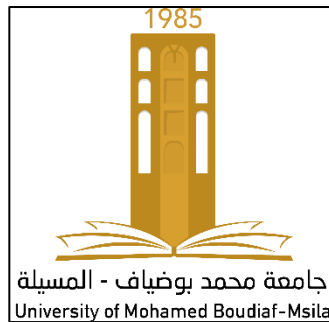




**FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE**

In partial fulfillment of the requirements for the degree of

Master in Computer science

By

Lemkhalti Aouni

Yasser Salamani

Title of the thesis

**Automating the Migration of an ISP Network
Infrastructure from MPLS to SRv6**

Under the supervision of

Lamri Sayad

Composition of the jury

DR A. Khettaf	University of Msila	President
DR LAMRI Sayad	University of Msila	Reporter
DR R. Yagoubi	University of Msila	Examiner

Juin, 2026

Abstract:

Internet Service Providers (ISPs) increasingly face the operational and scalability limitations of legacy Multiprotocol Label Switching (MPLS) networks in meeting the

demands of modern services such as cloud computing, 5G, and IoT. Segment Routing over IPv6 (SRv6) has emerged as a promising next-generation technology that leverages the IPv6 architecture to deliver a flexible, programmable, and simplified data plane, eliminating the need for legacy signaling protocols such as LDP and RSVP-TE. However, migrating from MPLS to SRv6 in a live ISP environment is a complex undertaking that carries significant risk of service disruption when performed manually. This dissertation studies the key concepts behind ISP network infrastructure, MPLS technology, Segment Routing, and SRv6, analyzes the technical and operational challenges of migration, and examines the coexistence strategies and migration paths available to operators. The work proposes an automated approach to facilitate the transition from MPLS to SRv6, aiming to reduce human intervention, minimize configuration errors, and support the modernization of ISP infrastructure.

Key words: MPLS; SRv6; Segment Routing; ISP Network; Network Automation; Migration

الملخص:

تواجه شبكات مزودي خدمة الإنترنت (ISP) تحديات متزايدة نتيجة القيود التشغيلية وحدود قابلية التوسع التي تفرضها شبكات MPLS التقليدية في مواجهة متطلبات الخدمات الحديثة كالحوسبة السحابية والجيل الخامس وإنترنت الأشياء. تبرز تقنية التوجيه القطعي عبر IPv6 بوصفها تقنية الجيل القادم التي تعتمد بنية IPv6 لتوفير مستوى بيانات مرّن وقابل للبرمجة ومبسّط، مما يُغني عن بروتوكولات الإشارة القديمة كـ LDP وRSVP-TE. غير أن تهجير شبكة ISP الحيّة من MPLS إلى SRv6 يُعدّ عملية بالغة التعقيد، وتنطوي على مخاطر انقطاع الخدمة عند إجرائها يدويًا. تدرس هذه المذكرة المفاهيم الأساسية لشبكات ISP وتقنية MPLS والتوجيه القطعي وSRv6، وتحلل التحديات التقنية والتشغيلية للتهجير، وتستعرض استراتيجيات التعايش ومسارات الانتقال المتاحة، وتقترح مقاربة آلية تُيسّر هذا الانتقال بهدف تقليص التدخل البشري والأخطاء وتحديث البنية التحتية لمزودي الخدمة.

الكلمات المفتاحية: SRv6؛ MPLS؛ التوجيه القطعي؛ شبكة ISP؛ أتمتة الشبكات؛ التهجي

Dedications

"I dedicate this work to the dearest people in my life, my parents, who have been supporting me at every step, and whose patience and diligence I carried with me until I had this achievement,

and to all who supported me and encouraged me with kind words or open prayers on my academic journey."

Lemkhalti Aouni

"I dedicate this work to my parents, whose endless love, support, and sacrifices have been the greatest source of strength throughout my academic journey. Their patience, encouragement, and faith in me have been the foundation of this achievement.

I also dedicate this work to everyone who stood beside me, supported me, and lifted me with their kind words, sincere prayers, and constant encouragement."

Salamani Yasser

Acknowledgments

We are deeply indebted to our thesis supervisor [Supervisor's name] whose unlimited steadfast support and inspirations have made this project a great success. In a very special way, we thank him for every support he has rendered unto us to see that we succeed in this challenging study.

Special thanks go to our friends and families who have contained the hectic moments and stress we have been through during the course of the research project.

We thank the school for giving us the grand opportunity to work as a team which has indeed promoted our team work spirit and communication skills. We also thank the individual group members for the good team spirit and solidarity.

Contents

List of Tables

List of Figures

<u>List Of Abbreviations</u>	
<u>GENERAL INTRODUCTION</u>	11
<u>Chapter 1 General Background on ISP Networks, MPLS, and Segment Routing</u>	12
<u>1.1. ISP Network Infrastructures</u>	12
1.1.1. Definition and Role of an Internet Service Provider (ISP)	12
1.1.2. ISP Network Architecture	13
<u>1.1.3. Service Requirements</u>	13
<u>1.2. Fundamentals of IP Routing in Service Provider Networks</u>	13
<u>1.2.1. Principles of Packet Forwarding in IP Networks</u>	14
<u>1.2.2. Interior and Exterior Routing Protocols</u>	14
<u>1.2.3. IP Routing Limits</u>	14
<u>1.3. Overview of MPLS Technology</u>	15
<u>1.3.1. Evolution of WAN technologies towards MPLS</u>	15
<u>1.3.2. Definition and operating principles of MPLS</u>	15
<u>1.3.3. MPLS Architecture and Main Components</u>	16
<u>1.3.4. MPLS Labels</u>	18
<u>1.3.5. Benefits of MPLS</u>	21
<u>1.4. MPLS Services and Traffic Engineering</u>	21
<u>1.4.1. VPN services over MPLS</u>	21
<u>1.4.2. Quality of Service and traffic engineering in MPLS</u>	21
<u>1.4.3. Operational Challenges and Scalability Limits of MPLS</u>	22
<u>1.5. Segment Routing</u>	22
<u>1.5.1. Principles of Segment Routing</u>	22
<u>1.5.2. Segment Routing Over MPLS (SR-MPLS)</u>	23
<u>1.5.3. Segment Identifiers and Path Computation</u>	23
<u>1.5.4. IPv6 Segment</u>	24
<u>Chapter 2 Fundamentals of SRv6 and Migration</u>	25
<u>2.1. SRv6</u>	25
<u>2.1.1. Definition SRv6</u>	25
<u>2.1.2. SRv6 Architecture</u>	26
<u>2.1.3. SRv6 Header</u>	27
<u>2.1.4. SRv6 Segment Format</u>	29
<u>2.1.5. BGP Protocol Extensions</u>	30
<u>2.1.6. Packet Routing in SRv6</u>	30
<u>2.2. Strategies for Migration to SRv6</u>	32
<u>2.2.1. Motivation for Migrating Toward SRv6</u>	32
<u>2.2.2. Indirect Migration Strategies</u>	32
<u>2.2.3. Technical and operational challenges of migration</u>	33

<u>2.2.4. Direct Evolution Strategies</u>	34
<u>Chapter 3 Implementation, Coexistence, and Verification of MPLS-to-SRv6 Migration</u>	
<u>3.1. Introduction</u>	37
<u>3.1.1. Objectives of the Implementation Phase</u>	37
<u>3.1.2. Adopted Migration Methodology</u>	38
<u>3.2. Work Environment and Experimentation</u>	38
<u>3.2.1. Physical Environment</u>	38
<u>3.2.2. Software Environment</u>	39
<u>3.2.3. Network Topology</u>	40
<u>3.3. Building the Basic MPLS L3VPN Network</u>	43
<u>3.3.1. Device Roles in the Network</u>	43
<u>3.3.2. IPv4 Addressing Plan</u>	44
<u>3.3.3. IS-IS Configuration</u>	48
<u>3.3.4. MPLS and LDP Configuration</u>	50
<u>3.3.5. Creating VPN Instances</u>	52
<u>3.3.6. Configuring MP-BGP VPNv4</u>	54
<u>3.3.7. Configuring PE-CE Connectivity</u>	56
<u>3.3.8. Analysis of the MPLS L3VPN Forwarding Mechanism</u>	58
<u>3.3.9. Testing and Verifying the MPLS Network</u>	59
<u>3.3.10. MPLS Phase Summary</u>	68
<u>3.4. Integration of IPv6 and SRv6</u>	68
<u>3.4.1. IPv6 addressing scheme</u>	69
<u>3.4.2. Enabling IS-IS IPv6 dual-stack</u>	71
<u>3.5. SRv6 Configuration and Locator Activation</u>	76
<u>3.5.1. SRv6 SID Structure</u>	77
<u>3.5.2. Defining the SRv6 Locator on PE-1 and PE-2</u>	77
<u>3.6. Migration of VPN_B to SRv6 – First Partial Phase</u>	78
<u>3.6.1. End.DT4 SID Configuration for VPN_B</u>	79
<u>3.6.2. IPv6 Configuration on CE-B1 and CE-B2</u>	80
<u>3.6.3. Configuring the VPN_B IPv6 Family and BGP VPNv6 on PE-1 and PE-2</u>	82
<u>3.7. Coexistence Phase Between MPLS and SRv6</u>	85
<u>3.7.1. Coexistence Architecture</u>	86
<u>3.7.2. Coexistence Verification</u>	86
<u>3.8. Full Migration —Moving VPN_A to SRv6</u>	89
<u>3.8.1. Configuring the End.DT4 SID for VPN_A</u>	89
<u>3.8.2. Configuring IPv6 on CE-A1 and CE-A2</u>	90
<u>3.8.3. Configuring VPN_A IPv6 family and BGP VPNv6 on PE-1 and PE-2</u>	91
<u>3.9. Final Verification of the Full Migration</u>	93

<u>3.9.1. Control-Plane Verification</u>	93
<u>3.9.2. Data-Plane Verification</u>	97
<u>3.9.3. Traceroute analysis — conclusive proof of pure SRv6</u>	99
<u>3.9.4. Summary of the Full-Migration Results</u>	100
<u>Chapter 4 Comparative Analysis of MPLS and SRv6 and Automation Framework</u>	
<u>4.1. Comparison of the network architecture</u>	102
<u>4.2. Control-plane comparison</u>	104
<u>4.3. Data-plane Comparison</u>	107
<u>4.4. Performance Comparison</u>	110
<u>4.5. Operational Complexity Comparison</u>	112
<u>4.6. Automation Framework</u>	113
<u>4.6.1. Objective of the Automation</u>	113
<u>4.6.2. Automation Architecture</u>	114
<u>4.6.3. Environment Preparation</u>	115
<u>4.6.4. SSH Connection Script</u>	119
<u>4.6.5. Automatic Information Collection</u>	120
<u>4.6.6. Challenges and Limitations</u>	125
<u>4.6.7. Migration Challenges</u>	126
<u>4.6.8. Implementation Challenges and Difficulties</u>	128
<u>4.6.9. Lessons Learned</u>	129
<u>GENERAL CONCLUSION</u>	131
<u>1. Conclusion</u>	131
<u>2. Future work</u>	132
<u>References</u>	133

List of Tables

<u>Tab1.1- Interior vs. exterior routing protocols</u>	14
<u>Tab3.1- Technical Specifications of the Experimental Environment</u>	38
<u>Tab3.2- Software Components and Versions Used in the Project</u>	39
<u>Tab3.3- Table of Device Roles and Loopback Addresses</u>	41
<u>Tab3.4-Table of Links and Addresses Between Network Devices</u>	42

<u>Tab3.5- Table of VPN Service Properties in the Migration Scenario</u>	43
<u>Tab3.6- Table of Device Descriptions and Roles in the Topology</u>	44
<u>Tab3.7- Table of IPv4 Addressing Plan</u>	44
<u>Tab3.8- Table of IS-IS Configuration Parameters for Core Devices</u>	48
<u>Tab3.9- Table of MPLS LDP Settings on Core Devices</u>	51
<u>Tab3.10- Table of VPN A and VPN B Service Properties</u>	53
<u>Tab3.11- Table of iBGP Session Configuration Between PE-1 and PE-2</u>	55
<u>Tab3.12- Table of Customer Mapping to PE Routers and Addresses</u>	58
<u>Tab3.13- Packet Forwarding from CE-A1 to CE-A2 Step by Step</u>	59
<u>Tab3.14- VPN Connectivity Test Results</u>	65
<u>Tab3.15- SRv6 Locator and IPv6 Addressing Table</u>	69
<u>Tab3.16- SRv6 Locator and SID Mapping Table</u>	77
<u>Tab3.17- Coexistence Architecture for VPN A and VPN B</u>	86
<u>Tab3.18- SRv6 Data-Plane Verification — Ping Tests</u>	97
<u>Tab3.19- Full Migration Summary</u>	100
<u>Tab4.1-Comparison of MPLS L3VPN and SRv6 End.DT4 in the Migration</u>	102
<u>Tab4.2-Control-Plane Comparison Between MPLS L3VPN and SRv6</u>	105
<u>Tab4.3-Data Plane Comparison Between MPLS L3VPN and SRv6 End.DT4</u>	107
<u>Tab4.4-MPLS L3VPN Forwarding Process</u>	108
<u>Tab4.5-SRv6 Packet Forwarding from CE-A1 to CE-A2</u>	109
<u>Tab4.6-Performance Summary Before and After Migration</u>	110
<u>Tab4.7-Operational Complexity Comparison Between MPLS L3VPN and SRv6 End.DT4</u>	112
<u>Tab4.8-Operational Benefits of Configuration Automation in SRv6 Deployment</u>	114
<u>Tab4.9-Software and Hardware Environment for the Automation Framework</u>	114
<u>Tab4.10-Device Connectivity Information Table for the Automation Environment</u>	118
<u>Tab4.11-Huawei VRP R011 Limitations in SRv6 Support</u>	125

List of Figures

<u>Figure1.1 –The layered MPLS architecture</u>	16
<u>Figure1.2 –Fig1.2 –MPLS Network Architecture</u>	17
<u>Figure1.3 –Structure of an MPLS header</u>	18
<u>Figure1.4 –MPLS Label Operations: Push, Swap, and Pop</u>	20
<u>Figure2.1 –L3VPN over SRv6 Illustration</u>	26

<u>Figure2.2 –Segment Routing Header (SRH) Format</u>	28
<u>Figure2.3 –SRv6 Segment Format</u>	29
<u>Figure2.4 –SRv6 Packet Transfer Process</u>	30
<u>Figure3.1 –Overall Topology of the MPLS to SRv6 Migration Lab</u>	41
<u>Figure3.2 –ISIS Neighbor Display on PE-1</u>	60
<u>Figure3.3 –ISIS Neighbor Display on P</u>	60
<u>Figure3.4 –ISIS Neighbor Display on PE-2</u>	60
<u>Figure3.5 –LDP Session Display on PE-1</u>	61
<u>Figure3.6 –LDP Session Display on P</u>	61
<u>Figure3.7 –LDP Session Display on PE-2</u>	62
<u>Figure3.8 –BGP VPNv4 Session Display on PE-1</u>	62
<u>Figure3.9 –BGP VPNv4 Session Display on PE-1</u>	63
<u>Figure3.10 –VPN-Instance A Route Table on PE-1</u>	63
<u>Figure3.11 –VPN-Instance A Routing Table on PE-1</u>	64
<u>Figure3.12 –VPN-Instance A Routing Table on PE-2</u>	64
<u>Figure3.13 –VPN-Instance A Routing Table on PE-2</u>	65
<u>Figure3.14 –Ping Test for VPN_A on PE-1</u>	66
<u>Figure3.15 –Ping Test for VPN_B on PE-1</u>	67
<u>Figure3.16 –Ping Test from CE-A1 to 4.4.4.4</u>	67
<u>Figure3.17 –Ping Test from CE-B1 to 5.5.5.5</u>	68
<u>Figure3.18 –Network topology with IPv6 addresses</u>	71
<u>Figure3.19 –IPv6 Ping Test on PE-1</u>	76
<u>Figure3.20 –SRv6 Locator Configuration Table</u>	78
<u>Figure3.21 –End.DT4 Forwarding Entry for VPN_B</u>	80
<u>Figure3.22 –Verification of BGP VPNv6 on PE-1</u>	84
<u>Figure3.23 –VPN_A Ping Results on PE-1</u>	87
<u>Figure3.24 –IPv6 Ping Test for VPN_B on PE-1</u>	88
<u>Figure3.25 –BGP VPNv4 Peer Status for VPN_A and VPN_B</u>	88
<u>Figure3.26 –BGP VPNv6 Peer Status for VPN_B</u>	89
<u>Figure3.27 –BGP VPNv6 Peer Status for VPN_A and VPN_B</u>	93
<u>Figure3.28 –Local SID End.DT4 Forwarding Table on PE-1</u>	97
<u>Figure3.29 –Ping Test from CE-A1 to CE-A2</u>	98
<u>Figure3.30 –Ping Test from CE-B1 to CE-B2</u>	98
<u>Figure3.31 –Actual Traceroute Result for VPN_B</u>	99
<u>Figure3.32 –Actual Traceroute Result for VPN_A</u>	100
<u>Figure4.1 –MPLS L3VPN Service Architecture Before Migration</u>	103
<u>Figure4.2 –SRv6 End.DT4 Architecture After Full Migration</u>	104

List Of Abbreviations

3G:Third generation

4G:Fourth generation

5G:Fifth generation

B2B:Business to Business

BGP:Border Gateway Protocol

EGP:Exterior Gateway Protocol

EGRP:Enhanced Interior Gateway Routing Protocol

ISP:Internet Service Provider

ICMP:Internet Control Message Protocol

IEFT:Internet Engineering Task Force

IPv4:Internet Protocol version 4

IPv6:Internet Protocol version 6

IS-IS:Intermediate System to Intermediate System

LAN:Local Area Network

MAN:Metropolitan Area Network

LDP:Label Distribution Protocol

LSR:Label Switch Router

MAC:Media Access Control

MPLS:Multiprotocol Label Switching

MPLS-TE:MPLS Traffic Engineering

NAT:Network Address Translation

OSI:Open Systems Interconnection

OSPF-TE:Open Shortest Path First with Traffic Engineering

OSPF:Open Shortest Path First

BANG :Personal Area Network

PE:Provider Edge

PPP:Point-to-Point Protocol

PPTP:Point-to-Point Tunneling Protocol

RFC:Request for Comments

RIB:Routing Information Base

RID:Router ID

RIP:Routing Information Protocol

UDP:User Datagram Protocol

VLAN:Virtual Local Area Network

VLSM:Variable Length Subnet Mask

VPLS:Virtual Private LAN Service

VPN:Virtual Private Network

SRv6:Segment Routing over IPv6

GENERAL INTRODUCTION

Over the past few years, the use of Internet services, cloud applications, video streaming and connected devices has increased which makes communication networks more important than ever. ISPs operate large and complex network manufacturers, all while providing good performance, scalability, and continuity of service. For that reason, many carriers seek modern technologies that can enhance network efficiency and simplified management, **Multiprotocol Label Switching (MPLS)** has been in use for many years on ISP networks as it allows reliable traffic forwarding; supports virtual private networks; and assists operators in managing traffic more granularly. But over time, MPLS has shown limitations with evolving network services and growing needs for flexibility. The shortcomings of these limitations drive researchers and network operators to seek more advanced solutions, **One of the interesting technologies targeted** for future service provider networks is Segment Routing over IPv6 (SRv6). It leverages the IPv6 architecture to create a more flexible and programmable packet-forwarding scheme. SRv6 can simplify network operations and reduce protocol dependencies while providing advance traffic engineering functions. Due to these benefits, many ISPs are looking to migrate from traditional MPLS based infrastructures to SRv6 based environment, Although this migration has its advantages, transitioning from **MPLS to SRv6** will not be a simple task. It involves detailed planning and reconfiguration of the network whilst keeping existing services running in parallel, which would not be possible without supporting both old and new architecture(s). Manually performing this migration would be time consuming and could also increase the potential for configuration errors, which could lead to service disruption. So, automation becomes a key solution enabling the migration process to be much quicker in addition to being safer and smoother.

This dissertation focuses on the automation of the migration of an ISP network infrastructure from MPLS to SRv6. The main objective is to study the existing technologies, understand the challenges of migration, and propose an automated approach that can facilitate this transition. The work also aims to highlight the role of network automation in reducing human intervention, improving accuracy, and supporting the modernization of ISP .

