-TE tunnel is 122.8 msec. On the other hand, figure 39 also shows maximum latency of 116.8 msec. at 35th minute and a minimum of 94.6 msec. at 45th minute for ODL coupled S-MPLS-TE tunnel. An average latency of 107.75 msec. is recorded for ODL coupled S-MPLS-TE tunnel. This shows that, there is 12.26% latency reduction in average in ODL coupled S-MPLS-TE compared to that of S-MPLS-TE. Figure 39 also shows that ODL coupled S-MPLS-TE tunnel instantaneous latency exceeded the latency of S-MPLS-TE tunnel at the 25th and 50th minute sampling times.

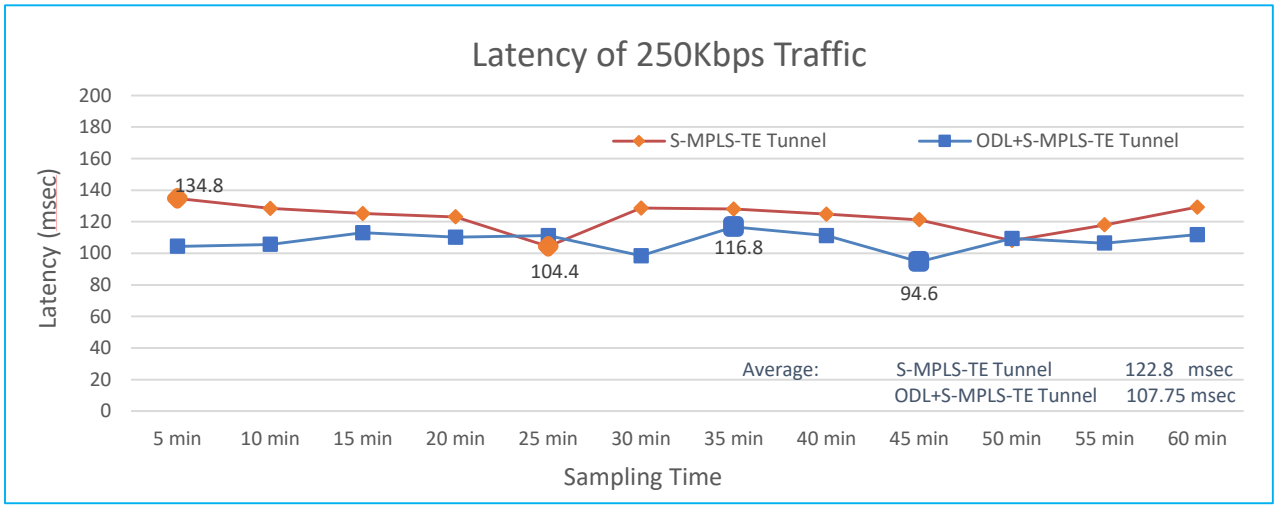


Figure 39: Latency of S-MPLS-TE and ODL + S-MPLS-TE Tunnels at 250Kbps

Figure 40 shows latency of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels in the second scenario (i.e. at 350Kbps of traffic), plotted against the sampling time. Figure 40 shows a maximum latency of 143.5 msec. at the 5th minute and a minimum latency of

114.4 msec. at the 30th minute for S-MPLS-TE tunnel. An average latency of 132.63 msec. is measured for S-MPLS-TE tunnel. On the other hand, ODL coupled S-MPLS-TE tunnel, in this scenario, has a maximum latency of 133.4 msec. at the 10th minute and a minimum of 94.6 msec. at the 55th minute. The average latency of ODL coupled S-MPLS-TE tunnel is 111.316 msec. This illustrates that there is 16.07% average latency reduction in ODL coupled S-MPLS-TE compared to that of S-MPLS-TE, in the second scenario.

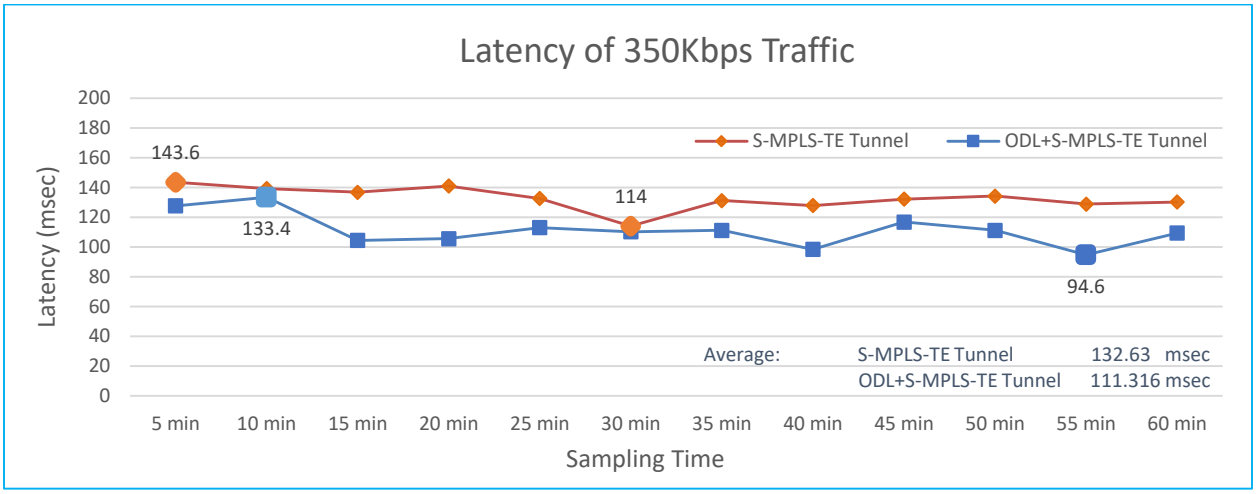


Figure 40: Latency of S-MPLS-TE and ODL + S-MPLS-TE Tunnels at 350Kbps

The reduction of average latency in ODL coupled S-MPLS-TE is increased from 15.05 msec. in the first scenario to 21.314 msec. in the second scenario. That is a 41.62% improvement in reduction of average latency. Figure 40 also shows that, unlike in the first scenario, ODL coupled S-MPLS-TE instantaneous tunnel latency never surpassed the latency of S-MPLS-TE tunnel in the second scenario. Regarding the average latency of S-MPLS-TE tunnel in both scenarios, it increased from 122.8 msec. in the first scenario to 132.63 msec. in the second scenario. This indicates an 8% surge in average latency of S-MPLS-TE tunnel from the first scenario to the second scenario. Similarly, the average latency of ODL coupled S-MPLS-TE tunnel increased from 107.75 msec. in the first scenario to 111.316 msec. in the second scenario. This also shows a 3.3% swell in the average latency of ODL coupled S-MPLS-TE from the first scenario to the second

scenario. The average latency percentage point increment from the first scenario to the second scenario is more than two-fold in S-MPLS-TE compared to ODL coupled S-MPLS-TE. Lower average latency increment is observed in ODL coupled S-MPLS-TE compared to S-MPLS-TE, as the amount of traffic is increased.

These latency reductions, in both scenarios, could arise from the fact that ODL coupled S-MPLS-TE tunnel employs strict EROs to build the tunnel path as compared to S-MPLS-TE tunnel, which uses a combination of strict and loose EROs. Loose next-hops require extra processing time by the ASBRs when crossing domains. This processing time adds to the overall latency of the S-MPLS-TE tunnel in both scenarios. The processing time by ASBRs increases as the amount of traffic passing through the tunnel also increases, which raises the latency of S-MPLS-TE tunnel at higher rate as compared to ODL coupled S-MPLS-TE tunnel, at higher traffic scenarios.

5.2. Packet Loss Analysis

Packet loss measurement results presented in tables 2 and 4, from both scenarios of S-MPLS-TE and ODL coupled S-MPLS-TE simulation, are analyzed in this section. Figure 41 shows packet loss of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels, in the first scenario (i.e. at 250Kbps of traffic), plotted against the sampling time. As shown in figure 41, S-MPLS-TE tunnel has a maximum of 6% packet loss measured at the 45th minute and a minimum of 0%. Average packet loss for S-MPLS-TE tunnel is 0.58%.

On the other hand, figure 41 also shows maximum packet loss of 2% at 25th, 50th and 60th minutes and a minimum of 0% for ODL coupled S-MPLS-TE tunnel. An average packet loss of 0.5% is recorded for ODL coupled S-MPLS-TE tunnel in 250Kbps traffic scenario. This shows that average packet loss of S-MPLS-TE tunnel is 0.08 percentage points higher than that of ODL coupled S-MPLS-TE tunnel. The average packet loss of ODL coupled S-MPLS-TE is 13.79% lower than that of S-MPLS-TE in the first scenario.

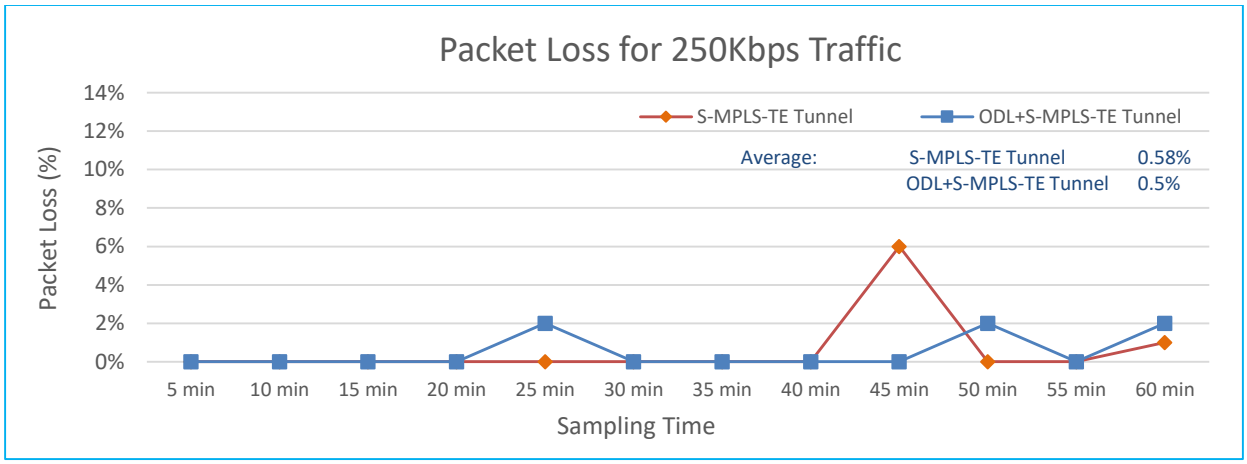


Figure 41: Packet Loss of S-MPLS-TE and ODL + S-MPLS-TE Tunnels at 250Kbps

Figure 42 shows packet loss of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels in the second scenario (i.e. at 350Kbps of traffic), plotted against the sampling time. Figure 42 shows a maximum packet loss of 4% at the 55th minute and a minimum of 0% for S-MPLS-TE tunnel. An average packet loss of 0.75% is measured for S-MPLS-TE tunnel. On the other hand, figure 42 also shows that ODL coupled S-MPLS-TE tunnel has a maximum packet loss of 4% at the 20th minute and a minimum of 0% in this scenario. The average packet loss of ODL coupled S-MPLS-TE tunnel, in 350Kbps traffic scenario, is 0.58%.

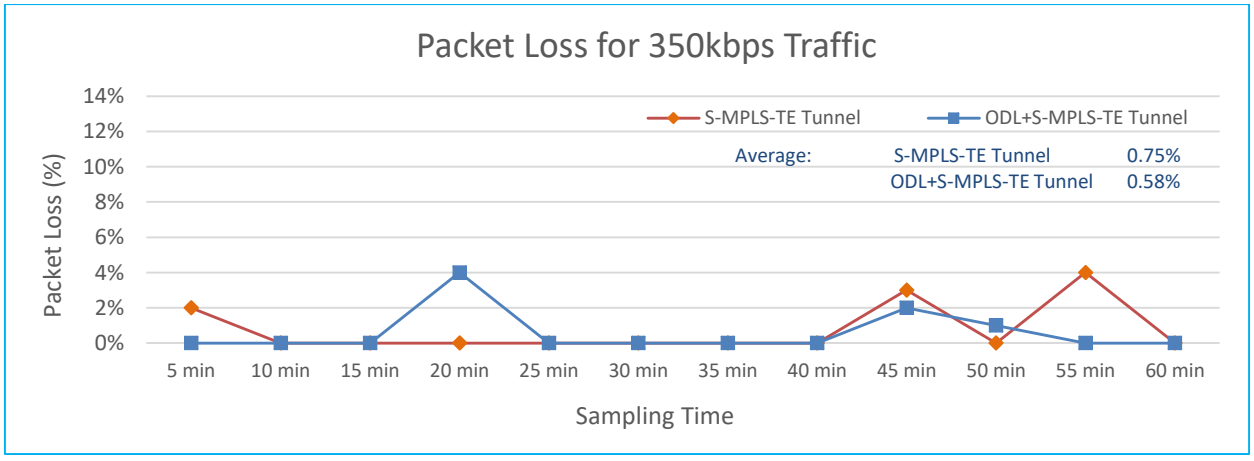


Figure 42: Packet Loss of S-MPLS-TE and ODL + S-MPLS-TE Tunnels at 350Kbps

This demonstrates that, average packet loss of S-MPLS-TE tunnel is 0.17 percentage points larger than that of ODL coupled S-MPLS-TE tunnel. The average packet loss of ODL coupled S-MPLS-TE is 22.667% lower than that of S-MPLS-TE.

Although instantaneous packet loss measurements were taken individually at different time periods in both scenarios, instantaneous packet loss of ODL coupled S-MPLS-TE tunnel exceeded that of S-MPLS-TE tunnel at 25th, 50th and 60th minutes in the first scenario and at 20th and 50th minutes in the second scenario.

Average packet loss in ODL coupled S-MPLS-TE is reduced by 13.79% in the first scenario and by 22.667% in the second scenario, compared to S-MPLS-TE. This average packet loss reduction shows a 64.37% improvement from the first scenario to the second scenario. Average packet loss of S-MPLS-TE tunnel increased from 0.58% in the first scenario to 0.75% in the second scenario. That is a 29.3% average packet loss growth in the second scenario. Similarly, ODL coupled S-MPLS-TE tunnel average packet loss raised from 0.5% in the first scenario to 0.58% in the second scenario. That is a 16% average packet loss growth in the second scenario which is much lower than that of S-MPLS-TE.