Chapter 1:General Background on ISP Networks, MPLS, and Segment Routing

Introduction

We start this chapter with an overview of the ISP networks and main routing concepts related to large-scale service provider scenarios. It then presents NAT, its architecture & key advantages over other types of backbone networks, before describing some of the limitations in terms of both operational and scalability capability. Following this, the chapter introduces the concept of Segment Routing and argues that SRv6 is a natural evolution built on top of v6 with great potential for flexibility. Lastly, this section covers how and why you would migrate from MPLS to SRv6, and provides some high level information about the role of network automation in automation that transition. The theoretical framework that this chapter provides is thus indispensable for the later chapters of this work.

1.1. ISP Network Infrastructures

Internet Service Provider (ISP) networks are one of the node fringes and are critical layers of the telecommunications infrastructure that connect users to Internet services, thus facilitating data exchanges between people and companies and also between platforms. These networks consist of multiple interconnected components, including access networks, aggregation networks, and the core network itself, with each component playing a crucial role in seamlessly routing data. The major growth in the use of Internet, along with the emergence of new services that require a high degree of speed and stability, has raised many challenges within ISP networks concerning performance, capacity and flexibility management. Because of this fact it is very important to study the general structure of these networks and understand how they work, which helps understanding of applicable technologies and preparation for transition towards new and more effective solutions such as MPLS and SRv6.

1.1.1. Definition and Role of an Internet Service Provider (ISP)

An Internet Service Provider (ISP) is an organization that provides consumers with access to the World Wide Web; through fiber, DSL, cable, satellite or wireless.. ISPs facilitate

communication between individual users and organizations with the global infrastructure required for data routing as well as value added services like email and security.

1.1.2. ISP Network Architecture

An Internet service provider network consists of several key components that operate with each other. There is an artifact that connects the organic to the big Internet and there exist some devices executing the network sending data crosswise to another place. In addition, there is a section that serves customers, allowing the user to access the Internet through fiber optics, ADSL or wireless network. We could also discover modem, servers and monitoring systems, and all of these help the network work good and stay the service available .[\[7\]](#)

1.1.3. Service Requirements

Today in the **ISP provider landscape**, just providing basic Internet connection is not enough, the provider must be fast, stable and mostly always up. The ISP should also be able to quickly turn on new services and respond systematically to customer demands such as subscription, upgrades, or additional bandwidth. The other crucial point is that continuous monitoring, the ability to address problems in a short period of time, providing good quality service and innovative automated systems for smoother management and reduction of errors [\[8\]](#)

1.2. Fundamentals of IP Routing in Service Provider Networks

IP routing is the technique of transferring data packets from one network device to another. Each router examines the destination IP address of the packet and consults its own routing table to determine where to forward it next. For example in the entire Service Provider domain, routers implement two classes of routing protocols: dedicated IGP protocols which are IS-IS or OSPF to learn route/communicate within their own autonomous system (AS) and BGP for learning routes from other AS and transport customer services over the backbone. These sets of protocols can function together to ensure that each packet gets to the right place in the most efficient and reliable means possible.[\[9\]](#)

1.2.1. Principles of Packet Forwarding in IP Networks

Packet forwarding in IP networks is the process of moving data packets from a source device to a destination device through one or more routers. When a packet arrives at a router, the router reads the destination IP address found in the packet header, then it checks its routing table to know the best next path. After that, the router sends the packet to the next hop until it reaches the final destination. This operation is repeated at every router on the path. In general, packet forwarding depends on logical addressing, routing tables, and the ability of network devices to make fast forwarding decisions.[\[4\]](#)

1.2.2. Interior and Exterior Routing Protocols

Routing protocols: Routers exchange path information and forward packets to destination using two categories of routing protocols known as Internal, and External routing protocol. Interior protocols are used within the same autonomous system AS, while exterior ones, between different autonomous systems.[\[3\]](#)

Type	Use	Examples
Interior routing protocols (IGP)	Used within the same autonomous system	OSPF, IS-IS, RIP, EIGRP
Exterior routing protocols (EGP)	Used between different autonomous systems.	BGP

Tab1.1- Interior vs. exterior routing protocols

1.2.3. IP Routing Limits

Traditional IP routing has certain limitations when it comes to large scale networks. In a growing network, routing tables become large and more substantial memory and processing power is required on the routers. Moreover, dynamic routing protocols exchange frequent updates that take bandwidth and device resources. The other limitation is convergence delay in case of topology change or link failure. As a result of these factors, regular IP routing can become increasingly inefficient in larger, more complex ISP scenarios.[\[4\]](#)

1.3. Overview of MPLS Technology

1.3.1. Evolution of WAN Technologies Towards MPLS

Enabling communications across multiple sites, whether within the same enterprise or different organizations, wide area networks (WANs) have an integral role in our increasingly connected world. Several technologies have been introduced over the years to meet the increasing demand of data transport which are distance apart. Of particular importance was the evolution of WANs from technologies like Frame Relay and ATM. Frame relay, overly simple and attractive in its form for inter-site communication, while ATM opened access to additional quality of service management features making it suitable for real-time applications such as voice and video.

But even with their benefits, these technologies quickly revealed deficiencies in flexibility and operating cost - and, as modern IP-based networks continued to scale rapidly over the decades, also in compatibility. The explosive growth of Internet traffic, the rollout of multimedia services and continued evolution of IP infrastructures created demand for a more efficient static or dynamic solution for converged environments.

This is in the context of MPLS (Multiprotocol Label Switching) came about. MPLS was created to take the benefits of switch speed and IP routing flexibility for a more efficient traffic transport across large networks. Consequently, it became more popular and more widely used among telecom operators and big enterprises.[\[5\]](#)

1.3.2. Definition and Operating Principles of MPLS

Multiprotocol Label Switching (MPLS) is a packet switching technology that was defined and standardized by the IETF as of 1997, It can be used to send data around quickly, efficiently, and flexibly in modern networks. MPLS sits between Layer 2 (data link) and Layer 3 (network) of the Open Systems Interconnection (OSI) model.

MPLS is multi-protocol because it can run over various lower-layer technologies such as Frame Relay, ATM and Ethernet. It is also known as “label switching” because it attaches fixed-length labels to packets at an ingress node in the network. These labels also enable routers to make forwarding decisions without looking at the IP address in each hop like traditional IP routing, which gives a great performance benefit.

MPLS provides this flexibility to enable heterogeneous network interconnection in a single, unified infrastructure. This enables operators to carry various types of traffic over a single MPLS infrastructure with an end-to-end architecture that maintains performance, reliability and quality of service. These benefits suited MPLS for ISPs and large enterprise with VPN (Virtual Private Network), Traffic Engineering needs etc. which required quality of service guarantees.[\[14\]](#)

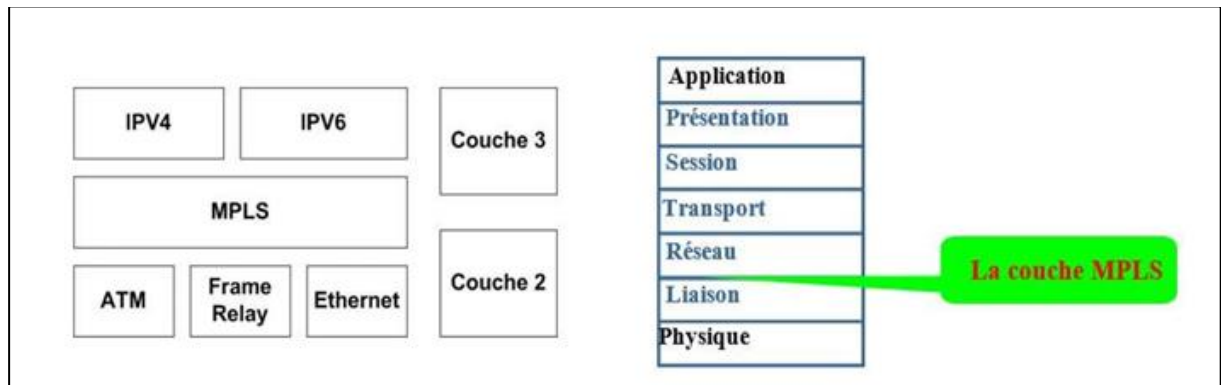


Fig1.1 –The layered MPLS architecture

1.3.3. MPLS Architecture and Main Components

Before going into the detail of the MPLS network architecture and its elements, two basic terms, or definitions should be understood: FEC and LSP. [\[6\]](#)

- **The Forwarding Equivalence Class [EFC]:** It is a class of packets that will share the same features(host address, source IP, destination IP) to accommodate similar QoS mechanisms and encapsulation. Any packets that belong to the same FEC receive identical differentiation treatment while being routed. They are given the same label value making forwarding simpler in the MPLS domain.
- **Label Switched Path (LSP):** is a unidirectional path through the MPLS network established by a specific set of LSrs. That will work like a circuit in a virtual manner. The packets that belong to the same FEC will follow a predetermined path, dictated by the labels (at each hop).

There are two types of LSP:

- The static LSP (manually configured).
- Dynamic LSP (Labeled Switched Path) set up by the network itself automatically.

The basic elements of an MPLS architecture are the following:

- a Customer Edge (CE) device,
- a Label Edge Router (LER),
- And a Label Switching Router (LSR).

The Figure illustrates the operating principle of an LSP as well as all the constituent elements of the MPLS network.

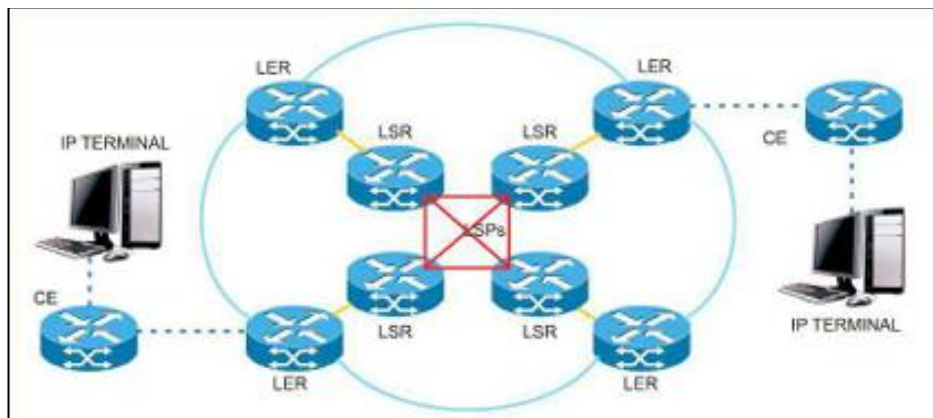


Fig1.2 –MPLS Network Architecture

- **Customer Edge (CE) routers:** Routers that belong to the customers of the MPLS network. They treat packets according to IP addresses and have no knowledge of the underlying MPLS infrastructure.

- **Label Edge Routers (LER):** These are the routers located at the border of the MPLS network. They provide the interconnection between traditional IP networks and the MPLS network. A packet always enters the MPLS network via an LER, called an ingress router (LER), and leaves the MPLS domain also via an LER, called an egress router (LER)

Label Switching Routers (LSRs) Router): These are routers located within the MPLS network, i.e., in the network core. They route packets based on MPLS labels, not on the IP address [E]. They are responsible for switching labels between the ingress router (Ingress LER) and the egress router (Egress LER).

1.3.4. MPLS Labels

MPLS labels are the main element used in MPLS networks, because they replace the IP address for forwarding inside the network. The idea is simple: when a packet enters an MPLS domain, it is assigned a short label, and then the routers inside the network forward the packet using this label instead of checking the full IP address at every hop.

a) Definition of an MPLS label?

- An MPLS label is a small bit of fixed-size value on top of data when it enters into the MPLS domain.
- It serves as a packet identifier through the network.
- Each router reads only the label and then forwards it to wherever it should receive this packet instead of reading full IP address.

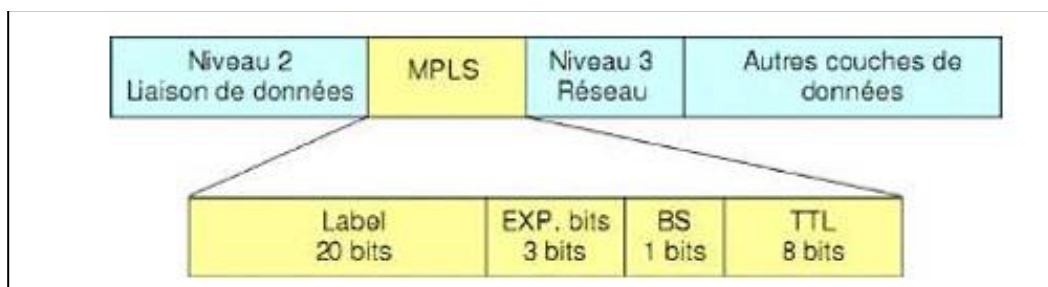


Fig1.3 – Structure of an MPLS header

Label: over 20 bits to designate the packet as belonging to a particular LSP (Label Switched Path). The value is likely changed at every hop within the MPLS network.

EXP ou CoS (Class of Service): A 3-bit field used for Quality of Service (QoS) information. For QoS support, Prefix can be set to allow certain packets to be "superior" over others based on the creation type service

Bit de pile (BS, bottom of stock): if an IP packet contains more than one label stack then this lays out whether the current label is the last one. 1 If this is the last label for that packet, 0 means more labels follow.

TTL (Time to Live): An 8-bit field representing the max lifetime of the packet in hops (or routers). The maximum value is 255. The TTL is decreased by 1 with each hop. If it arrives at before the packet reaches its destination, the concerned router throws out the packet and sends an error message in an ICMP (Internet Control Message Protocol) packet.[\[19\]](#)

b) Why do we use MPLS labels?

Related to make forwarding more efficient within the network.

For less burden on routers, so that routers do not need to examine the IP address at each hop.

For having more organized and performing traffic within ISP networks.[\[20\]](#)

c) How do MPLS labels work:

- The **first router** to touch down on this packet (obviously, the Ingress LER) will be the one that adds the label.
- Inside the MPLS net, the packet travels through LSRs, and these devices process packets based only on the label.
- Every router can swap the label for another one based on its MPLS table. This is called swap.
- The last LER, in this case Egress LER will pop the same label from its top so that it can resume normal IP forwarding.[\[21\]](#)

c) MPLS label operations

- Trois opérations PUSH, POP et SWAP sont implémentées sur les routeurs MPLS :

Push: adding a label when the packet enters the MPLS network.

Swap: replacing a label inside the network.

Pop: removing a label when the packet leaves the MPLS network.[\[28\]](#)

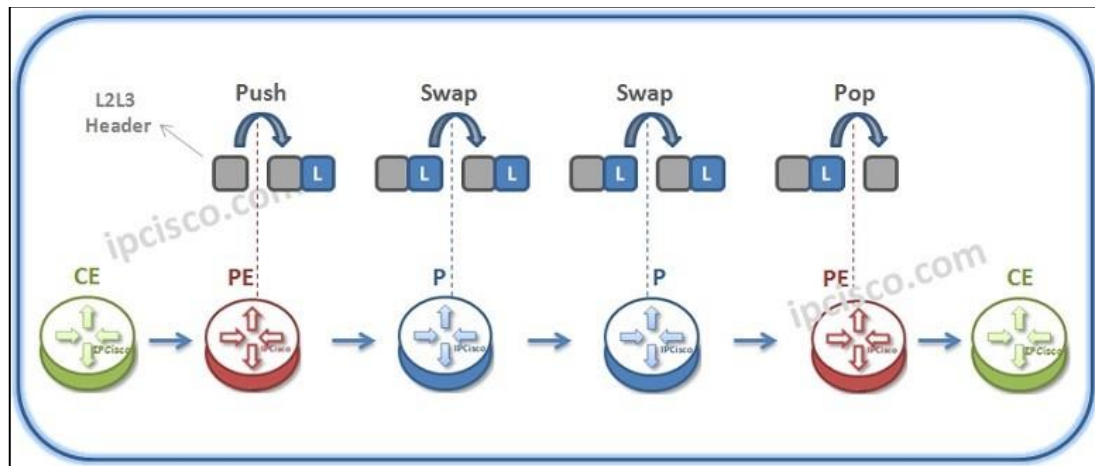


Fig1.4- MPLS Label Operations: Push, Swap, and Pop

Figure explanation :

CE to PE : When a packet enters the MPLS network it starts from customer device CE and reaches PE router.

Push : As the packet enters the MPLS domain, a label is added to the PE router from where it entered. This operation is called Push.

Swap : In the MPLS network itself, the P routers swap out the old label for a new one needed for the next hop. This operation is called Swap.

Swap again : The packet might go through multiple P routers, and the label is replaced at every hop per the LFIB table.

Pop : The PE router, before the packet leaves the MPLS network, removes the label. This operation is called Pop.

PE to CE : When the label gets stripped, it just becomes a normal IP packet which leaves the MPLS domain and hits the customer edge CE. .[\[22\]](#)

1.3.5. Benefits of MPLS

- Faster and more efficient packet forwarding.
- Better traffic engineering and path control.
- Better scalability in large ISP backbone networks.
- Support for Quality of Service
- More efficient use of the provider infrastructure.

1.4. MPLS Services and Traffic Engineering

1.4.1 VPN Services Over MPLS

VPN services over MPLS are used by service providers to interconnect multiple customer sites over a shared MPLS backbone while giving the illusion of a single private network. In this model, traffic between branches does not traverse the public Internet but is carried inside the provider's MPLS infrastructure. The provider builds Layer 3 MPLS VPNs (MPLS L3VPN), creating separate VRFs on PE routers so that each customer's routing information is fully isolated. MP-BGP is then used between PE routers to exchange VPNv4/VPNv6 routes, while MPLS labels in the core steer packets along pre-established LSPs from one site to another. This allows the customer to achieve any-to-any IP connectivity between branches with centralized control in the provider network. Because the service runs over MPLS, the provider can enforce QoS and traffic engineering policies and offer stronger SLAs compared to Internet-based VPNs. From the customer's perspective, each site simply connects its CE router to the provider, and all complexity of label switching and VPN separation remains hidden inside the MPLS core..[\[15\]](#)

1.4.2 Quality of Service and Traffic Engineering in MPLS

Quality of Service (QoS) and traffic engineering (TE) in MPLS are closely related mechanisms that allow service providers to control how traffic flows through the backbone and to guarantee performance for critical applications. MPLS first groups packets into forwarding equivalence classes (FECs) and forwards them along label-switched paths (LSPs), which makes it easier to apply differentiated QoS policies to each class. By combining MPLS with DiffServ, packets can be marked at the edge (for example using EXP or DSCP bits) and then scheduled

and queued inside the core according to their priority, ensuring low delay and jitter for real-time services like voice and video [\[16\]](#)

1.4.3 Operational Challenges and Scalability Limits of MPLS

However, MPLS does come with some operational difficulties as well as scalability constraints even in service provider network. At the same time, MPLS also has the disadvantages, such as consuming more control-plane resources, requiring more memory and many more manual configurations than other technologies in relatively larger ISP environments. This may also limit flexibility and add complexity to network operations, when the provider decides to scale services or re-engineer infrastructure. Thus, operational complexity & scalability are considered as major limitations of traditional MPLS networks. [\[17\]](#)

1.5. Segment Routing

Segment Routing is defined as an approach in which the first router encodes the entire path directly into the packet header as a list of Ordered Segment IDs (SIDs), and each subsequent router simply follows these instructions without storing additional state. This simplification of the network reduces the need and bypasses protocols like LDP or RSVP-TE. Segment Routing can be employed on MPLS (SR-MPLS) or over IPv6 (SRv6).

1.5.1. Principles of Segment Routing

Built on the idea of source routing, namely where the ingress router specifies via what path a packet should take within the network by inserting in to the packet header a list of instructions referred to as segments. Each snippet specifies a specific action, such as forwarding the packet to a particular link, node or service. In comparison, Segment Routing flattens the network architecture as it does not require additional signaling protocols or state information in the network itself and instead encodes path details into packets. This simplifies traffic engineering, reduces operational complexity and increases. [\[18\]](#)

1.5.2. Segment Routing Over MPLS (SR-MPLS)

SR-MPLS is essentially an evolved version of conventional MPLS (RFC 3031) technology. The key to SR operation within a given topology is the notion that instead of establishing LSPs connecting ingress and egress nodes, it will use segments to govern how packets traverse the topology. With standard MPLS, label distribution and traffic path establishment relied on protocols like LDP or RSVP-TE, making the network complex and difficult to manage. It simplifies everything because there are no longer separate signaling protocols that advertise labels, but rather they are known through the routing protocols such as IS-IS or OSPF (SR-MPLS).