Furthermore, the use of combined strict and loose EROs by S-MPLS-TE tunnel as compared to ODL coupled S-MPLS-TE, which utilizes only strict EROs, could contribute to the improvement of packet loss in ODL coupled S-MPLS-TE tunnel. As extra processing is needed by loose next-hops at the ASBRs, while traversing TE LSP across domains, the extra processing could cause additional packet loss to the S-MPLS-TE tunnel in both scenarios. The additional processing by ASBRs increases as the amount of traffic passing through the tunnel also increases, which increases the packet loss of S-MPLS-TE tunnel at higher rate as compared to ODL coupled S-MPLS-TE tunnel, at higher traffic scenarios.

6. Conclusion

Coupling S-MPLS-TE with SDN enables the separation of control and data plane, and programmability of a multi-domain S-MPLS-TE network. TE tunnels can be created, updated and deleted using SDN controller with an end-to-end view of the S-MPLS-TE network. These tunnels can be programmed centrally from SDN controller. This resolves the difficulty in the manageability, maintainability, and reconfigure ability of the entire S-MPLS-TE network.

The requirement of an offline path calculation in an inter-AS TE tunnel setup can be resolved by using SDN's PCE capability. This is because PCE will have a complete end-to-end view of S-MPLS-TE topology. PCE can setup inter-AS TE tunnels in S-MPLS-TE transport network using strict explicit EROs, which is difficult to setup in S-MPLS-TE. PCE is also capable of re-optimizing S-MPLS-TE tunnels during link failure in multi-domain transport networks (S-MPLS-TE).

The performance of S-MPLS-TE can be enhanced by coupling SDN controller. In this this thesis, ODL controller with PCE feature coupled with S-MPLS-TE improved the performance of S-MPLS-TE. The latency of S-MPLS-TE is reduced by 12.26% and 16.07% at 250Kbps and 350Kbps of TCP traffic respectively. This indicates that the performance of S-MPLS-TE, in terms of latency, improves as the amount of traffic increases. The performance of S-MPLS-TE in terms of packet loss can also be improved by coupling it with ODL controller's PCE feature. Packet loss of S-MPLS-TE, in this thesis, is reduced by 13.79% and 22.667% at 250Kbps and 350Kbps of TCP traffic respectively. This shows that packet loss reduction improves as the traffic volume increases.

Ethio telecom can benefit from coupling SDN with its multi-domain network. This coupling can enhance the performance of its transport network.

7. Future Works

The benefits of SDN for S-MPLS-TE can be further explored by using DCP and HCP control plane models. Furthermore, the performance of SDN coupled S-MPLS-TE with segment routing can also be further studied.

References

[1] Huawei Technologies Co., Ltd, "Technical White Paper for Seamless MPLS Networking," Huawei Technologies Co., Ltd, Shenzhen, China, 2010.

[2] I. Šeremet and S. Čaušević, "Advancing Multiprotocol Label Switching Traffic Engineering with Segment Routing in Software Defined Network environment," in *19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, March 2020, pp. 1-6.

[3] S. Patil and M. S. Subhedar, "Analyzing MPLS Performance by SDN," in *Advances in Intelligent Systems and Computing: Emerging Technologies in Data Mining and Information Security, Vol. 3*, Warsaw, Poland, 2018, pp. 589-598.

[4] Y. Sinha, S. Bhatia, V. S. Shekhawat and G. S. S. Chalapathi, "MPLS based Hybridization in SDN," in *Fourth International Conference on Software Defined Systems (SDS)*, Valencia, Spain, 2017, pp. 156-161.

[5] M. M. Tajiki, B. Akbari, N. Mokari and L. Chiaraviglio, "SDN-Based Resource Allocation in MPLS Networks: A Hybrid Approach," *Concurrency and Computation Practice and Experience,* vol. 31, no. 8, pp. 1-11, August, 2018.

[6] J. E. Cole, W. E. Thain and T. Fallon, "Modeling Multi-Protocol Label Switching Networks in the Laboratory," in *122nd ASEE Annual Conference & Exposition*, Seattle.WA, June, 2015.

[7] J. BHALLA, "Multiprotocol Label Switching," *International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE),* vol. 1, no. 4, pp. 31-35, November 2015.

[8] J. Joshi, S. Gupta, P. Gupta, N. Singh and M. Kumari, "Multi Protocol Label Switching with Quality of Service in High Speed Computer Network," *International Journal of Engineering Science and Innovative Technology (IJESIT),* vol. 2, no. 2, pp. 83-87, 2013.

[9] D. Sharma and R. Singla, "A Detail Review on Multiprotocol Label Switching (MPLS)," *International Journal of Engineering Research and General Science,* vol. 3, no. 2, pp. 354-361, March-April, 2015.

[10] N. H. Technologies, "MPLS Technology White Paper," New H3C Technologies Co., Ltd., 2020.

[11] M. F. K.M, "A Survey on Multiprotocol Label Switching in Virtual Private Networks," in *Second International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India , 2018.

[12] E. Stewart and B. Miller, "Multi-Protocol Label Switching (MPLS) Conformance and Performance Testing [White Paper]," Ixia, Calabasas, CA, 2004.

[13] S. Gurung, "IMPLEMENTATION OF MPLS VPN," TURKU UNIVERSITY OF APPLIED SCIENCES, Turku, 2015.

[14] "MPLS Label Distribution Protocols Overview," Juniper Networks, 18 08 2014. [Online].Available: https://www.juniper.net/documentation/en_US/junose15.1/topics/concept/mpls-label-distribution-protocols.html. [Accessed 5 05 2021].

[15] V. Bassoo and M. I. Goulamghoss, "Analysis of traffic engineering and fast reroute on multiprotocol label switching," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2409 - 2420, 24 July 2020.

[16] K. Ramasamy and D. Medhi, Network Routing Algorithms, Protocols, and Architectures, Cambridge, MA 02139, United States: Morgan Kaufmann, 2018.

[17] K. Yugandhar, V. P. Gulati, S. A. Rabara and N. Rajendran, "MPLS ARCHITECTURE FOR SERVICE PROVIDER," in *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, Setúbal, Portugal, August2004.

[18] S. MULUGU and V. JOSEPH, NETWORK CONVERGENCE Ethernet Applications and Next Generation Packet Transport Architectures, Waltham, MA, USA: Elsevier Inc., 2014.

[19] Alcatel-Lucent, "EVOLVING TO "END-TO-END MPLS" ARCHITECTURES TECHNICAL White Paper," Alcatel-Lucent, July 2013.

[20] J. Lucek and I. Minei, MPLS-Enabled Applications Emerging Developments and New Technologies, West Sussex, England: John Wiley & Sons Ltd, 2005.

[21] CISCO, *MPLS Traffic Engineering Path Link and Node Protection Configuration Guide, Cisco IOS XE 3S*, CISCO, 2020.

[22] J. BELLESSA, "IMPLEMENTING MPLS WITH LABEL SWITCHING IN SOFTWARE-DEFINED NETWORKS," University of Illinois, Urbana, Illinois, 2015.

[23] S. M. J. Yasini, "IMPLEMENTATION OF SEGMENT ROUTING AND MPLS TRAFFIC ENGINEERING IN SOFTWARE-DEFINED NETWORK BASED ON GNS3 NETWORK EMULATOR AND OPENDAYLIGHT SDN CONTROLLER," POLITECNICO DI MILANO Dipartimento di Elettronica, Informazione e Bioingegneria, Milan, Italy, 2018.

[24] M. Sisov, "Building a Software-Defined Networking System with OpenDaylight Controller," Helsinki Metropolia University of Applied Sciences, March 2016.

[25] A. Durresi and M. Karakus, "An economic framework for analysis of network architectures: SDN and MPLS cases," *Journal of Network and Computer Applications,* vol. 136, p. 132–146, June 2019.