Every segment is identified by a Segment ID (SID), and the ingress router inserts ordered SIDs in packet header as MPLS labels. The packet then moves along the path indicated by these labels until it arrives at its destination. However, one of the best benefits from SR-MPLS is that there is no need to maintain per-flow state at every router in the network, only the ingress router should know about the complete path which increases scalability significantly. Additionally, SR-MPLS can utilize the existing MPLS infrastructure without any hardware changes since only a software update is required.[\[10\]](#)

1.5.3. Segment Identifiers and Path Computation

The goal of Segment Routing consists in directing packets through the network based on instructions contained in the packet header, which list segments listed as a sequence of bits per segment identification (SEGMENT ID: SID), which instructs the network to perform desired actions. These sub-types include: a Node-SID that identifies the router itself, an Adjacency-SID which identifies a link between two routers, and Prefix-SID, which is essentially just coding for the route to a given specific node. These SIDs are flooded throughout the network by IGP routing protocols such as IS-IS or OSPF, meaning that every router in the SR domain is aware of all other nodes and links' SIDs. The ingress router builds up an ordered list of SIDs named segment list, this will define the precise path to travel the packet from source until destination. On advanced networks, specialized elements often called a PCE (Path Computation Element) are used to compute optimal paths according with network characteristics such as topology, traffic constraints, latency or bandwidth requirement for the path that needs to take place. In this case the element computes an ideal segment list and sends it the ingress who puts in on top

of its load. Segment Routing's flexible SID types and the ability to compute a path in a centralized or distributed manner allow for powerful, scalable traffic engineering.[\[11\]](#)

1.5.4. IPV6 segment

In the context of SRv6, an IPv6 segment is an IPv6 address (or a structured part of an address) that represents one step in the packet's path and often encodes a specific instruction. A segment can simply identify a node in the network (Node-SID), a particular interface or adjacency (Adjacency-SID), or a service function such as a VPN context or a decapsulation/forwarding behavior (Service-SID like End.DT4 or End.DT6). These IPv6 segments are carried inside the Segment Routing Header (SRH), which contains an ordered list of segments that the packet must follow. As the packet traverses the network, each router processes the active segment, performs the associated behavior (for example, forward to a given next-hop or send into a specific VPN), then moves on to the next segment in the list. This design allows the source (or an edge node) to program an explicit path and a sequence of functions directly into the packet, enabling advanced traffic engineering, VPN services, and service chaining without requiring per-flow state in the core .[\[12\]](#)

Conclusion

This chapter provided the basic background concepts of MPLS technology, which has emerged as one of the key means with respect to the efficient and scalable packet forwarding within service provider networks, It explained the MPLS architecture and forwarding model, which was based on labels and Label Switched Paths, as opposed to traditional hop-by-hop IP routing, The key MPLS services like Layer 3 VPNs along with few other value added service were highlighted to demonstrate the ability of MPLS for Traffic Isolation and Service Differentiation for multiple customers, Also, it will explore the idea of MPLS Traffic Engineering and how by using explicit paths and constraint-based routing (CBR) can optimize resource allocations while preventing node.

The limitations and the complexity of traditional MPLS-TE mechanisms made dependent on Segment Routing to be more flexible and simplified, Next, we gave a brief description of Segment Routing stating how it encodes the path as an ordered list of segments inside the packet header, which reduces control plane overhead, These will lay the foundation for next, on SRv6 (segment routing over IPv6) and role in modern network evolution.

Chapter 2: fundamentals of SRv6 and migration

Introduction

This chapter presents Segment Routing as the next evolution of segment-based architectures by replacing traditional MPLS data-planes for a pure IPv6 solution. This document specifies the SRv6 architecture, header format, segment type used in SRv6 network node format, and BGP protocol extensions necessary to integrate an SRv6 control-plane. This chapter also details the packet routing mechanisms using SRv6 networks, as well as migration strategies leading to SRv6 (indirect and direct evolution approaches) with main technical and operational challenges.

2.1. SRv6

2.1.1. Definition SRv6

SRv6 is a modern networking technology that utilizes the concepts of Segment Routing and incorporates them into the IPv6 protocol, resulting in a uniform, flexible, and programmable data plane. In contrast to SR-MPLS, which encodes segments with MPLS labels, SRv6 employs regular 128-bit IPv6 addresses as its Segment Identifiers (SIDs) and includes a new IPv6 extension header known as the Segment Routing Header (SRH), which carries an ordered list of SIDs representing the path that packet needs to adhere to. It also means that there is no need for a separate MPLS layer, since routing and service encoding can be done with the IPv6 header itself; thus SRv6 simplifies network architecture even further. SRv6 mainly aims to eliminate LDP and RSVP-TE from the network control plane, realize end-to-end programmability of networks, integrate advanced services (L3VPN / L2VPN / Traffic Engineering / Service Function Chaining) on a single unified data transport, and drive migration towards pure IPv6-based networking infrastructures. SRv6 is also designed to be compatible with existing IPv6 forwarding, allowing subsequent transit nodes that do not support SRv6 to forward packets as usual based on the IPv6 destination address which greatly facilitates incremental deployment.[\[26\]](#)

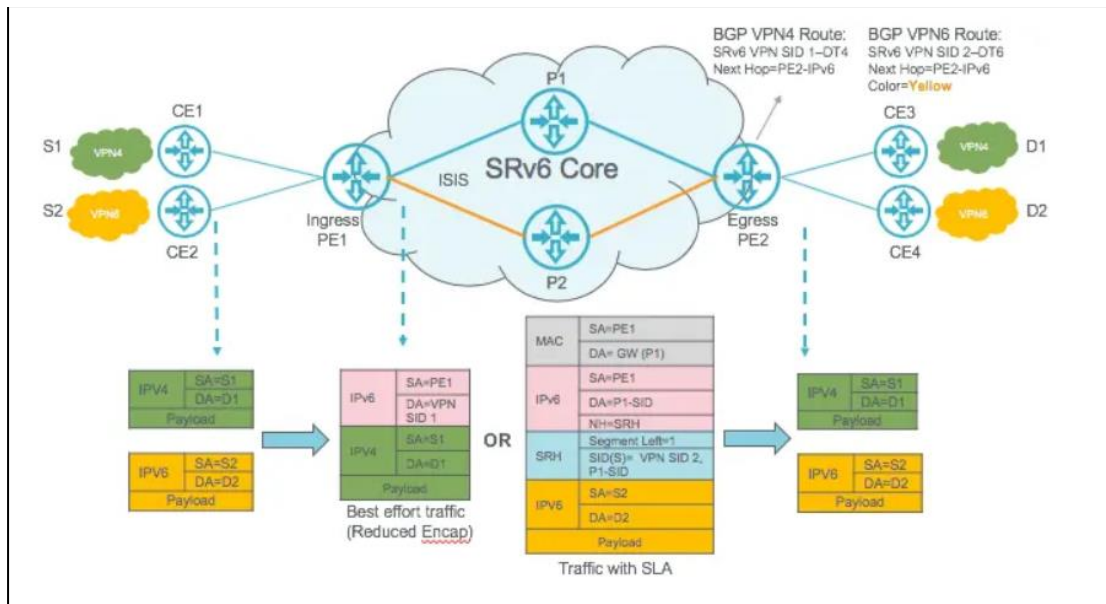


Fig2.1 L3VPN over SRv6 Illustration

2.1.2. SRv6 Architecture

SRv6 architecture leverages the existing IPv6 data plane, which means it uses the standard packet format for IPv6 packets and only appends a new extension header - Segment Routing Header (SRH) that accommodates list of segments directing the packets. An SRv6 Segment ID (SID) is a standard 128-bit IPv6 address divided into three fields: Locator, which identifies the node/network domain, Function, which indicates what to do with the packet at that node and optionally Arguments, which are parameters for that function. The Locator part is advertised as a regular IPv6 prefix by IGP protocols like IS-IS or OSPF, enabling all routers in the network to reach SRv6 nodes via native IPv6 forwarding even if they do not have full SRv6 capability. When a packet crosses into the SRv6 domain, the ingress node encapsulates it with an outer IPv6 header plus an SRH containing an ordered list of SIDs, and as the packet traverses the network, each and every SRv6 capable router reads the next SID from this list, copying it into the IPv6 destination address field while decrementing Segment Left. The SRv6 integration with IPv6 has the most important property of backward compatibility, because all routers that still only understand standard IPv6 and not SRv6 will still be able to forward PVs correctly by simply performing a "normal" longest-prefix-match lookup (it is exactly what they do for "traditional" IP addresses) on the destination address; this makes it very easy for incremental deployments across mixed

networks. This tight integration with IPv6 also means that SRv6 inherits the benefits of the enormous IPv6 address space, native QoS (Quality of Service) handling via the Traffic Class field, flow-aware-load-balancing capability using the Flow Label field and compatibility with modern cloud or SDN or 5G transport architectures.[\[27\]](#)

2.1.3. SRv6 Header

The SRv6 packet is comprised of the basic IPv6 header with a new extension header, the Segment Routing Header (SRH), appended, described in RFC 8754. The SRH consists of an ordered list of 128-bit SIDs to represent the path a packet needs to take, a Segments Left (SL) counter that shows how many segments still need visiting and other fields such as Next Header and Last Entry. Traversing the network, as a packet reaches each SRv6-enabled router, it will read the next SID in line and copy that into the IPv6 destination address field, decrement the SL counter and forward. For that network programming model defined by RFC 8986, each SID consists of two parts: a Locator part for routing the packet to the appropriate node, and a Function part that tells that node which action needs to be taken (End, End. You are content up to 20 October 2023. DT4/End. DT6 for L3VPN services. This model transforms the packet header into a small program that specifies precisely how the packet should be processed at each hop.[\[24\]](#)

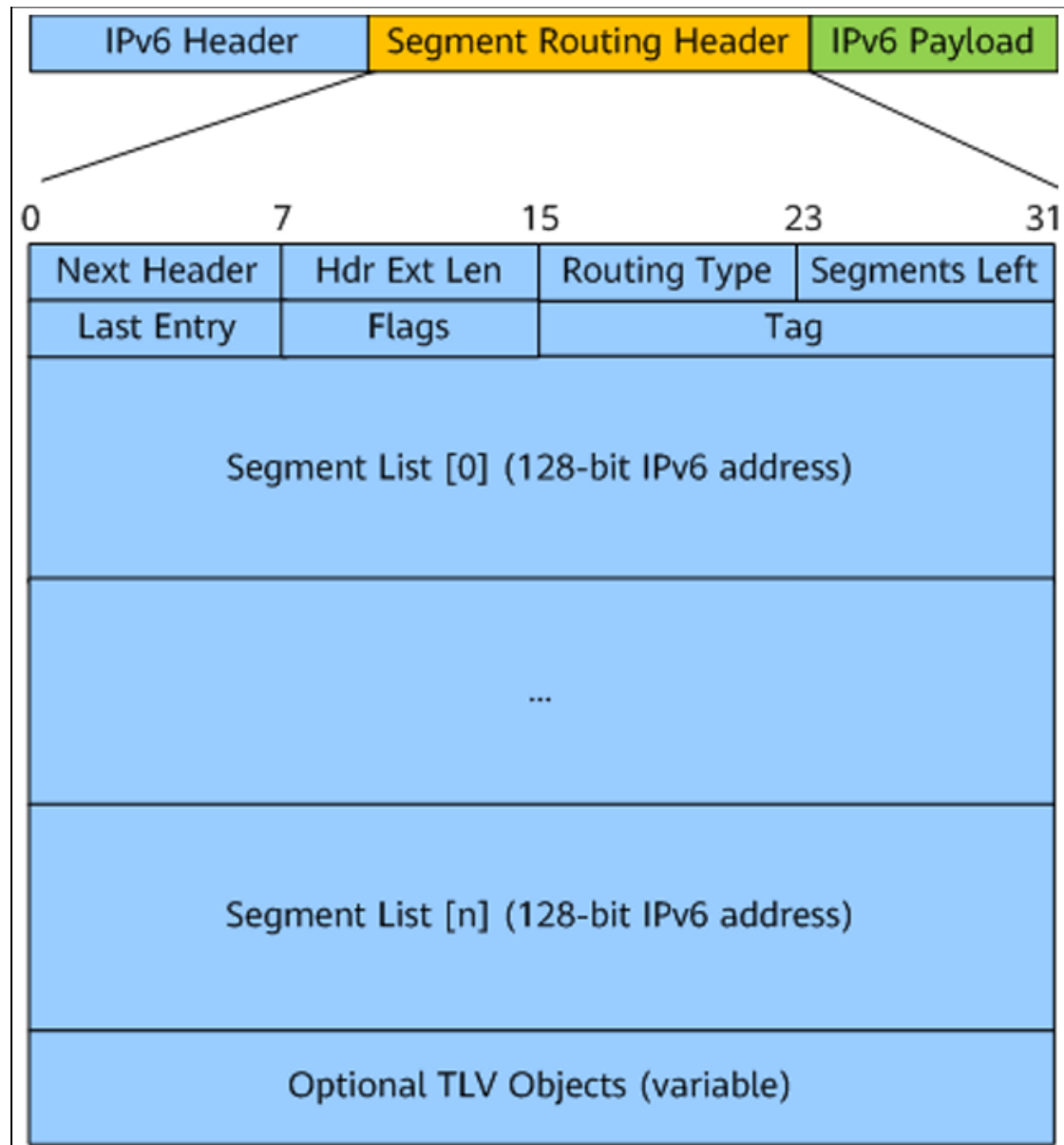


Figure2.2 –Segment Routing Header (SRH) Format

The main information contained in the Segment Routing Header (SRH) in IPv6 is as follows :

Routing Type: a value of 4 in this field identifies the header as being an HR manager

List of segments: it contains the segments (Segment[0], Segment[1], Segment[2], ..., Segment[n]) representing the different nodes or steps that the packet must follow.

Remaining segments (SL, Left Segments): This field indicates the number of segments remaining to be covered and points to the currently active segment.

2.1.4. SRv6 Segment Format

SRv6 programming capabilities are heavily reliant on three fundamental aspects. To begin with, the segment list allows multiple segments to jointly cooperate to construct a specific SRv6 path where the packets follow which path has been programmed in advance. Second, SRv6 utilizes a 128-bit Segment Identifier (SID), allowing for high granularity enabled by the IPv6 addressing space being sufficiently large. Third, SRv6 Segments are network instructions used in Network Programming specifically for the purpose that indicates the data on where to be delivered and how it should be forwarded (SRv6). SRv6 segments are represented by a SRv6 SID, which is simply the extended 128-bit IPv6 address typically broken into three parts: Locator + Function + Arguments — and transmitted in an encapsulated form.[\[24\]](#)

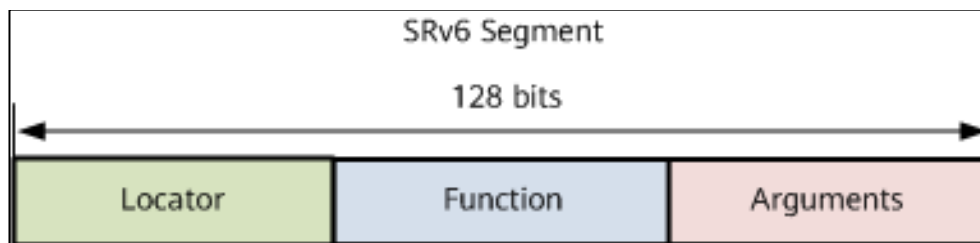


Fig2.3 –SRv6 Segment Format

Locator: The most significant part of the address. It allows standard IPv6 routing to route the packet towards the destination node. Usually locators can be aggregated and are hierarchic thus enhancing scalability of the network.

Field: This intermediate, and less important, field is used to specify what action needs performed on the packet when it reaches at targeted node. These could be forwarding the packet out a certain interface, decapsulating to get to some VPN service or performing an address translation.

Argument (optional): The final portion of the address may include extra values needed by the function. For example these arguments can represent a service id, port number, or any other data related to the processing that is going to be executed.

2.1.5. BGP Protocol Extensions

BGP Protocol Extensions, or Multiprotocol BGP (MP-BGP), are extensions to BGP-4 that allow the protocol to carry routing information for multiple network layer protocols, not only IPv4; according to RFC 4760, this is done through the MP_REACH_NLRI and MP_UNREACH_NLRI attributes, which make BGP suitable for IPv6, VPNs, and other address [23]

2.1.6. Packet Routing in srv6

Next, an example of how SRv6 packet forwarding works. The figure depicts forwarding a packet from host 1 to host 2 (the source and destination of the connection are not necessarily tied to the devices used in each node, but they are in this case). SNODE A B D E C SRv6 Yes NoYesSRV+/ || = nodes with innate support for IPv6 node. Thus we need to program Source node A such that packet is routed via host 2 through node E after going through link B-C and Link C-D.[24]

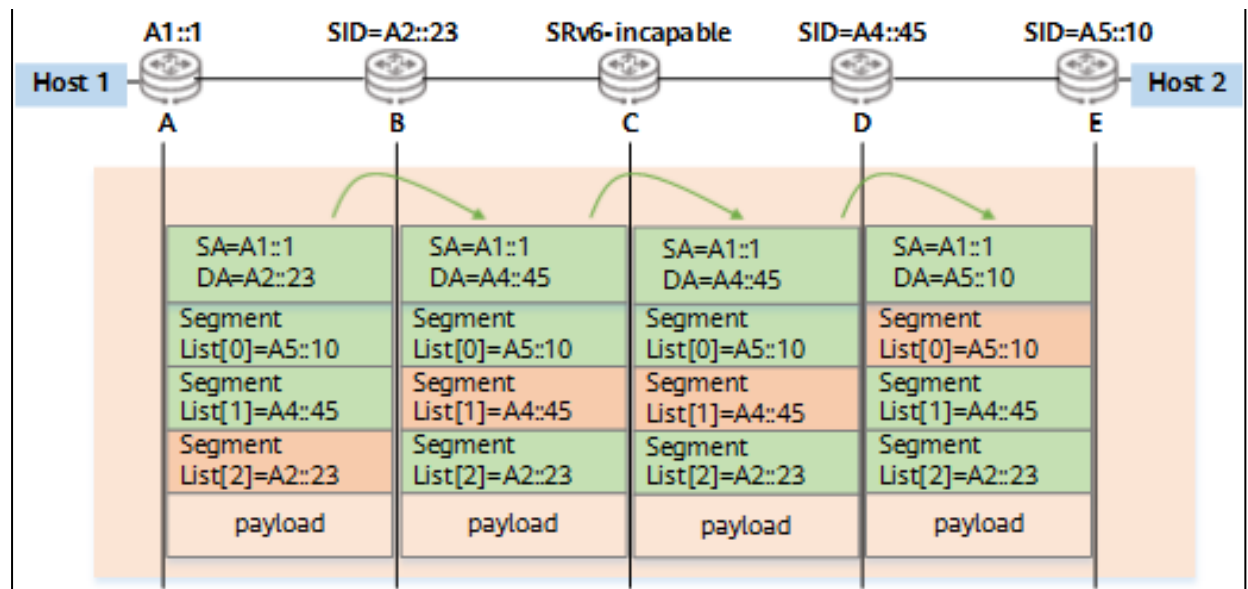


Fig2.4 –SRv6 Packet Transfer Process

1- Source node A encapsulates SRv6 path information carrying the SIDs of B-C and C-D links as well as SID A5::100 advertised by node E (this SID corresponds to an [IPv4 VPN](#) instance of node E) into an SRH. Note that the SIDs are encapsulated in reverse order, and the SL value in the encapsulated

packet is 2 as three SIDs exist in total. The SL field points to the segment list to be processed, which is Segment List [2]. Consequently, node A copies the value of Segment List [2] to the DA field in the outer [IPv6](#) header, searches the corresponding IPv6 [routing](#) table according to the longest match rule, and then forwards the packet to node B.

2- After receiving the packet, node B searches its local SID table (which stores the SRv6 SID information it generates) based on the destination address in the outer IPv6 header and finds a matching End.X SID. According to the instructions defined by the End.X SID, node B decrements the SL value by 1, updates the DA field in the outer IPv6 header with the Segment List [1] value, and then sends the packet over the link (B-C) bound to the SID.

3- The packet arrives at node C. Node C cannot identify the SRH as it does not have any SRv6 capabilities. As such, it treats the packet as a common IPv6 packet. Specifically, it searches the corresponding IPv6 routing table according to the longest match rule and then forwards the packet to node D, which is represented by the current destination address.