[26] G. Pujolle, Software Networks Virtualization, SDN, 5G and Security, London, UK : ISTE Ltd and John Wiley & Sons, Inc, 2020.

[27] S. Čaušević and I. Šeremet, "An analysis of reconvergence delay when using BGP-LS/PCEP as southbound protocols," in *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija Croatia, May 2019.

[28] J. Clarke, J. Lindblad and B. Claise, Network Programmability with YANG: The Structure of Network Automation with YANG, NETCONF, RESTCONF, and gNMI, Pearson Education, Inc., 2019.

[29] P. A. Sabaté, "Software Defined Networks (SDN) In Data Center Networks," Universitat Ramon Llull. La Salle, Barcelona, 2018.

[30] A. Hemid, "Facilitation of The OpenDaylight Architecture," in *CSCUBS 2017*, Bonn, Germany, 2017.

[31] F. Cugini, A. Giorgetti, N. Sambo, P. Castoldi and F. Paolucci, "A Survey on the Path Computation Element (PCE) Architecture," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 15, no. 4, pp. 1819-1841, 2013.

[32] J.-P. Vasseur, "The Path Computation Element (PCE) for MPLS networks and its application to the Internet of Things (IoT)," Télécom ParisTech, Paris, France, 2013.

[33] A. Durresi and M. Karakus, "Service Cost in Software Defined Networking (SDN)," in *31st International Conference on Advanced Information Networking and Applications*, Taipei, Taiwan, March 2017.

Appendix

SDN Coupled Seamless MPLS Traffic Engineering Performance Analysis

Medhin Worku Bihonegn
School of Electrical and Computer Engineering
Addis Ababa Institute of Technology
Addis Ababa University
Addis Ababa, Ethiopia
medhin.worku@gmail.com

Dr. Yalemzewd Negash
School of Electrical and Computer Engineering
Addis Ababa Institute of Technology
Addis Ababa University
Addis Ababa, Ethiopia
yalemzewdn@yahoo.com

***Abstract*— Internet Service Providers (ISPs), are requiring better performing MPLS networks. Seamless MPLS (S-MPLS) provides an end-to-end MPLS networks by integrating access, aggregation and core networks into a single MPLS domain, improving the performance of MPLS networks. The ISP network performance can also be further enhanced by implementing Traffic Engineering (TE). However, in S-MPLS Traffic Engineering (S-MPLS-TE), some of the tunnels fail to find a path and re-route traffic in the network when the best path goes down or gets congested. Software Defined Networking's (SDN's) Path Computation (PCE) feature can provide an approach to overcome these shortcomings of S-MPLS-TE. In this document, the performance of S-MPLS-TE coupled with SDN is analyzed and compared with S-MPLS-TE. In order to carry out the research, a simulation network, for S-MPLS-TE and SDN coupled S-MPLS-TE, is setup using GNS-3, OpenDaylight controller, Ostinato, Postman and PRTG. The topology of the simulation network setup is based on Ethio telecom's access, aggregation and core networks. For the setup, ISIS, MPLS, BGP-Labeled Unicast (BGP-LU), BGP-Link State (BGP-LS), PCEP, NETCONF and RESTCONF are utilized as networking protocols. In the simulation, latency and packet loss of both S-MPLS-TE and SDN coupled S-MPLS-TE simulations are measured, analyzed and compared for 250Kbps and 350Kbps TCP traffic scenarios. By coupling SDN with S-MPLS-TE, latency is enhanced by 12.26% in 250Kbps traffic scenario and by 16.07% in 350Kbps traffic scenario. Regarding packet loss, a 13.79% improvement in 250Kbps traffic scenario and a 22.667% improvement in 350Kbps traffic scenario are observed.**

Keywords—MPLS, Seamless MPLS, Software Defined Networking, Path Computation Element, Simulation, Performance, Analysis

I. INTRODUCTION

Highly scalable, resilient, flexible and manageable network architectures are being demanded by the increased adoption of Multi-Protocol Label Switching (MPLS) with in operator networks. Seamless MPLS (S-MPLS) provides a better substitute for implementing end-to-end MPLS networks by unifying access, aggregation and core networks into a single MPLS domain. With S-MPLS architecture, the entire network uses unified IP/MPLS networking technology, with an end-to-end control plane [1].

IP/MPLS network has control and data plane. The control plane is accountable for maintaining connections and exchanging protocol information with other routers or network devices. The data plane or the forwarding plane is in charge of forwarding packets from incoming interface to outgoing interface through the router. Software Defined Networking (SDN) is an architecture, which decouples control plane and data plane leaving only data plane on network devices. On the other hand control plane is retained on the SDN controller [2].

MPLS Traffic Engineering (MPLS-TE) faces the problem of creating backup path immediately when the best path goes down or gets congested. By using SDN, MPLS-TE performance is enhanced for the tunnel creation. It has been found that coupling SDN with MPLS offers better performance in terms of latency, response time and bandwidth utilization [3]. However, performance analysis of S-MPLS-TE coupled with SDN is left open for research. As the end-to-end control plane of access, aggregation and core networks are unified, coupling SDN to S-MPLS-TE will be feasible. This research will analyze the latency and packet loss of S-MPLS-TE coupled with SDN as compared to S-MPLS-TE.

II. SEAMLESS MPLS

Mobile operators like Ethio telecom are feeling the squeeze to enhance mobile backhaul speed, scale, and QoS, while controlling and diminishing the expense per bit in the transport network as mobile standards evolve. While Layer 3 (L3) packet transport has been principal in the core and generally in the aggregation domain of the network, the access network has largely been based on Layer 2 (L2) transport. Industry solutions regularly include a blend of L2 and L3 transport. Such models couple the transport and service layers and are tormented with both control and management plane translations at the edge of network. Interworking functions like VLAN in to Virtual Private LAN Service (VPLS), are regularly needed at the edge of these domains to provide the end-to-end services. These interworking elements are essentially touchpoints that lead to management intricacy and require repeated provisioning as new services are added to the network. They complicate the operation and management of the services, since they don't allow end-to-end service monitoring and require separate L2 and L3 Operation and Maintenance technologies. Seamless MPLS gives the design standard to making a scalable, resilient, and manageable IP backhaul framework that is optimized to seamlessly interwork with different telecom technologies. It is intended to simultaneously support numerous telecom services on a single unified network infrastructure. Seamless MPLS in backhaul networks unifies the MPLS transport across the access, aggregation, and core domains [4].

A. Seamless MPLS Design

Seamless MPLS employs hierarchical technique to overcome the scaling challenge in MPLS-based networks. Seamless MPLS utilizes a divide-and-conquer method where the core, aggregation, and access networks are partitioned in various MPLS/IP domains. A single - Autonomous System (AS) multi-area design or inter-AS organization could be used to segment the network between the core and aggregation domains. Seamless MPLS involves splitting the network's core, aggregation, and access layers into discrete Interior Gateway Protocol (IGP) / Label Distribution Protocol (LDP) domains, regardless of the kind of segmentation. Within each independent IGP domain, LDP is utilized for label distribution and the creation of Labeled Switched Paths (LSPs). This allows a device in an access, aggregation, or core domain to have reachability via intra domain LDP Label Switched Paths (LSPs) to any other device in the same domain. RFC 3107 procedures, which use BGP-Labeled Unicast (BGP-LU) as an inter-domain label distribution protocol to establish hierarchical LSPs across domains, are used to achieve reachability across domains. Interior BGP-LU (iBGP-LU) is used to construct intra-domain LSPs in single-AS multi-area designs. In inter-AS designs, iBGP-LU is used to create intra-domain LSPs inside the AS. To extend the end-to-end LSP across the AS boundary, exterior BGP-LU (eBGP-LU) is utilized. In both cases, seamless MPLS transport across domains is accomplished through hierarchical LSPs that rely on a BGP-distributed label for transiting isolated MPLS domains and an LDP distributed label for reaching the intra-domain Area Border Router (ABR) or Autonomous System Boundary Router (ASBR). The type of access and the size of the network are considered in structuring MPLS architectures [4].