4- After receiving the packet, node D searches its local SID table based on the destination address A4::45 and finds a matching SID (End.X SID). Similar to node B, node D decrements the SL value by 1, updates the DA field in the outer IPv6 header with A5::100, and then sends the packet over the link bound to the End.X SID.

5- After receiving the packet, node E searches its local SID table based on A5::100 and finds a matching SID (End.DT4 SID). According to the instructions defined by the SID, node E decapsulates the packet by removing the IPv6 header, searches the IPv4 routing table of the VPN instance bound to the End.DT4 SID, and ends the process by sending the inner IPv4 packet to host 2.

2.2. Strategies for Migration to SRv6

2.2.1. Motivation for Migrating Toward SRv6

The push for SRv6 adoption arises from the increasing constraints of legacy MPLS networks and new requirements presented by modern digital services. Hence, while MPLS initially provided many advantages for ISPs, multiple IPv4 limitations are making it increasingly difficult for them to handle emerging services like 5G, cloud computing, IoT or AI-based applications. Also note that traditional MPLS requires multiple signaling protocols (LDP, RSVP-TE) to work is operationally complex and costly while the SRv6 control plane only needs to use IPv6 and IGP routing protocols, greatly simplifying the control plane. SRv6 natively enables network programmability, allowing the operator to define sophisticated per-packet behaviors and modify their effects dynamically through APIs without specialized hardware not possible in traditional MPLS. A further strong motivator is the closer integration of SRv6 with cloud infrastructure, as it allows extending L2 and L3 VPN services directly to cloud endpoints without complex service termination in dedicated gateways. Early adopters such as SoftBank in Japan and Bell Canada have already shown that progressive migration to SRv6 in production networks is practical, cost-effective, and opens new services not possible with legacy systems. This means that SRv6 becomes the natural evolution face of ISPs who need for simplify their infrastructure, enable future services and overcome the complexity of MPLS networks.[\[13\]](#)

2.2.2. Indirect Migration Strategies

5G networks have now emerged. IPv6 addresses are also sought to facilitate users build up their VPN. To this end, SRv6 [10] was proposed: by extending the IPv6 header itself to mimic label forwarding semantics but maintaining the existing IPv6-forwarding technology stack. SRv6 instantiates specific IPv6 addresses as Segment Identifiers (Segment IDs or SIDs), which each have a well-defined purpose. These SIDs are used for performing different operations to make the VPN deployment easier and to use more flexible path planning.

2.2.3. Technical and Operational Challenges of Migration

MPLS LDP is a widely used tunneling technology that is commonly deployed in transport networks. Therefore, during the evolution from MPLS LDP to SR-MPLS, LDP and SR-MPLS will coexist for a long period of time. The interworking mechanism between LDP and SR-MPLS makes it possible to connect LDP-based and SR-MPLS-based networks, ensuring MPLS forwarding across both types of networks.

MPLS to SR-MPLS migration strategy : MPLS LDP is a widely deployed tunneling technology that is commonly used in transport networks. During the migration from MPLS LDP to SR-MPLS, both LDP and SR-MPLS are therefore expected to coexist for an extended period. The interworking mechanism between LDP and SR-MPLS enables connectivity between LDP-based and SR-MPLS-based networks, ensuring seamless MPLS forwarding across these two types of domains.

Migration strategy from SR-MPLS to SRv6 : In the previous section, the strategies for evolving from MPLS to SR-MPLS were described. Once the network is ready for SRv6 deployment, the evolution from SR-MPLS to SRv6 can be considered. The overall migration process can be divided into the following phases:

Initial state: L3VPN over SR-MPLS is already deployed.

Phase 1: Dual-stack deployment:

- (a) Evolve the IPv4 IGP to a dual-stack IPv4/IPv6 IGP.
- (b) Evolve the BGP peering relationship to dual-stack BGP/BGP4+.
- (c) Deploy SRv6 and SR-MPLS tunnels in a dual-stack model for L3VPN services.

Phase 2: Migrate L3VPN services to SRv6 tunnels, meaning that the service tunnels evolve from SR-MPLS to SRv6.

Phase 3: Remove the remaining SR-MPLS tunnels.

2.2.4. Direct Evolution Strategies

Direct evolution means that a network migrates directly from IPv4/MPLS to SRv6. The essential change among this change is the process of shifting over from IPv4 network to IPv6. The network also needs to support dual-stack IPv4/IPv6 coexistence in order to the path for migration easier and faster to deploy. When the network has completed the transition to IPv6, then establishing SRv6 capabilities from day one is simple. Perhaps the most compelling feature of SRv6 is its on-demand incremental migration. In the example shown above, SRv6 only need to be implemented in edge (PE) nodes of L3VPN over pure SRv6 BE, and the rest of the transit node just need to be able forward IPv6 packets without any functionality upgraded. There is no need to do an end-to-end upgrade, only key nodes such as ASBRs and PEs need SRv6 support, which makes further features like SERv6 TE very easy to implement in the future.[\[25\]](#)

a-SRv6 and IPMPLS coexistence strategy

Network migration or upgrades do not usually take place overnight. SRv6-compliant equipment may need to coexist with non-SRv6-compliant devices for an extended period of time. When there is no clear boundary between the deployment areas of these two types of equipment, a coexistence strategy must be adopted. This strategy typically relies on two main methods:

- the dual-stack IPv4/IPv6 coexistence method
- the coexistence overlay method

In the following, we describe these coexistence methods in more detail :

Dual-stack coexistence When the dual-stack coexistence solution is adopted, two main aspects must be considered: control-plane protocols and transport services :

- **Control protocols** : Dual-stack IGP coexistence: for example, enabling IPv4/IPv6 dual-stack for IS-IS, or running OSPF and OSPFv3 in parallel on the same network, BGP dual-stack: when both MPLS VPN and SRv6 VPN are used, they rely on BGP as the signaling protocol. MPLS VPN uses IPv4 addressing, whereas SRv6 uses IPv6 addressing. To ensure coexistence between these services, BGP EVPN is used.

- **Transport service** New services are carried over SRv6, while existing services continue to be transported over MPLS and can be seamlessly migrated to SRv6 over time. If a specific flow or service must traverse a given set of nodes and some of these nodes are not yet upgraded, that flow will be transported using both SRv6 and MPLS through coexistence overlay equipment.

Coexistence overlay The overlay coexistence solution can be used when the ingress and egress nodes of a service have been upgraded to SRv6-capable equipment, while the transit nodes either do not need to be upgraded or cannot be upgraded in the short term (for example, when the service crosses a third-party network, or when some transit hardware does not support SRv6). In this case, an IPv6 backbone underlay can be deployed on the transit nodes to provide end-to-end SRv6 services. If stricter requirements such as traffic optimization arise later, selected strategic nodes can be incrementally upgraded to support SRv6, depending on the specific use case.

Conclusion

This chapter has introduced the basic principles of SRv6, including its architecture, the extension of IPv6 header by Segment Routing Header (SRH), and the structure of an SRv6 segment that encodes both network topology and network functions. It demonstrates how BGP protocol extensions (in particular Multiprotocol BGP) allow for the carrying of information related to SRv6 and provides for its integration into existing control-plane functions. SRv6 packet routing was subsequently discussed in the context of service-function programming with example shows how ordered lists of SIDs permit source programmed end-to-end and service functions (in particular, micro-services) directly into the packet.

The chapter also explored migration strategies to SRv6, focusing on indirect (through SR-MPLS) and direct evolution of node capabilities from IPv4/MPLS to SRv6. Migration is characterized by various technical and operational obstacles such as the coexistence with legacy devices, dual-stack IPv4/IPv6 deployment as well as option for SRv6 functionality to be incrementally enabled at a limited number of nodes. Finally, it was highlighted that SRv6 supports an incremental, on-demand migration model allowing operators to leverage their investments in IPv6 infrastructure and deploy new

services and traffic engineering capabilities without requiring end-to-end disruptive upgrades.

Chapter 3: Implementation, Coexistence, and Verification of MPLS-to-SRv6 Migration

3.1. Introduction

This chapter represents the heart of the graduation project, since it is a detailed and correct documentation for each step based on reality that has been followed, as we went from completely traditional MPLS L3VPN network to SRv6 architecture using EVE-NG virtual environment with Huawei NE40E devices. This builds on realistic migration scenarios, which are of the sort undertaken by telecommunications companies and Internet service providers (ISPs) as they evolve their legacy network infrastructures to next generation architectures. Why, What & How of this chapter is that in addition to describing the technical implementation steps, it analyzes each stage of engineering decisions and discusses the stack constraints encountered per step and how innovative solutions were drawn out by the team to overcome them. Next in this chapter, it describes the automation framework developed using Python and Netmiko library for automating configurations and verification tasks to improve migration accuracy while reducing human intervention

3.1.1. Objectives of the Implementation Phase

This phase is oriented towards pursuing the objectives laid out in a set of technical and methodological goals to be integrated:

- At first begin with the build of an fully functional MPLS L3VPN network and test all VPN Services working before starting the migration.
- With dual-stack IS-IS, it is possible to add IPv6 as a dual underlay in parallel with an existing IPv4 infrastructure without disturbing current services.
- Configure SRv6 architecture at PE devices define as well as initiate SRv6 locators and functional SIDs (End. DT4) associated with each VPN.
- Show that MPLS and SRv6 can fully coexist on the same infrastructure, with some services remaining in MPLS while others are migrated to SRv6.
- Migrate VPN_A and VPN_B to SRv6 however ensure service continuity at every step of the transition time frame.
- Create an Automation framework in python and use netmiko for creating scalable automation for configuration and verification.

- Record the shortcoming of VRP version V800R011C00SPC607 and provide the workaround to work with that feature or capability based upon what you test and prove it works

3.1.2 Adopted Migration Methodology

The project used a **Progressive migration methodology** based on the principle of zero service impact (**Zero Service Impact**) The methodology is based on passing through sequential stages denoting a part of the migration, in which each layer contains an additional capability but leaves existing capabilities untouched or mildly modified, and where each next layer only enters production with full verification of its predecessor before. This methodology represents how large telecoms actually build their networks and due to the nature of telecommunications, service interruption can not be tolerated in any manner for even a temporary timeframe. This method has been very successful according to the test results each stage generated, which have consistently confirmed that whatever operation the previous services offered will continue to work as it did before, with the addition of new capabilities.

3.2 Work Environment and Experimentation

The work environment represents the foundation upon which the entire practical implementation is based. This environment was designed to faithfully simulate the real infrastructure of an ISP network, while providing sufficient flexibility for experimentation and testing without the costs of physical hardware.

3.2.1 Physical Environment

The project relies on a fully virtual environment built on a host computer with technical specifications that allow a sufficient number of virtual machines to run simultaneously:

Tab3.1 – Technical Specifications of the Experimental Environment

Specification	Value
Computer type	ASUS TUF Gaming FX50
Processor (CPU)	Ryzen 5 5600g
Memory (RAM)	16gb
Storage capacity	512 gb
Host operating system	Windows 11
Virtualization software	VMware Workstation Version26.0.0
Virtual network system	EVE-NG Community Edition

It should be noted that running seven virtual Huawei NE40E instances at once needs a lot of computing resources (especially RAM), because each NE40E instance consumes between 2GB of memory in normal mode.

3.2.2 Software Environment

The software environment consists of several integrated components that together form the project's complete working system:

Tab3.2 – Software Components and Versions Used in the Project

Component	version	role
EVE-NG Community	v6.0.2-4	Network emulation platform
Huawei VRP	V800R011C00SP C607B607	NE40E → NE40E router operating system
Python	3.14.5	Automation programming language
Netmiko	4.7.0	Device communication library
VMware Workstation	26.0.0	Virtualization environment
SecureCRT / PuTTY	latest release (0.83)	Device access client
WinSCP	v6.5	It is a file manager that works to transfer files from a system to VMware.

This table 3.2 presents the main components adopted in the project, along with their versions and their respective roles within the experimental environment. It also shows that EVE-NG, Huawei VRP, Python, Netmiko, VMware Workstation, and access tools such as SecureCRT/PuTTY formed the software foundation required for simulation, automation, and device connectivity.

a. EVE-NG Platform

EVE-NG (Emulated Virtual Environment - Next Generation) is one of the most widely used network simulation platforms in academic and professional environments. It allows the operation of real network device images in a virtual environment, making testing scenarios as close as possible to the real production environment. Its web interface is used to control devices and manage connections.

b. Huawei NE40E Devices

The Huawei NE40E is one of the most common edge routers (PE) in ISP networks, and it forms the backbone of large commercial networks. This device runs Huawei VRP (Versatile Routing Platform), which provides broad support for modern protocols including MPLS, MP-BGP, IS-IS, and SRv6. The version used in this project is V800R011C00SPC607B607, which represents a balance between stability and support for basic SRv6 technologies.

c. Python and Automation Environment

Python 3.14.5 is used together with Netmiko 4.7.0 as the automation framework. Netmiko provides a unified programming interface for communicating with different network devices over SSH and Telnet, which simplifies the process of sending commands and reading outputs in a structured programmatic way.

3.2.3 Network Topology

The network topology is designed to represent a miniature ISP architecture containing all the essential elements needed to clearly and completely illustrate the MPLS to SRv6 migration scenario .

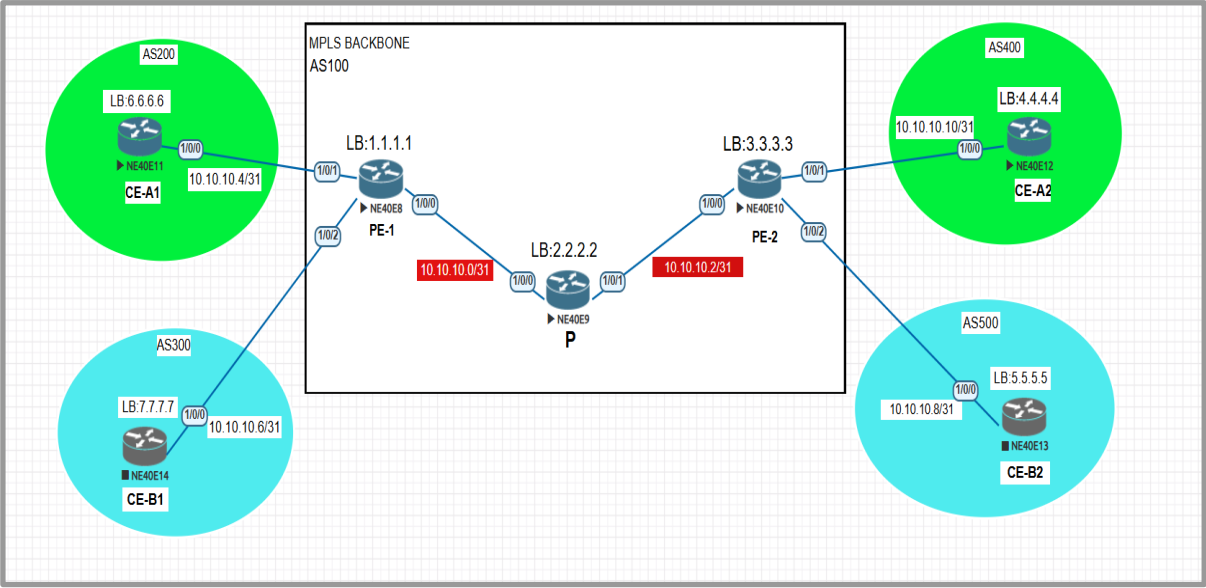


Fig3.1– Overall Topology of the MPLS to SRv6 Migration Lab

This figure 3.1 illustrates the overall testbed topology used in the simulation, with the MPLS backbone centered around the PE-1, P, and PE-2 devices and connected to customer networks CE-A1, CE-B1, CE-A2, and CE-B2 through Loopback addresses and /31 links. It also shows the distribution of the different AS domains, providing a realistic environment suitable for studying the MPLS to SRv6 migration scenario.

a. General Architecture

The network consists of seven Huawei NE40E devices distributed across three distinct functional layers, consistent with the classical ISP network model:

Tab3.3 – Table of Device Roles and Loopback Addresses

device	type	role	Loopback IPv4	Loopback IPv6	Number AS
PE-1 (NE40E8)	Provider Edge	Provider network edge — serving VPN_A and VPN_B	1.1.1.1/32	FC00:1::1/128	AS100
P (NE40E9)	Provider Core	Network core — redirection only	2.2.2.2/32	FC00:1::2/128	AS100
PE-2 (NE40E10)	Provider Edge	Provider network edge — serving VPN_A and VPN_B	3.3.3.3/32	FC00:1::3/128	AS100
CE-A1 (NE40E11)	Customer Edge	VPN_A Client Edge — Part 1	6.6.6.6/32	FC00:A:1::6/128	AS200
CE-A2 (NE40E12)	Customer Edge	VPN_A Client Edge — Part 2	4.4.4.4/32	FC00:A:2::4/128	AS400
CE-B1 (NE40E14)	Customer Edge	VPN_B Client Edge — Part 1	7.7.7.7/32	FC00:B:1::7/128	AS300
CE-B2 (NE40E13)	Customer Edge	VPN_B Client Edge — Part 2	5.5.5.5/32	FC00:B:2::5/128	AS500

This table 3.3 summarizes the network devices used in the topology, including each device type, role, and its IPv4 and IPv6 Loopback addresses, along with the AS number. It also highlights the separation between provider and customer devices, reflecting the logical structure of the lab.

b. Links Between Devices

The following table shows the physical links between the devices and the addresses assigned to each link in both the IPv4 and IPv6 layers.

Tab 3.4 –Table of Links and Addresses Between Network Devices

Link	Interface PE-1/P	Interface PE-2/CE	address IPv4	address IPv6
PE-1 ↔ P	Eth1/0/0	Eth1/0/0	10.10.10.0/31	FC00:0:12::/127
P ↔ PE-2	Eth1/0/1	Eth1/0/0	10.10.10.2/31	FC00:0:23::/127
PE-1 ↔ CE-A1	Eth1/0/1	Eth1/0/0	10.10.10.4/31	FC00:A:12::/127
PE-2 ↔ CE-A2	Eth1/0/1	Eth1/0/0	10.10.10.10/31	FC00:A:23::/127
PE-1 ↔ CE-B1	Eth1/0/2	Eth1/0/0	10.10.10.6/31	FC00:B:12::/127
PE-2 ↔ CE-B2	Eth1/0/2	Eth1/0/0	10.10.10.8/31	FC00:B:23::/127

This table 3.4 presents the interfaces used between provider and customer devices, together with the corresponding IPv4 and IPv6 addresses for each link. It also reflects the interconnection structure within the topology, ensuring clarity of both the physical and logical network design.

c. VPN_A and VPN_B Design

The network includes two distinct VPN services representing two different customers of the telecommunications company:

Property	VPN_A	VPN_B
Devices	CE-A1 (AS200) ↔ CE-A2 (AS400)	CE-B1 (AS300) ↔ CE-B2 (AS500)
Route Distinguisher	200:1	300:2
Route Target	200:1 (import/export)	300:2 (import/export)
Initial phase mode	MPLS VPNv4	MPLS VPNv4
Post-coexistence mode	MPLS VPNv4 (لا يزال)	SRv6 End.DT4
Final mode	SRv6 End.DT4	SRv6 End.DT4
SID on PE-1 (final)	FC00:A:1::200	FC00:A:1::100
SID on PE-2 (final)	FC00:A:3::200	FC00:A:3::100

Tab 3.5 –Table of VPN Service Properties in the Migration Scenario

This table 3.5 summarizes the main characteristics of the VPN_A and VPN_B services within the topology, including the associated devices, Route Distinguisher, Route Target, and the transition phases from MPLS VPNv4 to SRv6 End.DT4. It also shows the final SID addresses on PE-1 and PE-2 for each service, reflecting the service evolution through coexistence and final migration stages.

3.3 Building the Basic MPLS L3VPN Network

Developing the foundation MPLS L3VPN topology is the main kickoff of gradual migration project towards SRv6 based scenarios. This phase was designed very carefully on virtual Huawei NE40E devices from EVE-NG, treated as the prerequisite to have a functioning network which will be used for any following migration steps. You should fully confirm that the MPLS network works fine before carrying out any SRv6 related modification.

3.3.1 Device Roles in the Network

MPLS L3VPN architecture is based on a clear functional division among three types of devices: PE, P, and CE. The following table shows the precise functional role of each device in the implemented topology :

Tab 3.6 –Table of Device Descriptions and Roles in the Topology

Device	type	System name	Loopback	Role
NE40E8	Provider Edge (PE)	PE-1	1.1.1.1/32	Network edge – serves VPN_A and VPN_B for the first customer site.
NE40E9	Provider Core (P)	P	2.2.2.2/32	Core network – forwards MPLS labels only, no VPN services.
NE40E10	Provider Edge (PE)	PE-2	3.3.3.3/32	Network edge – serves VPN_A and VPN_B for the second customer site.
NE40E11	Customer Edge (CE)	CE-A1	6.6.6.6/32	VPN_A customer edge – first site (AS200).
NE40E12	Customer Edge (CE)	CE-A2	4.4.4.4/32	VPN_A customer edge – second site (AS400).
NE40E14	Customer Edge (CE)	CE-B1	7.7.7.7/32	VPN_B customer edge – first site (AS300).
NE40E13	Customer Edge (CE)	CE-B2	5.5.5.5/32	VPN_B customer edge – second site (AS500).

This table 3.6 presents the devices used in the proposed architecture, including each device type, system name, Loopback address, and functional role within the network. It also highlights the separation between provider and customer devices, reflecting the logical organization of the lab design.

Note: It should be pointed out that the P router does not contain any VPN configuration and is not aware of the customers or their services. Its role is limited to forwarding MPLS packets based solely on labels, which constitutes the core principle of MPLS forwarding and the main performance advantage of this technology.

3.3.2 IPv4 Addressing Plan

The first aspect of any network project is to design the address that will be used by every protocol — this is called the addressing plan. For the inter-device links, /31 masks are taken from 10.10.10.0/24 range (which is a normal practice in ISP networks to save addresses for point-point links).

Tab3.7 –Table of IPv4 Addressing Plan

Link	network	PE-1	P	PE-2	CE
Loopbacks	N/A	1.1.1.1/32	2.2.2.2/32	3.3.3.3/32	—
PE-1 ↔ P	10.10.10.0/31	10.10.10.0	10.10.10.1	—	—
P ↔ PE-2	10.10.10.2/31	—	10.10.10.2	10.10.10.3	—
PE-1 ↔ CE-A1 (VPN_A)	10.10.10.4/31	10.10.10.4	—	—	CE-A1: 10.10.10.5
PE-2 ↔ CE-A2 (VPN_A)	10.10.10.10/31	—	—	10.10.10.10	CE-A2: 10.10.10.11
PE-1 ↔ CE-B1 (VPN_B)	10.10.10.6/31	10.10.10.6	—	—	CE-B1: 10.10.10.7
PE-2 ↔ CE-B2 (VPN_B)	10.10.10.8/31	—	—	10.10.10.8	CE-B2: 10.10.10.9

This table 3.7 presents the addressing scheme used in the test network, showing the distribution of IP addresses across Loopback interfaces and the links between PE, P, and CE devices. It also uses /31 masks for inter-device links, which is a common practice in service provider networks to preserve addressing space.

Engineering Note: Using /31 on point-to-point links saves IP addresses compared to the traditional /30, and it is officially supported by RFC 3021 and commonly used in modern ISP networks.

As for the interface configuration on the devices:

On PE-1:

```
system-view
sysname PE-1

interface Ethernet1/0/0
 ip address 10.10.10.31 ← Link to P
 quit

interface LoopBack0
 ip address 1.1.1.1 32 ← Device identity in IS-IS, MPLS, and BGP
 quit
commit
```

On PE-2:

```
System-view
sysname PE-2

interface Ethernet1/0/0
 ip address 10.10.10.3 31
 quit

interface LoopBack0
 ip address 3.3.3.3 32
 quit
commit
```

On P:

```
system-view
sysname P

interface Ethernet1/0/0
 ip address 10.10.10.1 31
 quit

interface Ethernet1/0/1
 ip address 10.10.10.2 31
 quit

interface LoopBack0
 ip address 2.2.2.2 32
```

```
quit
commit
```

On CE-A1:

```
system-view
sysname CE-A1

interface Ethernet1/0/0
 ip address 10.10.10.5 31
 quit

interface LoopBack0
 ip address 6.6.6.6 32
 quit

commit
```

On CE-A2:

```
system-view
sysname CE-A2

interface Ethernet1/0/0
 ip address 10.10.10.11 31
 quit

interface LoopBack0
 ip address 4.4.4.4 32
 quit

commit
```

On CE-B1:

```
system-view
sysname CE-B1

interface Ethernet1/0/0
 ip address 10.10.10.7 31
 quit

interface LoopBack0
 ip address 7.7.7.7 32
```

```
quit
commit
```

On CE-B2:

```
system-view
sysname CE-B2
interface Ethernet1/0/0
 ip address 10.10.10.9 31
quit

interface LoopBack0
 ip address 5.5.5.5 32
quit

commit
```

The loopback interface is one of the most important interfaces in an MPLS network, as it is used as the Router-ID in IS-IS, BGP, and as the MPLS LSR-ID. This address must be reachable from all network devices through IS-IS so that LDP and BGP sessions can be successfully established.

3.3.3 IS-IS Configuration

IS-IS (Intermediate System to Intermediate System) was selected as the interior gateway protocol (IGP) on the core routers due to several technical advantages that make it an ideal choice for ISP networks: it supports carrying both IPv4 and IPv6 information within the same LSP without requiring separate processes, it provides faster convergence compared to OSPF in large-scale networks, and it supports extended SRv6 TLVs when using the wide cost style.

a. NET Network Identifier

Each device in IS-IS is identified by a unique Network Entity Title (NET) following the format AFI.Area.SystemID.SEL. The following table shows the NET assigned to each device :

Tab3.8 –Table of IS-IS Configuration Parameters for Core Devices

device	Network Entity Title (NET)	System ID	Area ID
PE-1	49.0001.0010.0100.1001.00	0010.0100.1001	49.0001
P	49.0001.0020.0200.2002.00	0020.0200.2002	49.0001
PE-2	49.0001.0040.0400.4004.00	0040.0400.4004	49.0001

This table 3.8 presents the network entity title, system ID, and area ID for the PE-1, P, and PE-2 devices in the IS-IS configuration. A Level-1-2 setup was adopted to provide greater routing flexibility within the network.

All devices were configured in a Level-1-2 setting to allow routing flexibility for participating both in the local Level-1 database and the global Level-2 database. This setup is ideal for a lab network and is often used in single-area ISP networks.

b. IS-IS Configuration on the Devices (Practical Example):

PE-1 :

```

system-view
isis 1
 is-level level-1-2
 network-entity 49.0001.0010.0100.1001.00
 quit

interface Ethernet1/0/0
 isis enable 1
 isis circuit-type p2p ←Essential on links /31
 quit

interface LoopBack0
 isis enable 1 ← 1.1.1.1/32 is announced in IS-IS
 quit
commit

```

PE-2 :

```

system-view
isis 1
 is-level level-1-2
 network-entity 49.0001.0040.0400.4004.00
 quit

interface Ethernet1/0/0
 isis enable 1

```

```
isis circuit-type p2p
quit

interface LoopBack0
isis enable 1
quit

commit
```

P :

```
isis 1
 is-level level-1-2
 network-entity 49.0001.0020.0200.2002.00
 quit

interface Ethernet1/0/0
 isis enable 1
 isis circuit-type p2p
 quit

interface Ethernet1/0/1
 isis enable 1
 isis circuit-type p2p
 quit

commit
```

Engineering Note: Enabling circuit-type p2p on /31 links is necessary to avoid the unnecessary Designated IS (DIS) mechanism on point-to-point links, which speeds up IS-IS convergence and reduces overhead.

3.3.4 MPLS and LDP Configuration

MPLS (Multiprotocol Label Switching) provides a label-based packet forwarding mechanism instead of relying on IP addresses, which achieves high-speed routing and enables the construction of L3VPN services. The LDP (Label Distribution Protocol) is responsible for automatically negotiating and distributing labels between neighboring devices.

a. Enabling MPLS and LDP on the Devices:

PE1 :

```
system-view
mpls lsr-id 1.1.1.1      ← Device identification in MPLS
mpls
  quit

mpls ldp
  quit

interface Ethernet1/0/0 ←Activating MPLS and LDP on the link towards P
  mpls
  mpls ldp
  quit
commit
```

PE2:

```
system-view
mpls lsr-id 3.3.3.3
mpls
quit

mpls ldp
quit

interface Ethernet1/0/0
Mpls
mpls ldp
quit

commit
```

P:

```
system-view
mpls lsr-id 2.2.2.2
mpls
quit

mpls ldp
quit

interface Ethernet1/0/0
  mpls
  mpls ldp
```

```

quit

interface Ethernet1/0/1
 mpls
 mpls ldp
quit

commit

```

Tab3.9 –Table of MPLS LDP Settings on Core Devices

device	MPLS LSR-ID	Interfaces enabled for MPLS/LDP
PE-1	1.1.1.1	Ethernet1/0/0 (towards P)
P	2.2.2.2	Ethernet1/0/0 (towards PE-1) + Ethernet1/0/1 (towards PE-2)
PE-2	3.3.3.3	Ethernet1/0/0 (towards P)

This table 3.9 shows the MPLS LSR-ID assigned to each device, along with the interfaces where MPLS/LDP was enabled within the core network. It also indicates that MPLS activation was limited to provider links only, excluding customer-facing interfaces, which is consistent with traditional MPLS network design.

Engineering Note: MPLS is never enabled on CE interfaces because customers do not participate in the MPLS network; instead, they connect to it through eBGP, completely isolated from label switching.

b. MPLS Forwarding Mechanism

After LDP is enabled and labels are exchanged, LSPs are established between the PE devices through the P device. When the packet arrives from CE-A1 at PE-1, PE-1 adds a stack of labels: an outer label to identify the path toward PE-2, and an inner label to identify the target VPN instance. The P device then forwards the packet based only on the outer label, without inspecting the packet contents or knowing the VPN.

3.3.5 Creating VPN Instances

The VPN Instance (or VRF — Virtual Routing and Forwarding) is the basic unit that provides isolation between different VPN customers. Each VPN Instance has a

completely separate routing table, allowing different customers to use overlapping address ranges without any conflict.

a. VPN Instances on PE-1 (and also PE-2)

On PE-1 and PE-2:

```

sys
ip vpn-instance A
  ipv4-family
    route-distinguisher 200:1 ←VPN_A routes are distinguished globally.
    vpn-target 200:1 export-extcommunity
    vpn-target 200:1 import-extcommunity
  quit
quit

ip vpn-instance B
  ipv4-family
    route-distinguisher 200:2 ←VPN_B routes are distinguished globally.
    vpn-target 200:2 export-extcommunity
    vpn-target 200:2 import-extcommunity
  quit
quit
commit

```

Tab3.10 –Table of VPN_A and VPN_B Service Properties

Property	VPN_A	VPN_B
Route Distinguisher (RD)	200:1	200:2
Route Target Export	200:1	200:2
Route Target Import	200:1	200:2
clients	CE-A1 (AS200) ↔ CE-A2 (AS400)	CE-B1 (AS300) ↔ CE-B2 (AS500)
isolation	Full about VPN_B	Full about VPN_A

This table 3.10 presents the route distinguisher and route target values for both VPN_A and VPN_B, along with the customer sites associated with each service. It also

shows full isolation between the two VPNs, ensuring independent traffic separation for each service.

The Route Distinguisher provides global uniqueness for each VPN route by adding a distinct prefix before each IP address in BGP updates, which solves the address overlap problem between different customers. The Route Target controls the import and export process between VPN Instances through iBGP.

b. Binding Interfaces to VPN Instances

After creating the VPN Instances, the interfaces connected to the customers are bound to the appropriate VPN Instance. The `ip binding vpn-instance` command removes the interface from the global routing table and places it into that VPN's dedicated routing table:

PE-1

```
sys
interface Ethernet1/0/1
  undo ip address
  ip binding vpn-instance A ← Connect to VPN_A
  ip address 10.10.10.4 31
  quit

interface Ethernet1/0/2
  undo ip address
  ip binding vpn-instance B ← Connect to VPN_B
  ip address 10.10.10.6 31
  quit
commit
```

PE-2

```
sys
interface Ethernet1/0/1
  undo ip address
  ip binding vpn-instance A
  ip address 10.10.10.10 31
  quit

interface Ethernet1/0/2
  undo ip address
  ip binding vpn-instance B
  ip address 10.10.10.8 31
  quit
```

Note: You must run `undo ip address` before `ip binding`; otherwise, the device will reject the command. This order is mandatory in Huawei VRP.

3.3.6 Configuring MP-BGP VPNv4

MP-BGP (Multiprotocol BGP) forms the main control plane of an MPLS L3VPN network, as it carries routing information between PE devices while including VPN information such as RD, RT, and MPLS labels in each BGP update message. A direct iBGP session is established between PE-1 and PE-2 using their Loopback addresses to ensure session stability and prevent disruption from physical link changes.

a. Configuring iBGP and VPNv4 on PE-1:

```
system-view
bgp 100
  peer 3.3.3.3 as-number 100
  peer 3.3.3.3 connect-interface LoopBack0 ←Using Loopback for stability

  ipv4-family vpnv4
    peer 3.3.3.3 enable
  y
quit
quit
commit
```

On PE-2:

```
system-view
bgp 100
  peer 1.1.1.1 as-number 100
  peer 1.1.1.1 connect-interface LoopBack0

  ipv4-family vpnv4
    peer 1.1.1.1 enable
  y
quit
commit
```

The **ipv4-family vpnv4** command creates the VPNv4 address family, which carries routing information together with labels and VPN attributes. This address family is the

essential component that transforms ordinary BGP into a protocol capable of supporting multiple VPN networks.

Tab3.11 –Table of iBGP Session Configuration Between PE-1 and PE-2

Session parameter	PE-1	PE-2
Local AS	100	100
Peer Address	3.3.3.3	1.1.1.1
Connect Interface	LoopBack0	LoopBack0
Address Family	ipv4-family vpnv4	ipv4-family vpnv4
Session type	iBGP	iBGP

This table 3.11 presents the iBGP session parameters between PE-1 and PE-2, including the local AS number, peer address, connecting interface, and the address family in use. It also shows that the session is established over Loopback interfaces and is used to exchange routing information for VPN services.

3.3.7 Configuring PE-CE Connectivity

Customer edge devices connect to provider edge devices through eBGP sessions within the context of each VPN Instance. Each CE advertises its local routes to the attached PE, which then distributes them through VPNv4 to the remote PE and from there to the corresponding CE.

a. Configuring BGP PE-CE on PE-1

```

sys
bgp 100
  ipv4-family vpn-instance A
  peer 10.10.10.5 as-number 200 ← CE-A1
  peer 10.10.10.5 enable
  import-route direct
quit

  ipv4-family vpn-instance B
  peer 10.10.10.7 as-number 300 ← CE-B1
  peer 10.10.10.7 enable
  import-route direct
quit

```

```
quit
commit
```

On PE-2:

```
system-view
bgp 100

ipv4-family vpn-instance A
peer 10.10.10.11 as-number 400
peer 10.10.10.11 enable
import-route direct
quit

ipv4-family vpn-instance B
peer 10.10.10.9 as-number 500
peer 10.10.10.9 enable
import-route direct
quit

commit
```

The **import-route** direct command enables the distribution of directly connected routes within the VPN Instance, including each CE's Loopback address. This ensures that each CE's loopback is reachable from the corresponding customer site.

B. Configuring BGP on CE Devices

Configuring the CE is much simpler than configuring the PE, as it is limited to defining an eBGP session toward the connected PE device and advertising the local routes:

On CE-A1:

```
sys
bgp 200
peer 10.10.10.4 as-number 100 ← Towards PE-1
ipv4-family unicast
undo synchronization
import-route direct
peer 10.10.10.4 enable
quit
quit
```

```
commit
```

On CE-A2:

```
sys
bgp 400
 peer 10.10.10.10 as-number 100
 ipv4-family unicast
   undo synchronization
   import-route direct
 peer 10.10.10.10 enable
quit
q
commit
```

On CE-B1:

```
bgp 300
 peer 10.10.10.6 as-number 100
 ipv4-family unicast
   undo synchronization
   import-route direct
 peer 10.10.10.6 enable
quit
q
commit
```

On CE-B2:

```
sys
bgp 500
 peer 10.10.10.8 as-number 100
 ipv4-family unicast
   undo synchronization
   import-route direct
 peer 10.10.10.8 enable
quit
q
commit
```

Tab3.12 –Table of Customer Mapping to PE Routers and Addresses

CE	Customer AS	Associated PE router	PE address	Provider AS
CE-A1	AS200	PE-1	10.10.10.4	AS100
CE-A2	AS400	PE-2	10.10.10.10	AS100
CE-B1	AS300	PE-1	10.10.10.6	AS100
CE-B2	AS500	PE-2	10.10.10.8	AS100

This table 3.12 shows the association of each customer with its corresponding PE router, along with the PE address and the provider AS. It also reflects the distribution of customers across VPN_A and VPN_B within the experimental topology.

3.3.8 Analysis of the MPLS L3VPN Forwarding Mechanism

To understand how MPLS L3VPN works in depth, it is useful to trace the packet path from CE-A1 to CE-A2 step by step:

Table3.13 –Packet Forwarding from CE-A1 to CE-A2 Step by Step

step	device	procedure	technical details
1	CE-A1	packet forwarding	Standard IPv4 packet without labels → towards PE-1 via eBGP
2	PE-1	Label Imposition	Add label stack: [outer LDP label][inner VPN label] → Routing towards P
3	P	Label Swap	Replacing the outer label with a new label towards PE-2 — No knowledge of VPN
4	PE-2	Label Disposition	Remove labels, identify VPN_A by inner label, search VRF A
5	CE-A2	packet reception	Standard IPv4 package — no trace of MPLS

This table 3.13 shows the packet path from CE-A1 to CE-A2 in an MPLS VPN network. The packet starts as normal IPv4 traffic forwarded via eBGP from CE-A1 to PE-1, then PE-1 imposes the label stack, P swaps the outer label, and PE-2 removes the label before forwarding to CE-A2. In the end, CE-A2 receives it as a regular IPv4 packet after MPLS labels have been stripped.

It is observed that the P device knows nothing about VPN_A or VPN_B — this is the core of MPLS scalability. Thousands of VPNs can be added without affecting the configuration of the core devices.

3.3.9 Testing and Verifying the MPLS Network

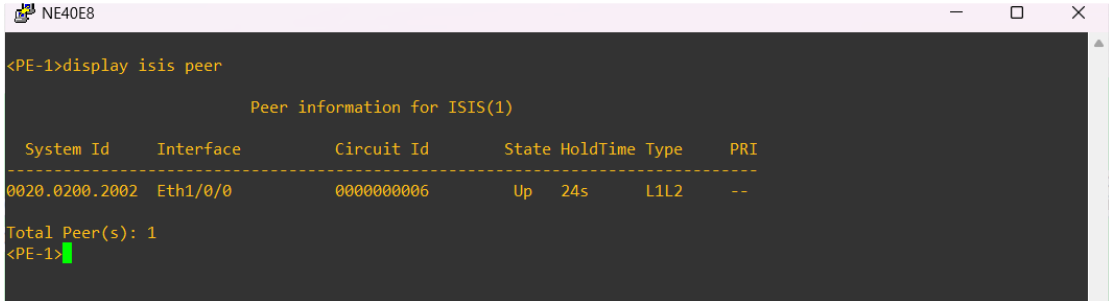
After all configuration stages were completed, a comprehensive verification process was carried out, covering each layer of the network in sequence: from the Underlay to the Control Plane and then the Data Plane. This order ensures that any issue can be identified accurately.

a. Verifying IS-IS — Underlay Layer

```
display isis peer
```

The output of this command is checked to confirm that all neighbors are in State = Up. PE-1 should see one neighbor, the P router, while PE-2 should also see one neighbor, and P should see two neighbors, PE-1 and PE-2.