B. Inter-AS MPLS Traffic Engineering

In MPLS-TE, information regarding all the links in the network is acquired by the distribution of link state by the IGP and is therefore confined within the same boundaries as the link state advertisements. Because the visibility of the topology and of the constraints is limited to a single IGP area or AS, TE LSPs dynamically computed by the head end are also limited in the same way. This presents a problem in big networks that deploy several IGP areas and ASs for scalability, or when services span across several service providers. Inter-AS LSPs exist both within the same provider and across different providers. Multiple ASs can be present within a single service provider's network. Separate ASs are maintained for a variety of reasons, including administrative authority and the desire to preserve routing protocol isolation between geographical domains and prevent the entire network from collapsing in the case of a local IGP failure. Setting up inter-AS tunnels is possible without any extensions, by computing the path offline and specifying the hops at the headend using ERO (IP explicit path). A path will be signaled across ASs once it is specified. Using hop-by-hop signaling between adjacent neighbors, an end-to-end LSP between ingress and egress LERs is created across AS boundaries. It is identical to the setup of a TE LSP within a single domain. A combination of strict and loose next-hop addresses are used to build the ERO, as fully strict explicit paths are not applicable in building inter-AS TE LSPs. Strict or loose next-hops can be used in intra-domain TE LSP building and only loose next-hops are used to traverse the TE LSP across domains through ASBR. ASBR forced link flooding can be used to advertise links without IGP adjacencies to all inter-AS nodes. TE LSPs can traverse these links at the edge of a network between two nodes running BGP even if the exit ASBR is not listed in the IP explicit path. Therefore, a headend LSR can consider the link when it computes its TE LSP path [5] [6].

C. Inter-AS MPLS Traffic Engineering Limitations

Setting up inter AS LSPs by computing the path offline and specifying hops at the headend has the drawback of forcing a service provider to adopt an operational model that relies on offline computation for both primary and secondary paths. Furthermore, unless the bypass tunnels protecting the inter AS links are also computed offline, the issue of fast re-route is also not addressed. In addition, strict EROs are also not applicable in inter-AS tunnel building [5] [6].

II. SOFTWARE DEFINED NETWORKING

In traditional networks, control-plane and the data-plane (or forwarding-plane), are tightly coupled. Furthermore, the entire network structure is very decentralized. The configuration of a single device can be completely independent of all others. These characteristics make managing, debugging, and reconfiguring huge networks extremely challenging. Software Defined Networking (SDN), aims to address these issues by radically rethinking network management. In a nutshell, SDN is an emerging networking architecture in which network control (control plane) is detached from the forwarding mechanism (data plane) and is directly programmable. The decoupling of data and control planes, as well as centralized control of the entire network, are the two defining characteristics of SDN. A third feature emerges from these two: programmability. New functionality can be added as needed by exposing a centralized view and control over an entire network infrastructure. The SDN controller, sometimes known as the network operating system (NOS), is the most important component of a software-defined network. To facilitate the central control logic, SDN discovers and updates a global abstract network view.

SDN is composed of three logical layers. The three layers are infrastructure layer, control layer and application layer. The infrastructure layer (also known as data plane) entails forwarding hardware, such as switches and routers, as well as forwarding components and Application Programming Interfaces (APIs). Network devices perform set of elementary forwarding operations. They are programmable devices that respond to the instructions sent by the SDN controller. The communication between SDN controller and network devices is enabled by southbound (SB) APIs. The control plane (also called control layer) contains all of the complexity that traditional network devices have as a result of numerous protocols, algorithms, and other factors. Control plane is composed of a set of software called SDN controllers that provide control capabilities for network forwarding behavior supervision. SDN controller is the brain of the network. The control layer manages forwarding hardware and installs forwarding rules via SB APIs. SDN controller creates a network abstraction view, comprising statistics and network state, and provides it to the application level. When an instruction is sent

from the application level, the controller receives it and passes it on to the lower level devices. The northbound APIs provide applications, on top of the SDN stack, with a network abstraction interface. This interface offers application-level network programmability. In application layer, applications and services take over control and infra-structure layer via northbound APIs such as Representational State Transfer (REST) APIs. The application layer contains a collection of applications that leverage the northbound API's functionalities to implement operational logic and network control [7][8][9] [10].

A. SDN Controller

The controller is the most important component in the SDN architecture. It implements the network's rules, executes the instructions provided by the applications, and distributes them among the network's various physical layer devices. The SDN controller platform often includes a number of "plug - and - play" modules that can be used to perform various network functions. Extensions can be added to improve functionality and support more advanced features like running algorithms for analytics and composing new rules across the network. NETCONF/YANG, RESTCONF, BGP-LS, PCEP, OpenFlow and OpenvSwitch Database (OVSDB) are some of the protocols used by SDN controllers to interact with switches/routers. The SDN controller contributes to the network by managing and monitoring its state and coordinating all of its installed services. TCP sessions are used to connect all devices to the controller. OpenDaylight, NOX, POX, Beacon, Floodlight, and ONOS are some of the SDN controller platforms that have been developed [11] [12].

B. Path Computaion Element

The separation between routing process and resource reservation operations, during MPLS network dynamic LSP provisioning path computation process, could lead to sub-optimal TE solutions. Additional issues, such as restricted topology visibility due to scalability and the requirement of several weeks for manual provision of sub-optimal TE solutions across multiple domains, arise when there is a move from single domain to multi domain networks. Inter-domain TE performance will be affected due to the complication of path computation by such limitations.

A set of techniques, defined under the parasol of the Path Computation Element (PCE) architecture, are proposed by the Internet Engineering Task Force (IETF). Path computation by dedicated network entities (PCEs) is the basis for these techniques. Fundamentally PCE is a path computation engine capable of computing a path for a TE LSP. Computed TE LSP will fulfill several constraints such as TE LSP bandwidth, destination, priority and other attributes. No assumption is made on the nature of the algorithm used for path computation. The PCE gathers link-state information and carry out path computation on behalf of network nodes. The Path Computation Element Protocol (PCEP) facilitates the communication between the network devices, referred to as Path Computation Client (PCC), and the PCE. The PCE, in its own domain, has full visibility and updated information on available network resources and is responsible for the path computation. In a multi domain scenarios, cooperation between PCEs takes place by sharing the result of each intra-domain path computation expressed as, for instance, border node(s) to traverse, encrypted intra-domain routes and metric values. The whole source-destination path will be provided by the combination of these results without additional information exchange among different domains. Specific techniques and procedures have been proposed and defined for each specific scenario. There are two modes of operation for PCE:

- Stateless: PCEs process path computation requests independently without considering previous path computation requests and established TE LSPs.
- State full: All previous path computation requests and established LSPs, stored in LSP database, will be considered during path computation. Because of this, State full PCE performs efficient path computation.