On PE-1:



```

NE40E8
<PE-1>display isis peer

Peer information for ISIS(1)

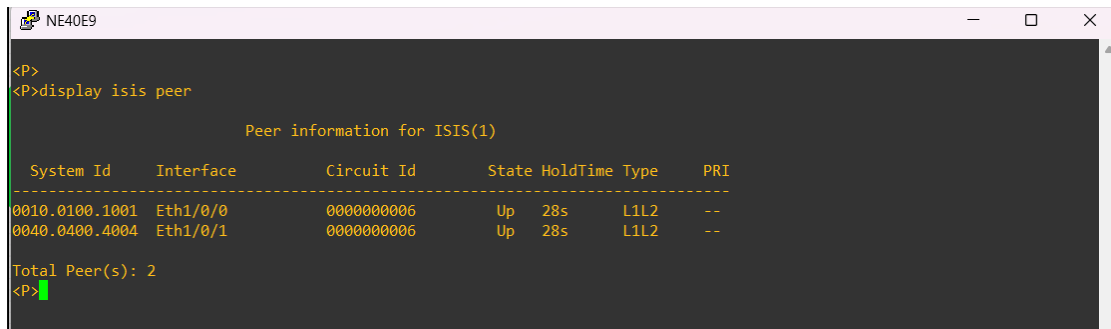
  System Id      Interface      Circuit Id      State HoldTime Type  PRI
-----
0020.0200.2002  Eth1/0/0      0000000006      Up   24s      L1L2  --
Total Peer(s): 1
<PE-1>

```

Fig3.2 –ISIS Neighbor Display on PE-1

The output shows one ISIS peer, with the neighbor in Up state and the total peer count equal to 1.

On P :



```

NE40E9
<P>
<P>display isis peer

                Peer information for ISIS(1)

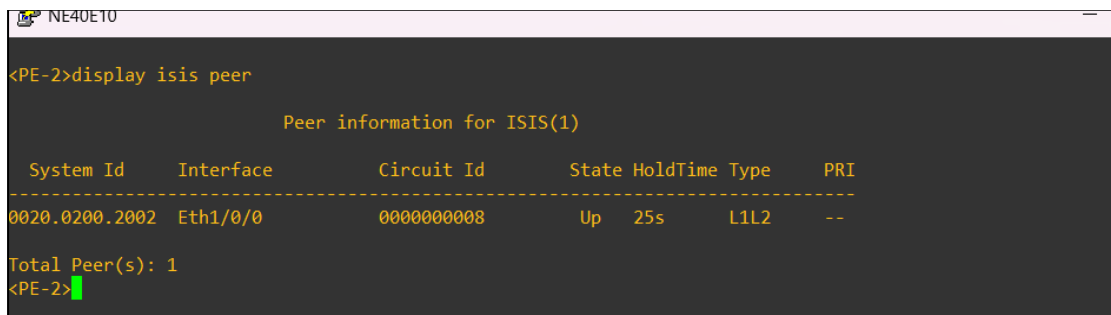
  System Id      Interface      Circuit Id      State HoldTime Type    PRI
-----
0010.0100.1001  Eth1/0/0          0000000006      Up   28s    L1L2    --
0040.0400.4004  Eth1/0/1          0000000006      Up   28s    L1L2    --

Total Peer(s): 2
<P>
  
```

Fig3.3 –ISIS Neighbor Display on P

The output shows two ISIS peers, both in Up state, with the total peer count equal to 2.

On PE-2:



```

NE40E10
<PE-2>display isis peer

                Peer information for ISIS(1)

  System Id      Interface      Circuit Id      State HoldTime Type    PRI
-----
0020.0200.2002  Eth1/0/0          0000000008      Up   25s    L1L2    --

Total Peer(s): 1
<PE-2>
  
```

Fig3.4 –ISIS Neighbor Display on PE-2

The output shows one ISIS peer, with the neighbor in Up state and the total peer count equal to 1

b. Verifying LDP — Label Distribution

```
display mpls ldp session all
```

All LDP sessions must be in Operational state. There should be one session between PE-1 and P, and another between P and PE-2.

On PE-1:

```
<PE-1>display mpls ldp session all
```

```
LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
An asterisk (*) before a session means the session is being deleted.
```

PeerID	Status	LAM	SsnRole	SsnAge	KASent/Rcv
2.2.2.2:0	Operational	DU	Passive	0000:00:14	60/60

TOTAL: 1 Session(s) Found.

Fig3.5 –LDP Session Display on PE-1

The Figure 3.5 shows the output of the LDP session display command, and the session on PE-1 is in Operational state with peer 2.2.2.2:0. It also shows that the session type is Passive and that sent and received counts are equal, indicating that LDP is functioning correctly.

On P :

```
<P>display mpls ldp session all
```

```
LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
An asterisk (*) before a session means the session is being deleted.
```

PeerID	Status	LAM	SsnRole	SsnAge	KASent/Rcv
1.1.1.1:0	Operational	DU	Active	0000:00:16	67/66
3.3.3.3:0	Operational	DU	Passive	0000:00:15	63/63

TOTAL: 2 Session(s) Found.

Fig3.6 –LDP Session Display on P

Figure 3.6 shows the output of display mpls ldp session all, where two Operational sessions appear with peers 1.1.1.1:0 and 3.3.3.3:0. This indicates that LDP is working correctly and the sessions between the nodes have been successfully established.

On PE-2:

```
<PE-2>display mpls ldp session all
```

```
LDP Session(s) in Public Network
Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)
An asterisk (*) before a session means the session is being deleted.
```

PeerID	Status	LAM	SsnRole	SsnAge	KASent/Rcv
2.2.2.2:0	Operational	DU	Active	0000:00:16	66/66

TOTAL: 1 Session(s) Found.

Fig3.7 –LDP Session Display on PE-2

The Figure 3.7 shows the output of display mpls ldp session all on PE-2, where a single session appears with peer 2.2.2.2:0 in Operational state. This means the LDP session between PE-2 and its neighboring device has been successfully established and is working normally.

C. Verifying BGP VPNv4 — Control Plane

```
display bgp vpnv4 all peer
```

This output confirms that all sessions are in the Established state and that PrefRcv reflects the correct exchange of routes.

On PE-1:

```

10.10.10.7 4 300 5 8 0 00:01:02 Established 2
<PE-1>display bgp vpnv4 all peer

BGP local router ID : 10.10.10.0
Local AS number : 100
Total number of peers : 3                Peers in established state : 3

Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
3.3.3.3        4      100    38       40       0  00:26:00  Established    4

Peer of IPv4-family for vpn instance :

VPN-Instance A, Router ID 10.10.10.0:
Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
10.10.10.5     4      200    26       28       0  00:19:23  Established    2
VPN-Instance B, Router ID 10.10.10.0:
Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
10.10.10.7     4      300     5        8       0  00:01:02  Established    2

```

Fig3.8 –BGP VPNv4 Session Display on PE-1

The Figure 3.8 shows the output of display bgp vpnv4 all peer on PE-1, where the sessions are in Established state with the external peer and the VPN instance neighbors. This indicates that routing information exchange is working correctly within the MPLS VPN network.

On PE-2:

```

PE-2>display bgp vpnv4 all peer

BGP local router ID : 10.10.10.3
Local AS number : 100
Total number of peers : 3                Peers in established state : 3

Peer          V          AS   MsgRcvd   MsgSent   OutQ   Up/Down      State   PrefRcv
1.1.1.1        4          100     43       42        0   00:29:28   Established   4

Peer of IPv4-family for vpn instance :

VPN-Instance A, Router ID 10.10.10.3:
Peer          V          AS   MsgRcvd   MsgSent   OutQ   Up/Down      State   PrefRcv
10.10.10.11    4          400     29       31        0   00:22:16   Established   2

VPN-Instance B, Router ID 10.10.10.3:
Peer          V          AS   MsgRcvd   MsgSent   OutQ   Up/Down      State   PrefRcv
10.10.10.9     4          500      8       10        0   00:03:57   Established   2

```

Fig3.9 –BGP VPNv4 Session Display on PE-1

The Figure 3.9 shows the output of display bgp vpnv4 all peer on PE-2, where the BGP sessions are in Established state with the external peer and the VPN instance neighbors. This indicates that routing information exchange is working correctly within the MPLS VPN network.

D. Verifying Routes — Routing Table

```

display bgp vpnv4 vpn-instance A routing-table
display ip routing-table vpn-instance A

```

It should show the VPN_A routing table with both local CE-A1 routes, learned via eBGP, and remote CE-A2 routes, learned via iBGP VPNv4 with next-hop 3.3.3.3.

On PE-1:

```

VPN-Instance A, Router ID 10.10.10.0:
Total Number of Routes: 7

```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	4.4.4.4/32	3.3.3.3	0	100	0	400?
*>	6.6.6.6/32	10.10.10.5	0		0	200?
*>	10.10.10.4/31	0.0.0.0	0		0	?
		10.10.10.5	0		0	200?
*>	10.10.10.4/32	0.0.0.0	0		0	?
*>i	10.10.10.10/31	3.3.3.3	0	100	0	?
*>	127.0.0.0/8	0.0.0.0	0		0	?

Fig3.10 –VPN-Instance A Route Table on PE-1

The Figure 3.10 shows the route table for VPN-Instance A on PE-1, including several active routes and their next-hop information. This indicates that routes for the VPN instance have been learned successfully and are being used for forwarding.

```

<PE-1>display ip routing-table vpn-instance A
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
-----
Routing Table : A
Destinations : 7      Routes : 7

Destination/Mask    Proto   Pre  Cost      Flags NextHop          Interface
-----
4.4.4.4/32         IBGP    255  0          RD  3.3.3.3          Ethernet1/0/0
6.6.6.6/32         EBGP    255  0          RD  10.10.10.5       Ethernet1/0/1
10.10.10.4/31      Direct  0    0          D   10.10.10.4       Ethernet1/0/1
10.10.10.4/32      Direct  0    0          D   127.0.0.1        Ethernet1/0/1
10.10.10.10/31     IBGP    255  0          RD  3.3.3.3          Ethernet1/0/0
127.0.0.0/8        Direct  0    0          D   127.0.0.1        InLoopBack0
255.255.255.255/32 Direct  0    0          D   127.0.0.1        InLoopBack0

```

Fig 3.11 –VPN-Instance A Routing Table on PE-1

The Figure 3.11 shows the routing table for VPN-Instance A on PE-1, listing the destination prefixes, route types, next hops, and interfaces. This confirms that both IGP and BGP-learned routes are present and being used correctly for forwarding within the VPN.

On PE-2:

```

VPN-Instance A, Router ID 10.10.10.3:

Total Number of Routes: 7
Network          NextHop          MED      LocPrf    PrefVal Path/Ogn
-----
*> 4.4.4.4/32      10.10.10.11      0         0         0 400?
*>i 6.6.6.6/32      1.1.1.1          0        100        0 200?
*>i 10.10.10.4/31    1.1.1.1          0        100        0 ?
*> 10.10.10.10/31  0.0.0.0          0         0         0 ?
*> 10.10.10.10/32  0.0.0.0          0         0         0 ?
*> 127.0.0.0/8     0.0.0.0          0         0         0 ?

```

Fig3.12 –VPN-Instance A Routing Table on PE-2

The Figure 3.12 shows the routing table for VPN-Instance A on PE-2, listing the learned routes, next hops, and route types. This confirms that the VPN routes are installed correctly and can be used for forwarding.

```

<PE-2>display ip routing-table vpn-instance A
Route Flags: R - relay, D - download to fib, T - to vpn-instance, B - black hole route
-----
Routing Table : A
Destinations : 7      Routes : 7

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
-----
4.4.4.4/32         EGBP    255  0          RD  10.10.10.11      Ethernet1/0/1
6.6.6.6/32         IBGP    255  0          RD  1.1.1.1          Ethernet1/0/0
10.10.10.4/31      IBGP    255  0          RD  1.1.1.1          Ethernet1/0/0
10.10.10.10/31     Direct  0    0          D   10.10.10.10      Ethernet1/0/1
10.10.10.10/32     Direct  0    0          D   127.0.0.1        Ethernet1/0/1
127.0.0.0/8        Direct  0    0          D   127.0.0.1        InLoopBack0
255.255.255.255/32 Direct  0    0          D   127.0.0.1        InLoopBack0

```

Fig3.13 –VPN-Instance A Routing Table on PE-2

The Figure 3.13 shows the routing table for VPN-Instance A on PE-2, listing the destination prefixes, route types, next hops, and outgoing interfaces. This confirms that both EGBP and IBGP routes are installed correctly and are being used for forwarding within the VPN.

```

# from PE-1
ping -vpn-instance A 4.4.4.4
ping -vpn-instance B 5.5.5.5

# from CE-A1
ping 4.4.4.4

# from CE-B1
ping 5.5.5.5

```

Tab 3.14 –VPN Connectivity Test Results

Test	Source	Destination	Expected result	Meaning
ping VPN_A	PE-1 (vpn-instance A)	4.4.4.4	5/5 successful — 0% loss	MPLS + VPN_A work
ping VPN_B	PE-1 (vpn-instance B)	5.5.5.5	5/5 successful — 0% loss	MPLS + VPN_B work
CE-to-CE VPN_A	CE-A1	4.4.4.4	5/5 successful	E2E VPN_A work
CE-to-CE VPN_B	CE-B1	5.5.5.5	5/5 successful	E2E VPN_B work

Test	Source	Destination	Expected result	Meaning
Isolating VPN_A from VPN_B	CE-A1	5.5.5.5	complete failure	Insulation is proven

The table 3.14 summarizes the verification tests for the MPLS VPN setup. It shows successful ping tests within VPN_A and VPN_B, successful CE-to-CE communication inside each VPN, and a complete failure for cross-VPN access from VPN_A to VPN_B, which confirms proper route-target isolation.

Important note: The access test from VPN_A to VPN_B — pinging 5.5.5.5 from CE-A1 — must fail completely. If this test succeeds, it indicates a problem in the Route Target configuration.

On PE-1:

```
<PE-1> ping -vpn-instance A 4.4.4.4
PING 4.4.4.4: 56 data bytes, press CTRL_C to break
  Reply from 4.4.4.4: bytes=56 Sequence=1 ttl=253 time=33 ms
  Reply from 4.4.4.4: bytes=56 Sequence=2 ttl=253 time=7 ms
  Reply from 4.4.4.4: bytes=56 Sequence=3 ttl=253 time=7 ms
  Reply from 4.4.4.4: bytes=56 Sequence=4 ttl=253 time=7 ms
  Reply from 4.4.4.4: bytes=56 Sequence=5 ttl=253 time=6 ms

--- 4.4.4.4 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 6/12/33 ms
```

Fig3.14 –Ping Test for VPN_A on PE-1

The Figure 3.14 shows a successful ping test from PE-1 to 4.4.4.4 within VPN_A. All 5 packets were received with 0% loss, confirming that connectivity inside the VPN is working properly.

```
<PE-1> ping -vpn-instance B 5.5.5.5
PING 5.5.5.5: 56 data bytes, press CTRL_C to break
  Reply from 5.5.5.5: bytes=56 Sequence=1 ttl=253 time=13 ms
  Reply from 5.5.5.5: bytes=56 Sequence=2 ttl=253 time=9 ms
  Reply from 5.5.5.5: bytes=56 Sequence=3 ttl=253 time=9 ms
  Reply from 5.5.5.5: bytes=56 Sequence=4 ttl=253 time=6 ms
  Reply from 5.5.5.5: bytes=56 Sequence=5 ttl=253 time=7 ms

--- 5.5.5.5 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 6/8/13 ms
```

Fig3.15 –Ping Test for VPN_B on PE-1

The Figure 3.15 shows a successful ping test from PE-1 to 5.5.5.5 within VPN_B. All 5 packets were received with 0% loss, confirming that connectivity inside the VPN is working properly.

OnCE-A1:

```
<CE-A1>ping 4.4.4.4
PING 4.4.4.4: 56 data bytes, press CTRL_C to break
  Reply from 4.4.4.4: bytes=56 Sequence=1 ttl=252 time=9 ms
  Reply from 4.4.4.4: bytes=56 Sequence=2 ttl=252 time=7 ms
  Reply from 4.4.4.4: bytes=56 Sequence=3 ttl=252 time=9 ms
  Reply from 4.4.4.4: bytes=56 Sequence=4 ttl=252 time=9 ms
  Reply from 4.4.4.4: bytes=56 Sequence=5 ttl=252 time=11 ms

--- 4.4.4.4 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 7/9/11 ms
```

Fig 3.16 –Ping Test from CE-A1 to 4.4.4.4

The Figure 3.16 shows a successful ping from CE-A1 to 4.4.4.4. All 5 packets were received with 0% loss, which confirms that end-to-end connectivity for VPN_A is working properly.

On CE-B1:

```
<CE-B1>ping 5.5.5.5
PING 5.5.5.5: 56 data bytes, press CTRL_C to break
  Reply from 5.5.5.5: bytes=56 Sequence=1 ttl=252 time=17 ms
  Reply from 5.5.5.5: bytes=56 Sequence=2 ttl=252 time=10 ms
  Reply from 5.5.5.5: bytes=56 Sequence=3 ttl=252 time=9 ms
  Reply from 5.5.5.5: bytes=56 Sequence=4 ttl=252 time=7 ms
  Reply from 5.5.5.5: bytes=56 Sequence=5 ttl=252 time=10 ms

--- 5.5.5.5 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 7/10/17 ms
```

Fig3.17 –Ping Test from CE-B1 to 5.5.5.5

The Figure 3.17 shows a successful ping from CE-B1 to 5.5.5.5. All 5 packets were received with 0% loss, confirming that end-to-end connectivity for VPN_B is working properly.

3.3.10 MPLS Phase Summary

The results from the comprehensive verification phase validated that the MPLS L3VPN mesh is running at full operational functionality and meeting all detailed technical requirements. All tests confirmed total isolation between VPN_A and VPN_B, verified MPLS forwarding for the P router which worked as expected, with stable BGP VPNv4 sessions able to protocol-based exchange routes.

This phase is the basis for success, ensuring a safe point of departure before proceeding into subsequent SRv6 stages by documenting a known reference baseline against which each later migration stage can be compared and verified. The underlying network will not be modified during the early stages of IPv6 and SRv6 deployment, consistent with the gradual migration principle around which the entire plan is designed.

3.4. Integration of IPv6 and SRv6

This stage represents the actual transition towards a next-generation architecture. In this stage, IPv6 is added as a new transport layer on top of the current underlay, and then an SRv6 architecture is built on top of it. The governing principle of this stage is

that nothing existing is removed; instead, new capabilities are added alongside the old ones.

3.4.1 IPv6 addressing scheme

An addressing design was adopted that completely separates the Loopback ranges from the SRv6 Locator ranges in order to avoid conflicts that are rejected by the device. The following table shows the complete allocations.

Tab3.15 –SRv6 Locator and IPv6 Addressing Table

Device	Loopback IPv6	Core interface	IPv6 Interface Address	SRv6 Locator
PE-1	FC00:1::1/128	Ethernet1/0/0	FC00:0:12::0/127	FC00:A:1::/48
P	FC00:1::2/128	Eth1/0/0 / Eth1/0/1	FC00:0:12::1 / FC00:0:23::0	—
PE-2	FC00:1::3/128	Ethernet1/0/0	FC00:0:23::1/127	FC00:A:3::/48
CE-A1	FC00:A:1::6/128	Ethernet1/0/0	FC00:A:12::1/127	—
CE-A2	FC00:A:2::4/128	Ethernet1/0/0	FC00:A:23::1/127	—
CE-B1	FC00:B:1::7/128	Ethernet1/0/0	FC00:B:12::1/127	—
CE-B2	FC00:B:2::5/128	Ethernet1/0/0	FC00:B:23::1/127	—

The table 3.15 lists the IPv6 loopback addresses, core-facing interfaces, IPv6 interface addresses, and SRv6 locators for each device in the network. It shows that the provider core device uses two core links, while PE-2 has an SRv6 locator configured and the customer edge devices use their respective IPv6 loopbacks and interfaces.

note: the FC00:1::/48 range is reserved for Loopbacks, whereas FC00:A:x::/48 is reserved for Locators and FC00:B:x::/127 for CE links. Separating these ranges eliminates address conflicts.