The PCE design relies on two functional elements: the PCE and the PCC. The PCE is in charge of carrying out constraint-based path computation for path requesting PCCs. It can be implemented on a dedicated server. On the other hand, PCC is implemented on a Network Management System (NMS) or on network device (e.g. LSP requesting source node). A PCE may also request path computation to other (peer or hierarchically higher) PCEs, behaving as a PCC. The two functional elements communicate PCEP [13] [14].

II. SIMULATION NETWORK SETUP

The simulation network setup, depicted in fig. 1, is based on Ethio telecom's core, aggregation and access networks. GNS-3 is used as emulation software to simulate S-MPLS-TE and SDN coupled S-MPLS-TE. Both S-MPLS-TE and SDN coupled S-MPLS-TE simulation network setup have three independent network domains each with its own AS. The first access and aggregation domain is based on Bole area aggregation and access networks with an AS umber of 65109. This domain is composed of one access Cell Service Gateway (BL-CSG) and two aggregation Radio Service Gateways (BL-RSG-1 and BL-RSG-2) routers with a single ostinato traffic generator. SDN capable Cisco IOS XRv routers are used for the BL-CSG emulation. The second core domain is composed of five provider Backbone Routers (BR-1, BR-2, BR-3, BR-4 and BR-5) and two provider edge routers, BL-NE40E-A and BD- ER-A with an AS number of 65108. The third access and

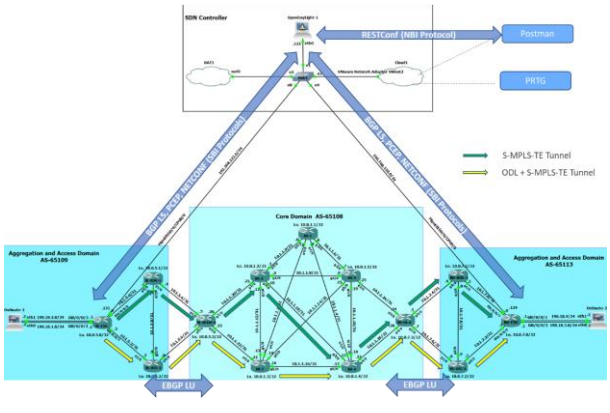


Fig. 1. Simulation network setup

aggregation domain is based on Bahir Dar access and aggregation networks with an AS number of 65113. This domain is composed of one access BD-CSG and two aggregation BD-RSG-1 and BD-RSG-2 routers with a one ostinato traffic generator. SDN capable Cisco IOS XRv routers are used for the BD-CSG emulation. For the SDN part of the simulation, OpenDaylight (ODL) controller is used. The ODL controller deployed version is Boron Stable Release 0.5.3 (Boron SR-3). To host ODL controller, GNS-3 Ubuntu Docker/container is utilized. The controller is connected to the S-MPLS-TE simulation setup through a hub. This hub also offers connectivity to the NAT cloud which provides internet connectivity for the Ubuntu Docker/container and to the Cloud-1 that provides host connectivity for the simulation network setup. Cloud-1 provides the simulation network setup access to Postman and PRTG. Postman is used to program the ODL controller. To measure the performance of both simulations, PRTG network monitoring tool is employed.

A. S-MPLS-TE Simulation

S-MPLS-TE is simulated on the three domains of the network setup. This is achieved by configuring loopback and link IP addresses, IGP (ISIS), MPLS/LDP, BGP-LU and TE on each router in the network setup. Since the network setup is segmented in to three independent AS domains, each domain will run its own IGP. For this simulation IS-IS is used as an IGP. IS-IS provides loopback to loopback reachability among the routers in the respective AS domain. MPLS LDP is configured on all intra-AS routers of all domains to build intra-AS LSPs. In order to traverse these intra-AS LSPs in to the neighboring AS domains, BGP-LU is used. BGP-LU unifies the access, aggregation and core domains as if they are a single MPLS domain (i.e. the MPLS is seamless (S-MPLS)). This helps to create end-to-end hierarchical LSPs. S-MPLS is realized by configuring BGP-LU on all intra-AS routers and ASBRs. To extend end-to-end LSP across AS boundaries, eBGP-LU is used at the ASBRs. Routers BL-RSG-1 and BL-RSG-2 form an eBGP-LU neighborship with BL-NE40E-A. In similar fashion, routers BD-RSG-1 and BD-RSG-2 form an eBGP-LU neighborship with BD-ER-A. The last step in S-MPLS-TE simulation network setup is TE configuration. MPLS TE is configured globally on each domain's S-MPLS router and specifically on interfaces that are members of the IGP (IS-IS). Since MPLS TE tunnels are restricted to single IGP domain, they cannot cross to other AS domains. So as to traverse TE tunnels over AS boundaries, ASBR forced flooding is configured on all ASBR interfaces without IGP adjacencies.

To simulate TE tunnel on S-MPLS-TE network setup, Explicit Route Object (ERO) is configured on Head End (HE) S-MPLS router. For this simulation, as shown in fig.2, TE tunnel te-21 is configured on the HE router, i.e. BL-CSG. The source of the S-MPLS-TE tunnel te-21 is BL-CSG and the destination of the tunnel is BD-CSG. The ERO for tunnel te-21 contains next-hop (NH) addresses spanning the three AS domains. The ERO is configured as a mix of strict and loose hops, as dynamic EROs are not possible in inter-AS TE tunnels. In addition, strict only EROs are also not possible in inter-AS TE tunnels. This is because path computation in inter-AS TE is limited to single IGP domain. Two traffic scenarios are deployed to measure the performance of S-MPLS-TE tunnel te-21. In the first scenario, 250kbps TCP traffic is generated by Ostinato-1 traffic generator through BL-CSG. In the second scenario, 350kbps of TCP traffic is generated by Ostinato-1 traffic generator through BL-CSG. In both scenarios, generated traffic is forwarded through BL-CSG tunnel te-21 destined to BD-CSG. The performance of the tunnel te-21 is measured in both scenarios independently for one hour using PRTG network monitoring platform. Latency and packet loss are used as performance parameters.

B. SDN Coupled S-MPLS-TE Simulation

To simulate SDN coupled S-MPLS-TE, ODL controller is installed on Ubuntu Docker/Container and coupled with the S-MPLS-TE network setup emulated in previous section A. ODL Boron release, version SR-3 Karaf distribution 0.5.3, is downloaded and installed on Ubuntu Docker on GNS3. In this simulation, ODL controller is used as PCE and the HE routers, i.e. BL-CSG and BD-CSG, are PCCs. To proceed with the simulation, BGP, PCEP, NETCONF and RESTCONF features are installed on ODL controller. The ODL controller gathers TE topology on the HE routers by establishing BGP-LS neighborship with the PCCs. To create, modify and remove TE tunnels on S-MPLS-TE network setup, PCEP peering between the ODL controller and PCCs is established. To establish BGP-LS neighborship and PCEP peering with the PCCs, the ODL controller is configured by updating 41-bgp-example configuration file in karaf directory. The modules in 41-bgp-example.xml configuration file are example-bgp-rib, which sets BGP Routing Information Base (RIB) of ODL controller, and example-bgp-peer which connects ODL controller to BGP-LS peer (PCCs). example-bgp-rib module is updated with RIB ID (IP address of ODL controller) and local AS (AS number of Bole and Bahir Dar Access and Aggregation domains). example-bgp-peer module is updated with IP addresses of BGP-LS speakers (PCCs) to which the ODL controller connect. To realize the coupling, the PCCs also need additional BGP-LS, PCE, NETCONF and SSH configurations. BGP-LS configuration on both S-MPLS-TE HE routers is achieved by adding ODL controller as BGP neighbor to the global BGP configuration with Link-State address family. PCEP peering between ODL controller (PCE) and S-MPLS-TE HE routers (PCCs) is attained by adding PCE configuration on the traffic engineering configuration of the PCCs.