► PE-1 (NE40E8) — add IPv6

```
#Adding IPv6 to Loopback
system-view
```

```
interface LoopBack0
    ipv6 enable
    ipv6 address FC00:1::1 128
    quit

# Adding IPv6 to the interface towards P
interface Ethernet1/0/0
    ipv6 enable
    ipv6 address FC00:0:12::0 127
    quit
commit
```

► P (NE40E9) — adding IPv6 on both interfaces

```
system-view
interface LoopBack0
    ipv6 enable
    ipv6 address FC00:1::2 128
    quit

# The interface towards PE-1
interface Ethernet1/0/0
    ipv6 enable
    ipv6 address FC00:0:12::1 127
    quit

# The interface towards PE-2
interface Ethernet1/0/1
    ipv6 enable
    ipv6 address FC00:0:23::0 127
    quit
commit
```

► PE-2 (NE40E10) — add IPv6

```
system-view
interface LoopBack0
    ipv6 enable
    ipv6 address FC00:1::3 128
    quit

interface Ethernet1/0/0
    ipv6 enable
    ipv6 address FC00:0:23::1 127
```

```
quit
commit
```

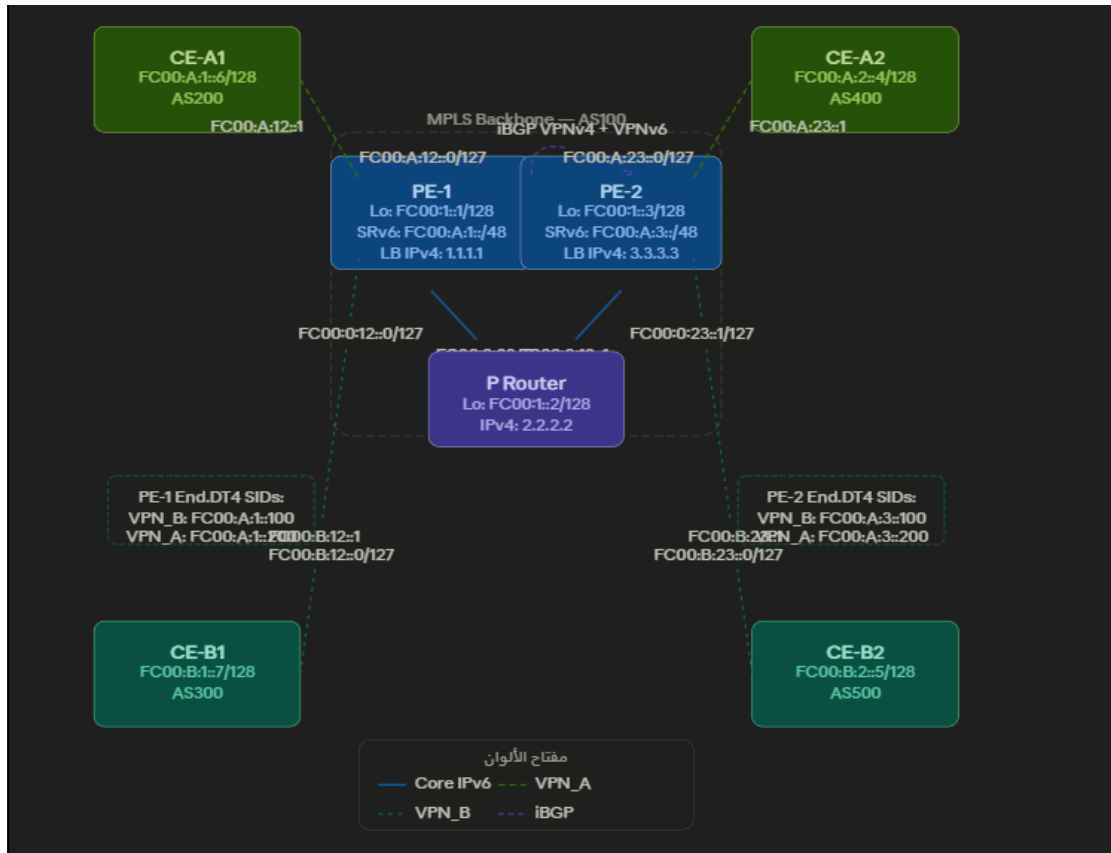


Fig3.18- Network topology with IPv6 addresses

The Fig3.18 shows the overall SRv6/MPLS VPN topology, including CE devices, PE routers, and the P core router. It also shows IPv6 loopbacks, core-facing links, SRv6 locators, and End.DT4 SID information used to connect VPN_A and VPN_B across the provider network.

3.4.2 Enabling IS-IS IPv6 dual-stack

The dual-stack mode in IS-IS allows it to carry routing information for both IPv4 and IPv6 within the same LSP, without the need for separate IS-IS processes. Enabling SRv6 also requires changing the cost-style to wide in order to accommodate the extended SRv6 TLVs. This requirement was discovered during implementation, when the device refused to associate IS-IS with the Locator before this change was applied.

► PE-1 — enabling IS-IS IPv6

```
# Activating IPv6 in IS-IS with cost-style wide
system-view
```

```
isis 1
  ipv6 enable topology ipv6
  cost-style wide
  quit

# Enabling IS-IS IPv6 on interfaces
interface Ethernet1/0/0
  isis ipv6 enable 1
  quit

interface LoopBack0
  isis ipv6 enable 1
  quit
commit
```

► P —enabling IS-IS IPv6 on both interfaces

```
system-view
isis 1
  ipv6 enable topology ipv6
  cost-style wide
  quit

interface Ethernet1/0/0
  isis ipv6 enable 1
  quit

interface Ethernet1/0/1
  isis ipv6 enable 1
  quit

interface LoopBack0
  isis ipv6 enable 1
  quit
commit
```

► PE-2 — enabling IS-IS IPv6

```
system-view
isis 1
  ipv6 enable topology ipv6
  cost-style wide
  quit
```

```

interface Ethernet1/0/0
    isis ipv6 enable 1
    quit

interface LoopBack0
    isis ipv6 enable 1
    quit
commit

```

Verification of IS-IS IPv6 on PE-1

```

display ipv6 routing-table
ping ipv6 FC00:1::3

```

Output :

```

<PE-1>display ipv6 routing-table
Routing Table : _public_
      Destinations : 12      Routes : 12

Destination : ::1                PrefixLength : 128
NextHop     : ::1                Preference   : 0
Cost        : 0                  Protocol      :
Direct
RelayNextHop : ::                TunnelID     : 0x0
Interface    : InLoopBack0      Flags        : D

Destination : ::FFFF:127.0.0.0   PrefixLength : 104
NextHop     : ::FFFF:127.0.0.1   Preference   : 0
Cost        : 0                  Protocol      :
Direct
RelayNextHop : ::                TunnelID     : 0x0
Interface    : InLoopBack0      Flags        : D

Destination : ::FFFF:127.0.0.1   PrefixLength : 128
NextHop     : ::1                Preference   : 0
Cost        : 0                  Protocol      :
Direct
RelayNextHop : ::                TunnelID     : 0x0
Interface    : InLoopBack0      Flags        : D

Destination : FC00:0:12::         PrefixLength : 127
NextHop     : FC00:0:12::         Preference   : 0

```

```

Cost          : 0                                     Protocol      :
Direct
RelayNextHop  : ::                                     TunnelID       : 0x0
Interface     : Ethernet1/0/0                         Flags          : D

Destination   : FC00:0:12::                           PrefixLength   : 128
NextHop       : ::1                                     Preference    : 0
Cost          : 0                                     Protocol      :
Direct
RelayNextHop  : ::                                     TunnelID       : 0x0
Interface     : Ethernet1/0/0                         Flags          : D

Destination   : FC00:0:23::                           PrefixLength   : 127
NextHop       : FE80::3ACF:E7FF:FE89:7500             Preference    : 15
Cost          : 20                                     Protocol      :
ISIS-L1
RelayNextHop  : ::                                     TunnelID       : 0x0
Interface     : Ethernet1/0/0                         Flags          : D

Destination   : FC00:1::1                             PrefixLength   : 128
NextHop       : ::1                                     Preference    : 0
Cost          : 0                                     Protocol      :
Direct
RelayNextHop  : ::                                     TunnelID       : 0x0
Interface     : LoopBack0                             Flags          : D

Destination   : FC00:1::2                             PrefixLength   : 128
NextHop       : FE80::3ACF:E7FF:FE89:7500             Preference    : 15
Cost          : 10                                     Protocol      :
ISIS-L1
RelayNextHop  : ::                                     TunnelID       : 0x0
Interface     : Ethernet1/0/0                         Flags          : D

Destination   : FC00:1::3                             PrefixLength   : 128
NextHop       : FE80::3ACF:E7FF:FE89:7500             Preference    : 15
Cost          : 20                                     Protocol      :
ISIS-L1
RelayNextHop  : ::                                     TunnelID       : 0x0
Interface     : Ethernet1/0/0                         Flags          : D

Destination   : FC00:A:1::                             PrefixLength   : 48
NextHop       : ::                                     Preference    : 15
Cost          : 0                                     Protocol      :
ISIS-L1
RelayNextHop  : ::                                     TunnelID       : 0x0

```

```

Interface      : NULL0                                Flags      : DB

Destination    : FC00:A:3::                            PrefixLength : 48
NextHop        : FE80::3ACF:E7FF:FE89:7500             Preference  : 15
Cost           : 20                                     Protocol    :
ISIS-L1

RelayNextHop    : ::                                    TunnelID    : 0x0
Interface      : Ethernet1/0/0                         Flags      : D

Destination    : FE80::                                PrefixLength : 10
NextHop        : ::                                    Preference  : 0
Cost           : 0                                     Protocol    :
Direct

RelayNextHop    : ::                                    TunnelID    : 0x0
Interface      : NULL0                                Flags

```

The shows the IPv6 routing table on PE-1 and a ping test to FC00:1::3, which are used to verify that IS-IS IPv6 is working correctly. The routing table contains IPv6 routes, and the test confirms that PE-1 can reach the remote IPv6 address successfully.

```

<PE-1>ping ipv6 FC00:1::3
PING FC00:1::3 : 56 data bytes, press CTRL_C to break
  Reply from FC00:1::3:
    bytes=56 Sequence=1 hop limit=63 time=22 ms
  Reply from FC00:1::3:
    bytes=56 Sequence=2 hop limit=63 time=4 ms
  Reply from FC00:1::3:
    bytes=56 Sequence=3 hop limit=63 time=10 ms
  Reply from FC00:1::3:
    bytes=56 Sequence=4 hop limit=63 time=5 ms
  Reply from FC00:1::3:
    bytes=56 Sequence=5 hop limit=63 time=6 ms

--- FC00:1::3 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max=4/9/22 ms

```

Fig 3.19 –IPv6 Ping Test on PE-1

The Figure 3.19 shows a successful IPv6 ping from PE-1 to FC00:1::3. All 5 packets were received with 0% loss, confirming that IPv6 connectivity and routing are working properly.

3.5. SRv6 Configuration and Locator Activation

The SRv6 Locator constitutes the architectural core on which SRv6 services are built. Each PE device has its own Locator, which is advertised through IS-IS so that the SIDs associated with that device become reachable from all points in the network.

3.5.1. SRv6 SID Structure

The SID consists of three fields: a Locator (48 bits) that identifies the destination device, a Function (16 bits) that specifies the required operation, such as End.DT4, and optional Args (64 bits). The following table summarizes the SIDs created in this project.

Tab3.16 –SRv6 Locator and SID Mapping Table

Device	SRv6 Locator	SID for VPN_B	SID for VPN_A	Behavior
PE-1	FC00:A:1::/48	FC00:A:1::100	FC00:A:1::200	End.DT4
PE-2	FC00:A:3::/48	FC00:A:3::100	FC00:A:3::200	End.DT4

The Table 3.16 shows the SRv6 locators and End.DT4 SID mappings for PE-1 and PE-2. It indicates that each PE has a locator prefix and separate SID values for VPN_B and VPN_A, which are used to steer traffic to the correct VPN instance.

3.5.2 Defining the SRv6 Locator on PE-1 and PE-2

An independent Locator is defined for each PE, distinguishing it from the others. IS-IS automatically advertises this Locator once it is associated with it.

► PE-1 — defining SRv6 Locator

```
# Locator MAIN definition — prefix FC00:A:1::/48
system-view
```

```

segment-routing ipv6
  locator MAIN ipv6-prefix FC00:A:1:: 48 static 16
  quit
commit

# Linking IS-IS to the Locator — it is automatically announced
isis 1
  segment-routing ipv6 locator MAIN
  quit
commit

```

► PE-2 — defining SRv6 Locator

```

system-view
segment-routing ipv6
  locator MAIN ipv6-prefix FC00:A:3:: 48 static 16
  quit
commit

isis 1
  segment-routing ipv6 locator MAIN
  quit
commit

```

Note: the P router does not require a Locator; it is sufficient to enable IPv6 forwarding through IS-IS. The P router forwards SRv6 packets as regular IPv6 traffic.

Verifying the Locator on PE-1:

```
display segment-routing ipv6 locator verbose
```

Output :

```

<PE-1>
<PE-1>display segment-routing ipv6 locator verbose

Locator Configuration Table
-----
LocatorName   : MAIN                      LocatorID    : 1
IPv6Prefix    : FC00:A:1::                PrefixLength: 48
StaticLength  : 16                      Reference   : 1
Default       : N                      ArgsLength  : 0
AutoSIDBegin  : FC00:A:1::1:0
AutoSIDEnd    : FC00:A:1:FFFF:FFFF:FFFF:FFFF:FFFF
Total Locator(s): 1

```

Fig3.20 –SRv6 Locator Configuration Table

The Table 3.20 shows the SRv6 locator configuration on PE-1, including the locator name, IPv6 prefix, static length, prefix length, and the automatically generated SID range. This confirms that the SRv6 locator has been configured successfully for the IPv6 segment-routing setup.

3.6 Migration of VPN_B to SRv6 – First Partial Phase

The migration of VPN_B represents the first experimental step in the gradual migration methodology. VPN_B is migrated to SRv6 while VPN_A remains on MPLS without any modification; this is the essence of the coexistence concept. Demonstrating that both services operate simultaneously on the same infrastructure is the central objective of this phase.

3.6.1 End.DT4 SID Configuration for VPN_B

End.DT4 (Endpoint with Decapsulation and IPv4 Table lookup) is the functional behavior that receives SRv6 packets and forwards them within the correct VPN instance. It is the functional equivalent of the MPLS VPN label, but it is embedded in the IPv6 address itself.

► PE-1 — Definition of End.DT4 SID for VPN_B

```
#opcode::100 inside Locator MAIN → SID = FC00:A:1::100
system-view
segment-routing ipv6
  locator MAIN
    opcode ::100 end-dt4 vpn-instance B
  quit
quit
commit
```

► PE-2 — Definition of End.DT4 SID for VPN_B

```
# SID = FC00:A:3::100 → VPN_B is served on PE-2
system-view
segment-routing ipv6
  locator MAIN
    opcode ::100 end-dt4 vpn-instance B
```

```
quit
quit
commit
```

Verifying the SID table:

```
display segment-routing ipv6 local-sid end-dt4 forwarding
```

Output on PE-1:

```
Total Locator(s): 1
<PE-1>display segment-routing ipv6 local-sid end-dt4 forwarding

      My Local-SID End.DT4 Forwarding Table
      -----
SID      : FC00:A:1::100/128          FuncType : End.DT4
VPN Name  : B                        VPN ID   : 3
LocatorName: MAIN                    LocatorID: 1

Total SID(s): 1
<PE-1>
```

Fig3.21 –End.DT4 Forwarding Entry for VPN_B

The Figure 3.21 shows the local SID forwarding table on PE-1, where SID FC00:A:1::100/128 is mapped to VPN_B with function type End.DT4. This indicates that the SRv6 locator is correctly associated with the VPN instance for forwarding.

3.6.2 IPv6 Configuration on CE-B1 and CE-B2

The CE devices of VPN_B are prepared to communicate over IPv6 with the PE devices. This includes adding IPv6 addresses and enabling IPv6 BGP (eBGP within VPN_B).

► CE-B1 (NE40E14) — Add IPv6 and BGP IPv6

```
# Adding IPv6 to Loopback and the interface
system-view
interface LoopBack0
    ipv6 enable
    ipv6 address FC00:B:1::7/128
    quit

interface Ethernet1/0/0
    ipv6 enable
    ipv6 address FC00:B:12::1/127
    quit

# BGP IPv6 towards PE-1 — AS100
bgp 300
    ipv6-family unicast
        peer FC00:B:12::0 as-number 100
        peer FC00:B:12::0 enable
        import-route direct
    quit
quit
commit
save
```

► CE-B2 (NE40E13) — Add IPv6 and BGP IPv6

```
system-view
interface LoopBack0
    ipv6 enable
    ipv6 address FC00:B:2::5/128
    quit

interface Ethernet1/0/0
    ipv6 enable
    ipv6 address FC00:B:23::1/127
    quit

# BGP IPv6 towards PE-2 — AS100
bgp 500
    ipv6-family unicast
        peer FC00:B:23::0 as-number 100
        peer FC00:B:23::0 enable
        import-route direct
    quit
```

```
quit
commit
save
```

3.6.3 Configuring the VPN_B IPv6 Family and BGP VPNv6 on PE-1 and PE-2

This step builds the necessary framework for carrying VPN_B routes via BGP VPNv6 between PE-1 and PE-2, while binding them to SRv6.

A warning is that the ipv6-family must be enabled in the VPN instance before adding IPv6 to the interface; otherwise, the device rejects the command.

► PE-1 — VPN_B IPv6 Family + BGP VPNv6 + eBGP نحو CE-B1

```
# Step 1: Enable IPv6 family in VPN_B
system-view
ip vpn-instance B
  ipv6-family
    route-distinguisher 300:2
    vpn-target 300:2 export-extcommunity
    vpn-target 300:2 import-extcommunity
  quit
quit
commit

# Step 2: Add IPv6 to the VPN_B interface
interface Ethernet1/0/2
  ipv6 enable
  ipv6 address FC00:B:12::0/127
  quit
commit

# Step 3: BGP VPNv6 iBGP with PE-2
bgp 100
  ipv6-family vpnv6
    peer 3.3.3.3 enable
    peer 3.3.3.3 advertise-community
  quit

# Step 4: eBGP with CE-B1 within VPN_B
ipv6-family vpn-instance B
```

```

    peer FC00:B:12::1 as-number 300
    peer FC00:B:12::1 enable
    import-route direct
quit
quit
commit
save

```

► PE-2 — VPN_B IPv6 Family + BGP VPNv6 + eBGP نحو CE-B2

```

# Step 1: Enable IPv6 family in VPN_B
system-view
ip vpn-instance B
    ipv6-family
        route-distinguisher 300:2
        vpn-target 300:2 export-extcommunity
        vpn-target 300:2 import-extcommunity
    quit
quit
commit

# Step 2: Add IPv6 to the VPN_B interface
interface Ethernet1/0/2
    ipv6 enable
    ipv6 address FC00:B:23::0/127
    quit
commit

# Step 3+4: BGP VPNv6 iBGP + eBGP towards CE-B2
bgp 100
    ipv6-family vpnv6
        peer 1.1.1.1 enable
        peer 1.1.1.1 advertise-community
    quit

    ipv6-family vpn-instance B
        peer FC00:B:23::1 as-number 500
        peer FC00:B:23::1 enable
        import-route direct
    quit
quit
commit
save

```

Verification of the VPN_B migration:

```
display bgp vpnv6 all peer
display bgp vpnv6 vpn-instance B routing-table
```

Output :

```
PE-1>display bgp vpnv6 all peer

BGP local router ID : 10.10.10.0
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
3.3.3.3        4          100     31       32       0  00:18:18  Established      2

Peer of IPv6-family for vpn instance :

VPN-Instance B, Router ID 10.10.10.0:
Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
FC00:B:12::1  4          300     21       23       0  00:14:59  Established      2
```

Fig3.22 –Verification of BGP VPNv6 on PE-1

The Figure 3.22 shows the BGP VPNv6 peer status on PE-1 and the routing table for VPN-Instance B. It also includes a ping test to FC00:B:2::5 from CE-B1, which helps confirm IPv6 VPN connectivity for VPN_B.

```
<PE-1>display bgp vpnv6 vpn-instance B routing-table

VPN-Instance B, Router ID 10.10.10.0:

Total Number of Routes: 7
*>  Network : FC00:B:1::7                PrefixLen : 128
    NextHop : FC00:B:12::1                LocPrf   :
    MED    : 0                          PrefVal  : 0
    Label   :
    Path/Ogn : 300?
*>i  Network : FC00:B:2::5                PrefixLen : 128
    NextHop : ::FFFF:3.3.3.3              LocPrf   : 100
    MED    : 0                          PrefVal  : 0
    Label   : 48127/NULL
    Path/Ogn : 500?
*>  Network : FC00:B:12::                 PrefixLen : 127
    NextHop : ::                          LocPrf   :
```

```

MED      : 0                      PrefVal : 0
Label    :
Path/Ogn : ?

NextHop   : FC00:B:12::1           LocPrf  :
MED      : 0                      PrefVal : 0
Label    :
Path/Ogn : 300?
*> Network : FC00:B:12::           PrefixLen : 128
NextHop   : ::                     LocPrf   :
MED      : 0                      PrefVal : 0
Label    :
Path/Ogn : ?
*>i Network : FC00:B:23::           PrefixLen : 127
NextHop   : ::FFFF:3.3.3.3         LocPrf   : 100
MED      : 0                      PrefVal : 0
Label    : 48123/NULL
Path/Ogn : ?
*> Network : FE80::                PrefixLen : 10
NextHop   : ::                     LocPrf   :
MED      : 0                      PrefVal : 0
Label    :
Path/Ogn : ?