After the coupling, the next step is to simulate ODL controller coupled S-MPLS-TE tunnel by programming the ODL controller. The ODL controller is programed using Postman. It uses XML and JSON formats to exchange data between Postman and itself. The ODL controller in turn configures the PCC according to the program, through NETCONF. To simulate TE tunnel on ODL coupled S-MPLS-TE network setup, REST POST request is sent using Postman in XML format. The REST POST request's body has the PCC (BL-CSG and BD-CSG) IP address, signaled name of the TE tunnel, source IP address of the TE tunnel, destination IP address of the TE tunnel and the ERO through which the tunnel traverses the three domains. In this simulation, as shown in fig. 2, TE tunnel te-100 is programmed on the ODL controller. The source of the ODL controller coupled S-MPLS-TE tunnel te-100 is BL-CSG and the destination of the tunnel is BD-CSG. The ERO for tunnel te-100 contains only strict next-hop (NH)

```
RP/0/0/CPU0:BL-CSG#show mpls traffic-eng tunnels brief
Sat May  1 13:17:24.786 UTC

                  TUNNEL NAME          DESTINATION      STATUS  STATE
                  tunnel-te21            10.0.7.0         up  up
                 >tunnel-te100           10.0.7.0         up  up
                  BDCSG-->BLCSG          10.0.5.0         up  up
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 1 (of 1) tails
Displayed 2 up, 0 down, 0 recovering, 0 recovered heads
RP/0/0/CPU0:BL-CSG#
```

Fig. 2. Status of S-MPLS-TE tunnel te-21 and SDN coupled S-MPLS-TE tunnel te-100 on BL-CSG

addresses spanning the three AS domains, unlike TE tunnel te-21 simulated in S-MPLS-TE simulation. To measure the performance of ODL controller coupled S-MPLS-TE tunnel te-100, two traffic scenarios are deployed. In the first scenario, 250kbps TCP traffic is generated by Ostinato-1 traffic generator through BL-CSG. In the second scenario, 350kbps of TCP traffic is generated by Ostinato-1 traffic generator through BL-CSG. In both scenarios, generated traffic is forwarded through BL-CSG ODL configured tunnel te-100 destined to BD-CSG. The performance (i.e. latency and packet loss) of tunnel te-100 is measured in both scenarios independently for a duration of one hour using PRTG network monitoring platform.

I. PERFORMANCE ANALYSIS

A. Latency Analysis

Fig. 3 shows latency of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels in the first scenario (i.e. at 250Kbps of traffic). As shown in fig. 3, latency of S-MPLS-TE tunnel has a maximum value of 134.8 msec. and a minimum value of 104.4 msec. The average latency of S-MPLS-TE tunnel is 122.8 msec. On the other hand, it also shows maximum latency of 116.8 msec. and a minimum of 94.6 msec. for ODL coupled S-MPLS-TE tunnel. An average latency of 107.75 msec. is recorded for ODL coupled S-MPLS-TE tunnel. This shows that, there is 12.26% latency reduction in average in ODL coupled S-MPLS-TE compared to that of S-MPLS-TE.

Fig. 4 shows latency of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels in the second scenario (i.e. at 350Kbps of traffic), plotted against the sampling time. It shows a maximum latency of 143.5 msec. and a minimum latency of 114.4 msec. for S-MPLS-TE tunnel. An average latency of 132.63 msec. is measured for S-MPLS-TE tunnel. Then again, ODL coupled S-MPLS-TE tunnel, in this scenario, has a maximum latency of 133.4 msec. and a minimum of 94.6 msec. The average latency of ODL coupled S-MPLS-TE tunnel is 111.316 msec. in this scenario. This illustrates that there is 16.07% average latency reduction in ODL coupled S-MPLS-TE compared to that of S-MPLS-TE, in the second scenario.

The reduction of average latency in ODL coupled S-MPLS-TE is increased from 15.05 msec. in the first scenario to 21.314 msec. in the second scenario. That is a 41.62% improvement in reduction of average latency. Regarding the average latency of S-MPLS-TE tunnel, it indicated an 8% surge in average latency from the first scenario to the second scenario. Similarly, the average latency of ODL coupled S-MPLS-TE tunnel showed a 3.3% swell from the first scenario to the second scenario. The average latency percentage point increment from the first scenario to the second scenario is more than two-fold in S-MPLS-TE compared to ODL coupled S-MPLS-TE. Lower average latency increment is observed in ODL coupled

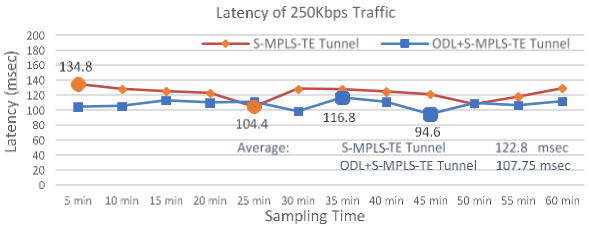


Fig. 3. Latency of 250Kbps traffic

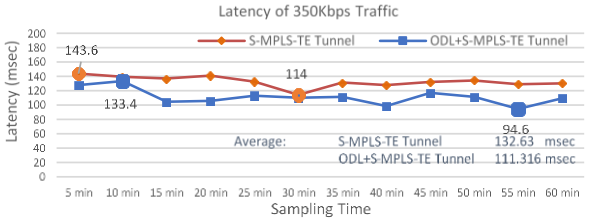


Fig. 4. Latency of 350Kbps traffic

S-MPLS-TE compared to S-MPLS-TE, as the amount of traffic is increased.

These latency reductions, in both scenarios, could arise from the fact that ODL coupled S-MPLS-TE tunnel employed strict EROs to build the tunnel path as compared to S-MPLS-TE tunnel, which uses a combination of strict and loose EROs. Loose next-hops require extra processing time by the ASBRs when crossing domains. This processing time adds to the overall latency of the S-MPLS-TE tunnel in both scenarios. The processing time by ASBRs increases as the amount of traffic passing through the tunnel also increases, which raises the latency of S-MPLS-TE tunnel at higher rate as compared to ODL coupled S-MPLS-TE tunnel, at higher traffic scenarios.

B. Packet Loss Analysis

Fig. 5 shows packet loss of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels, in the first scenario (i.e. at 250Kbps of traffic). As can be seen in fig. 5, S-MPLS-TE tunnel has a 6% maximum and 0% minimum packet loss. Average packet loss for S-MPLS-TE tunnel is 0.58%. On the other hand, it also shows maximum packet loss of 2% and a minimum of 0% for ODL coupled S-MPLS-TE tunnel. An average packet loss of 0.5% is recorded for ODL coupled S-MPLS-TE tunnel in 250Kbps traffic scenario. This shows that average packet loss of S-MPLS-TE tunnel is 0.08 percentage points higher than that of ODL coupled S-MPLS-TE tunnel. The average packet loss of ODL coupled S-MPLS-TE is 13.79% lower than that of S-MPLS-TE in the first scenario.

Fig. 6 shows packet loss of S-MPLS-TE and ODL coupled S-MPLS-TE tunnels in the second scenario (i.e. at 350Kbps of traffic). It shows a maximum packet loss of 4% and a minimum of 0% for S-MPLS-TE tunnel. An average packet loss of 0.75% is measured for S-MPLS-TE tunnel. Alternatively, it also shows that ODL coupled S-MPLS-TE tunnel has a maximum packet loss of 4% and a minimum of 0% in this scenario. The average packet loss of ODL coupled S-MPLS-TE tunnel, in 350 Kbps traffic scenario, is 0.58%. This demonstrates that, average

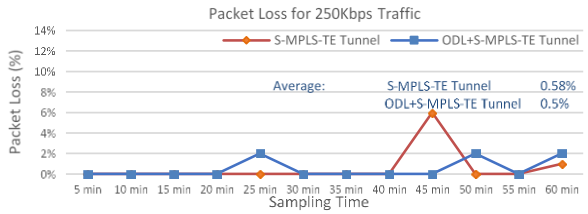


Fig. 1. Packet loss for 250Kbps traffic

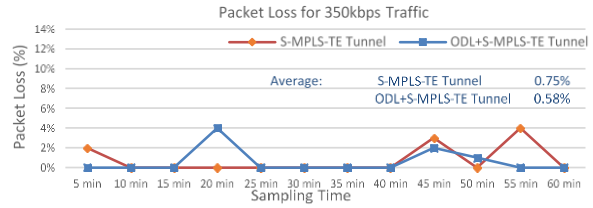


Fig. 2. Packet loss for 350Kbps traffic

packet loss of S-MPLS-TE tunnel is 0.17 percentage points larger than that of ODL coupled S-MPLS-TE tunnel. The average packet loss of ODL coupled S-MPLS-TE is 22.667% lower than that of S-MPLS-TE.

Average packet loss in ODL coupled S-MPLS-TE is reduced by 13.79% in the first scenario and by 22.667% in the second scenario, compared to S-MPLS-TE. This average packet loss reduction shows a 64.37% improvement from the first scenario to the second scenario. Average packet loss of S-MPLS-TE tunnel showed a 29.3% average packet loss surge in the second scenario. Similarly, ODL coupled S-MPLS-TE tunnel average packet loss exhibited a 16% average packet loss growth in the second scenario which is much lower than that of S-MPLS-TE.

Furthermore, the use of combined strict and loose EROs by S-MPLS-TE tunnel as compared to ODL coupled S-MPLS-TE, which utilizes only strict EROs, could contribute to the improvement of packet loss in ODL coupled S-MPLS-TE tunnel. As extra processing is needed by loose next-hops at the ASBRs, while traversing TE LSP across domains, the extra processing causes additional packet loss to the S-MPLS-TE tunnel in both scenarios. The additional processing by ASBRs increases as the amount of traffic passing through the tunnel also increases, which increases the packet loss of S-MPLS-TE tunnel at higher rate as compared to ODL coupled S-MPLS-TE tunnel, at higher traffic scenarios.

I. Conclusion

Coupling S-MPLS-TE with SDN enables the separation of control and data plane, and programmability of a multi-domain S-MPLS-TE network. TE tunnels can be created, updated and deleted using SDN controller with an end-to-end view of the S-MPLS-TE network. These tunnels can be programmed centrally from SDN controller. This resolves the difficulty in the manageability, maintainability, and reconfigure ability of the entire S-MPLS-TE network. The requirement of an offline path calculation in an inter-AS TE tunnel setup can be resolved by using SDN's PCE capability, since PCE will have a complete end-to-end view of S-MPLS-TE topology. PCE can setup inter-AS TE tunnels in S-MPLS-TE transport network using strict explicit EROs, which are difficult to setup in S-MPLS-TE. PCE is also capable of re-optimizing S-MPLS-TE tunnels during link failure in multi-domain transport networks (S-MPLS-TE). The performance of S-MPLS-TE can also be enhanced by coupling SDN controller. In this this research, ODL controller with PCE feature coupled with S-MPLS-TE improved the performance of S-MPLS-TE. The latency of S-MPLS-TE is reduced by 12.26% and 16.07% at 250Kbps and 350Kbps of TCP traffic respectively. This indicates that the performance of S-MPLS-TE, in terms of latency, improves as the amount of traffic increases. The performance of S-MPLS-TE in terms of packet loss can also be improved by coupling it with ODL controller's PCE feature. Packet loss of S-MPLS-TE is reduced by 13.79% and 22.667% at 250Kbps and 350Kbps of TCP traffic respectively. This shows that packet loss reduction improves as the traffic volume increases.

References

[1] Huawei Technologies Co., Ltd, "Technical White Paper for Seamless MPLS Networking," Huawei Technologies Co., Ltd, Shenzhen, China, 2010.

[2] I. Šeremet and S. Čaušević, "Advancing Multiprotocol Label Switching Traffic Engineering with Segment Routing in Software Defined Network environment," in 19th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2020, pp. 1-6.

[3] S. Patil and M. S. Subhedar, "Analyzing MPLS Performance by SDN," in Advances in Intelligent Systems and Computing: Emerging Technologies in Data Mining and Information Security, Vol. 3, Warsaw, Poland, 2018, pp. 589-598.

[4] S. MULUGU and V. JOSEPH, NETWORK CONVERGENCE Ethernet Applications and Next Generation Packet Transport Architectures, Waltham, MA, USA: Elsevier Inc., 2014.

[5] J. Lucek and I. Minei, MPLS-Enabled Applications Emerging Developments and New Technologies, West Sussex, England: John Wiley & Sons Ltd, 2005.

[6] CISCO, MPLS Traffic Engineering Path Link and Node Protection Configuration Guide, Cisco IOS XE 3S, CISCO, 2020.

[7] J. BELLESSA, "IMPLEMENTING MPLS WITH LABEL SWITCHING IN SOFTWARE-DEFINED NETWORKS," University of Illinois, Urbana, Illinois, 2015.

[8] S. M. J. Yasini, "IMPLEMENTATION OF SEGMENT ROUTING AND MPLS TRAFFIC ENGINEERING IN SOFTWARE-DEFINED NETWORK BASED ON GNS3 NETWORK EMULATOR AND OPENDAYLIGHT SDN CONTROLLER," POLITECNICO DI MILANO Dipartimento di Elettronica, Informazione e Bioingegneria, Milan, Italy, 2018.

[9] M. Sisov, "Building a Software-Defined Networking System with OpenDaylight Controller," Helsinki Metropolia University of Applied Sciences, March 2016.

[10] A. Durresi and M. Karakus, "An economic framework for analysis of network architectures: SDN and MPLS cases," Journal of Network and Computer Applications, vol. 136, p. 132–146, June 2019.

[11] P. A. Sabaté, "Software Defined Networks (SDN) In Data Center Networks," Universitat Ramon Llull. La Salle, Barcelona, 2018.

[12] A. Hemid, "Facilitation of The OpenDaylight Architecture," in CSCUBS 2017, Bonn, Germany, 2017.

[13] F. Cugini, A. Giorgetti, N. Sambo, P. Castoldi and F. Paolucci, "A Survey on the Path Computation Element (PCE) Architecture," IEEE COMMUNICATIONS SURVEYS & TUTORIALS, vol. 15, no. 4, pp. 1819-1841, 2013.

[14] J.-P. Vasseur, "The Path Computation Element (PCE) for MPLS networks and its application to the Internet of Things (IoT)," Télécom ParisTech, Paris, France, 2013.