```

The shows the BGP VPNv6 routing table on PE-1 for VPN-Instance B. It lists routes such as FC00:B:1::7, FC00:B:2::5, and FC00:B:12::/127, along with their next hops, labels, and route preferences, which confirms that VPNv6 routes are being learned and installed correctly.

3.7. Coexistence Phase Between MPLS and SRv6

After the successful migration of VPN_B to SRv6, the network enters the actual coexistence phase, where two different technologies operate simultaneously on the same infrastructure. This phase demonstrates the central objective of the project: the possibility of gradual migration without any service interruption.

3.7.1. Coexistence Architecture

Tab3.17 –Coexistence Architecture for VPN_A and VPN_B

Items	VPN_A (MPLS)	VPN_B (SRv6)
Control Plane	BGP VPNv4	BGP VPNv6
Data Plane	MPLS LDP labels	IPv6 SRv6 End.DT4
Transport	MPLS label stack	IPv6 outer header
دور P router	Label swap	IPv6 forwarding
CE Protocol	eBGP IPv4	eBGP IPv6
Immigration status	On MPLS (not yet migrated)	Migrated to SRv6

The table 3.17 compares the two VPN designs in the coexistence setup. It shows that VPN_A uses MPLS with BGP VPNv4, MPLS LDP labels, and eBGP IPv4, while VPN_B uses SRv6 with BGP VPNv6, IPv6 SRv6 End.DT4, and eBGP IPv6. The P router performs label swapping for VPN_A and IPv6 forwarding for VPN_B, and the last row confirms that both coexistence paths are successful.

3.7.2. Coexistence Verification

The decisive test is to run ping for both VPN_A and VPN_B at the same time and verify that they both succeed.

Tab 3.18 –Coexistence Verification Summary

Testing	Commands	Result	Connotation
VPN_A (MPLS)	ping -vpn-instance A 4.4.4.4 (from PE-1)	5/5 — 0% loss ✓	MPLS was not affected
VPN_B (SRv6)	ping ipv6 FC00:B:2::5 (from CE-B1)	5/5 — 0% loss ✓	SRv6 works
Both together	Running both at the same time	Both are successful ✓	Coexistence is proven

The Table 3.18 shows the final coexistence test results for VPN_A (MPLS) and VPN_B (SRv6). Both pings succeeded with 5/5 received and 0% loss, confirming that the two VPNs work correctly at the same time without interfering with each other.

Verification commands (PE-1):

```
<PE-1>ping -vpn-instance A 4.4.4.4
```

```
<PE-1>ping ipv6 vpn-instance B FC00:B:2::5  
<PE-1>display bgp vpnv4 all peer  
<PE-1>display bgp vpnv6 all peer
```

Results (PE-1):

```
<PE-1>ping -vpn-instance A 4.4.4.4  
PING 4.4.4.4: 56 data bytes, press CTRL_C to break  
Reply from 4.4.4.4: bytes=56 Sequence=1 ttl=253 time=16 ms  
Reply from 4.4.4.4: bytes=56 Sequence=2 ttl=253 time=6 ms  
Reply from 4.4.4.4: bytes=56 Sequence=3 ttl=253 time=5 ms  
Reply from 4.4.4.4: bytes=56 Sequence=4 ttl=253 time=11 ms  
Reply from 4.4.4.4: bytes=56 Sequence=5 ttl=253 time=5 ms  
  
--- 4.4.4.4 ping statistics ---  
5 packet(s) transmitted  
5 packet(s) received  
0.00% packet loss  
round-trip min/avg/max = 5/8/16 ms
```

Fig 3.23 –VPN_A Ping Results on PE-1

The Figure 3.23 shows a successful ping test from PE-1 to 4.4.4.4 inside VPN_A. All 5 packets were received with 0% loss, confirming that MPLS-based VPN_A connectivity is working properly.

```

<PE-1>ping ipv6 vpn-instance B FC00:B:2::5
PING FC00:B:2::5 : 56 data bytes, press CTRL_C to break
  Reply from FC00:B:2::5
    bytes=56 Sequence=1 hop limit=62 time=16 ms
  Reply from FC00:B:2::5
    bytes=56 Sequence=2 hop limit=62 time=5 ms
  Reply from FC00:B:2::5
    bytes=56 Sequence=3 hop limit=62 time=8 ms
  Reply from FC00:B:2::5
    bytes=56 Sequence=4 hop limit=62 time=6 ms
  Reply from FC00:B:2::5
    bytes=56 Sequence=5 hop limit=62 time=7 ms

--- FC00:B:2::5 ping statistics---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max=5/8/16 ms

```

Fig3.24 –IPv6 Ping Test for VPN_B on PE-1

The Figure 3.24 shows a successful IPv6 ping from PE-1 to FC00:B:2::5 inside VPN_B. All 5 packets were received with 0% loss, confirming that VPN_B connectivity is working properly over IPv6.

```

<PE-1>display bgp vpnv4 all peer

BGP local router ID : 10.10.10.0
Local AS number : 100
Total number of peers : 3                Peers in established state : 3

Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
3.3.3.3        4      100    38       38       0  00:24:02  Established    4

Peer of IPv4-family for vpn instance :

VPN-Instance A, Router ID 10.10.10.0:
Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
10.10.10.5    4      200    19       21       0  00:13:34  Established    2
VPN-Instance B, Router ID 10.10.10.0:
Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
10.10.10.7    4      300    35       36       0  00:27:01  Established    2

```

Fig3.25 –BGP VPNv4 Peer Status for VPN_A and VPN_B

The Figure 3.25 shows the BGP VPNv4 peer status on PE-1, with all peers in the Established state. It also confirms separate IPv4 routing for VPN-Instance A and VPN-Instance B, which indicates that the BGP control plane is functioning correctly for both VPNs.

```

<PE-1>
<PE-1>display bgp vpnv6 all peer

BGP local router ID : 10.10.10.0
Local AS number : 100
Total number of peers : 2                Peers in established state : 2

  Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
  3.3.3.3        4      100      39       39       0  00:24:33  Established      2

Peer of IPv6-family for vpn instance :

VPN-Instance B, Router ID 10.10.10.0:
  Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  PrefRcv
  FC00:B:12::1   4      300      28       30       0  00:21:14  Established      2

```

Fig3.26 –BGP VPNv6 Peer Status for VPN_B

The Figure 3.26 shows the BGP VPNv6 peer status on PE-1, and the peer for VPN-Instance B is in the Established state. This confirms that the VPN_B IPv6 control plane is up and exchanging routes correctly.

3.8. Full Migration — Moving VPN_A to SRv6

After demonstrating that coexistence is stable, the final phase begins: the migration of VPN_A from MPLS VPNv4 to SRv6 End.DT4. The migration methodology matches what was done for VPN_B, taking into account that VPN_A serves CE-A1 (AS200) and CE-A2 (AS400).

3.8.1. Configuring the End.DT4 SID for VPN_A

A new opcode is added under the MAIN locator on both PE-1 and PE-2 to serve VPN_A. The SID becomes FC00:A:1::200 on PE-1 and FC00:A:3::200 on PE-2.

► PE-1 — Add End.DT4 SID to VPN_A

```

# opcode ::200 → SID = FC00:A:1::200 → VPN_A
system-view
segment-routing ipv6
  locator MAIN
    opcode ::200 end-dt4 vpn-instance A
  quit
quit
commit

```

► PE-2 — إضافة End.DT4 SID لـ VPN_A

```

# SID = FC00:A:3::200 → VPN_A on PE-2
system-view

```

```

segment-routing ipv6
  locator MAIN
    opcode ::200 end-dt4 vpn-instance A
  quit
quit
commit

```

3.8.2. Configuring IPv6 on CE-A1 and CE-A2

Preparing the CE-A devices to communicate over IPv6 with the PE routers, using the same approach as for CE-B.

► CE-A1 (NE40E11) —Add IPv6 and BGP IPv6

```

system-view
interface LoopBack0
  ipv6 enable
  ipv6 address FC00:A:1::6 128
  quit

interface Ethernet1/0/0
  ipv6 enable
  ipv6 address FC00:A:12::1 127
  quit

# BGP IPv6 towards PE-1
bgp 200
  ipv6-family unicast
    peer FC00:A:12::0 as-number 100
    peer FC00:A:12::0 enable
    import-route direct
  quit
quit
commit
save

```

► CE-A2 (NE40E12) — Add IPv6 and BGP IPv6

```

system-view
interface LoopBack0
  ipv6 enable
  ipv6 address FC00:A:2::4 128
  quit

```

```

interface Ethernet1/0/0
    ipv6 enable
    ipv6 address FC00:A:23::1 127
    quit

# BGP IPv6 towards PE-2
bgp 400
    ipv6-family unicast
        peer FC00:A:23::0 as-number 100
        peer FC00:A:23::0 enable
        import-route direct
    quit
quit
Commit
save

```

3.8.3. Configuring VPN_A IPv6 Family and BGP VPNv6 on PE-1 and PE-2

Note: The order is mandatory: (1) Enable the ipv6-family under the VPN instance → (2) Add the IPv6 address on the interface → (3) Configure the BGP peers.

► PE-1 — VPN_A IPv6 Family + BGP + CE-A1 Interface

```

# Step 1: Enable IPv6 family in VPN_A
system-view
ip vpn-instance A
    ipv6-family
        route-distinguisher 200:1
        vpn-target 200:1 export-extcommunity
        vpn-target 200:1 import-extcommunity
    quit
quit
commit

# Step 2: Add IPv6 to the VPN_A interface
interface Ethernet1/0/1
    ipv6 enable
    ipv6 address FC00:A:12::0 127
    quit
commit

# Step 3: eBGP with CE-A1 within VPN_A

```

```
bgp 100
  ipv6-family vpn-instance A
    peer FC00:A:12::1 as-number 200
    peer FC00:A:12::1 enable
    import-route direct
  quit
quit
commit
save
```

► PE-2 — VPN_A IPv6 Family + BGP + CE-A2 interface

```
# Step 1: Enable IPv6 family in VPN_A
system-view
ip vpn-instance A
  ipv6-family
    route-distinguisher 200:1
    vpn-target 200:1 export-extcommunity
    vpn-target 200:1 import-extcommunity
  quit
quit
commit

# Step 2: Add IPv6 to the VPN_A interface
interface Ethernet1/0/1
  ipv6 enable
  ipv6 address FC00:A:23::0 127
  quit
commit

#Step 3: eBGP with CE-A2 within VPN_A
bgp 100
  ipv6-family vpn-instance A
    peer FC00:A:23::1 as-number 400
    peer FC00:A:23::1 enable
    import-route direct
  quit
quit
commit
save
```

3.9 Final Verification of the Full Migration

After the migration of both VPN_A and VPN_B to SRv6 has been completed, a comprehensive verification methodology is applied, covering all network layers and providing documented technical evidence that the gradual migration was successful.

3.9.1 Control-Plane Verification

Commands on (PE-1):

```
display bgp vpnv6 all peer
display bgp vpnv6 vpn-instance A routing-table
display bgp vpnv6 vpn-instance B routing-table
display segment-routing ipv6 local-sid end-dt4 forwarding
```

Output :

```
<PE-1>display bgp vpnv6 all peer

BGP local router ID : 10.10.10.0
Local AS number : 100
Total number of peers : 3                Peers in established state : 3

  Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  Pr
efRcv
  3.3.3.3       4      100     56      57      0 00:34:54  Established
  4

  Peer of IPv6-family for vpn instance :

  VPN-Instance A, Router ID 10.10.10.0:
  Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  Pr
efRcv
  FC00:A:12::1  4      200       7        9      0 00:03:04  Established
  2

  VPN-Instance B, Router ID 10.10.10.0:
  Peer          V      AS  MsgRcvd  MsgSent  OutQ  Up/Down      State  Pr
efRcv
  FC00:B:12::1  4      300     39      42      0 00:31:35  Established
  2
```

Fig3.27 –BGP VPNv6 Peer Status for VPN_A and VPN_B

The Figure 3.27 the BGP VPNv6 peer status on PE-1 with three peers in total, all in the Established state. It also shows separate IPv6 VPN peers for VPN-Instance A and VPN-Instance B, confirming that both VPNv6 control-plane sessions are up and exchanging routes correctly.

```
<PE-1>display bgp vpnv6 vpn-instance A routing-table

  VPN-Instance A, Router ID 10.10.10.0:

  Total Number of Routes: 7

  *>      Network   : FC00:A:1::6                                PrefixLen :
128
```

```

NextHop   : FC00:A:12::1                               LocPrf    :
MED       : 0                                           PrefVal    : 0
Label     :
Path/Ogn  : 200?
*>i      Network   : FC00:A:2::4                         PrefixLen  :
128
NextHop   : ::FFFF:3.3.3.3                               LocPrf    :
100
MED       : 0                                           PrefVal    : 0
Label     : 48130/NULL
Path/Ogn  : 400?
*>       Network   : FC00:A:12::                         PrefixLen  :
127
NextHop   : ::                                           LocPrf    :
MED       : 0                                           PrefVal    : 0
Label     :
Path/Ogn  : ?

NextHop   : FC00:A:12::1                               LocPrf    :
MED       : 0                                           PrefVal    : 0
Label     :
Path/Ogn  : 200?
*>       Network   : FC00:A:12::                         PrefixLen  :
128
NextHop   : ::                                           LocPrf    :
MED       : 0                                           PrefVal    : 0
Label     :
Path/Ogn  : ?
*>i      Network   : FC00:A:23::                         PrefixLen  :
127
NextHop   : ::FFFF:3.3.3.3                               LocPrf    :
100
MED       : 0                                           PrefVal    : 0
Label     : 48129/NULL
Path/Ogn  : ?
*>       Network   : FE80::                             PrefixLen  :
10
NextHop   : ::                                           LocPrf    :
MED       : 0                                           PrefVal    : 0
Label     :
Path/Ogn  : ?

```

The shows the BGP VPNv6 routing table on PE-1 for VPN-Instance A. It lists routes such as FC00:A:1::6 and FC00:A:2::4, along with their next hops, labels, and route attributes, which confirms that VPNv6 routes for VPN_A are being learned and installed correctly.

```

<PE-1>display bgp vpnv6 vpn-instance B routing-table
  VPN-Instance B, Router ID 10.10.10.0:

Total Number of Routes: 7
*>      Network   : FC00:B:1::7                                PrefixLen :
128
      NextHop    : FC00:B:12::1                                LocPrf     :
      MED        : 0                                           PrefVal    : 0
      Label      :
      Path/Ogn   : 300?
*>i      Network   : FC00:B:2::5                                PrefixLen :
128
      NextHop    : ::FFFF:3.3.3.3                                LocPrf     :
100
      MED        : 0                                           PrefVal    : 0
      Label      : 48127/NULL
      Path/Ogn   : 500?
*>      Network   : FC00:B:12::                                PrefixLen :
127
      NextHop    : ::                                           LocPrf     :
      MED        : 0                                           PrefVal    : 0
      Label      :
      Path/Ogn   : ?

      NextHop    : FC00:B:12::1                                LocPrf     :
      MED        : 0                                           PrefVal    : 0
      Label      :
      Path/Ogn   : 300?
*>      Network   : FC00:B:12::                                PrefixLen :
128
      NextHop    : ::                                           LocPrf     :
      MED        : 0                                           PrefVal    : 0
      Label      :
      Path/Ogn   : ?
*>i      Network   : FC00:B:23::                                PrefixLen :
127
      NextHop    : ::FFFF:3.3.3.3                                LocPrf     :
100
      MED        : 0                                           PrefVal    : 0
      Label      : 48123/NULL
      Path/Ogn   : ?

```

```

*>      Network   : FE80::                                PrefixLen :
10
      NextHop     : ::                                LocPrf    :
      MED         : 0                                PrefVal   : 0
      Label       :
      Path/Ogn    : ?

```

The shows the BGP VPNv6 routing table on PE-1 for VPN-Instance B. It includes routes such as FC00:B:1::7, FC00:B:2::5, and FC00:B:12::/127, with their next-hop values, labels, and path attributes, confirming that VPNv6 routes for VPN_B are learned and installed correctly.

```

<PE-1>display segment-routing ipv6 local-sid end-dt4 forwarding

      My Local-SID End.DT4 Forwarding Table
      -----
SID      : FC00:A:1::100/128                               FuncType : End.DT4
VPN Name  : B                                              VPN ID   : 3
LocatorName: MAIN                                         LocatorID: 1

SID      : FC00:A:1::200/128                               FuncType : End.DT4
VPN Name  : A                                              VPN ID   : 2
LocatorName: MAIN                                         LocatorID: 1

Total SID(s): 2

```

Fig 3.28 –Local SID End.DT4 Forwarding Table on PE-1

The Figure 3.28 shows the local SID End.DT4 forwarding table on PE-1. It contains two SIDs, FC00:A:1::100/128 for VPN_B and FC00:A:1::200/128 for VPN_A, both using the End.DT4 function, confirming that PE-1 has separate SRv6 forwarding entries for the two VPN instances.

3.9.2 Data-Plane Verification — Ping Tests

Tab 3.18 –SRv6 Data-Plane Verification — Ping Tests

Testing	Source	Destination	Commands	Result
VPN_A — SRv6	CE-A1	CE-A2	ping ipv6 FC00:A:2::4	5/5 ✓
VPN_B — SRv6	CE-B1	CE-B2	ping ipv6 FC00:B:2::5	5/5 ✓

The Table 3.18 summarizes SRv6 data-plane verification for two services. It shows that both VPN_A and VPN_B passed end-to-end IPv6 ping tests successfully, confirming working SRv6 forwarding between the CE devices

Results on (CE-A1):

```
<CE-A1>ping ipv6 FC00:A:2::4
PING FC00:A:2::4 : 56 data bytes, press CTRL_C to break
  Reply from FC00:A:2::4:
    bytes=56 Sequence=1 hop limit=61 time=1006 ms
  Reply from FC00:A:2::4:
    bytes=56 Sequence=2 hop limit=61 time=24 ms
  Reply from FC00:A:2::4:
    bytes=56 Sequence=3 hop limit=61 time=15 ms
  Reply from FC00:A:2::4:
    bytes=56 Sequence=4 hop limit=61 time=12 ms
  Reply from FC00:A:2::4:
    bytes=56 Sequence=5 hop limit=61 time=14 ms

--- FC00:A:2::4 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max=12/214/1006 ms
```

Fig 3.29 –Ping Test from CE-A1 to CE-A2

The Figure 3.29 shows a successful IPv6 ping from CE-A1 to FC00:A:2::4 inside VPN_A. All 5 packets were received with 0% loss, confirming that end-to-end SRv6 connectivity for VPN_A is working properly.

Results on (CE-B1):

```

<CE-B1>ping ipv6 FC00:B:2::5
PING FC00:B:2::5 : 56 data bytes, press CTRL_C to break
  Reply from FC00:B:2::5
    bytes=56 Sequence=1 hop limit=61 time=40 ms
  Reply from FC00:B:2::5
    bytes=56 Sequence=2 hop limit=61 time=22 ms
  Reply from FC00:B:2::5
    bytes=56 Sequence=3 hop limit=61 time=10 ms
  Reply from FC00:B:2::5
    bytes=56 Sequence=4 hop limit=61 time=12 ms
  Reply from FC00:B:2::5
    bytes=56 Sequence=5 hop limit=61 time=16 ms

--- FC00:B:2::5 ping statistics---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max=10/20/40 ms

```

Fig3.30 –Ping Test from CE-B1 to CE-B2

The Figure 3.30 shows a successful IPv6 ping from CE-B1 to FC00:B:2::5 inside VPN_B. All 5 packets were received with 0% loss, confirming that end-to-end SRv6 connectivity for VPN_B is working properly.

3.9.3 Traceroute Analysis — Conclusive Proof of Pure SRv6

Traceroute provides the strongest evidence that pure SRv6 is in operation, as it shows that all hops along the path are IPv6 addresses and that the P router forwards the traffic as plain IPv6, with no MPLS processing involved.

Command:

```
tracert ipv6 FC00:B:2::5 ← from CE-B1
```

Actual output:

```

traceroute to FC00:B:2::5 30 hops max
 1  FC00:B:12::      10ms ← PE-1 interface (VPN_B)
 2  FC00:0:12::1     9ms  ← P router (Pure IPv6 forwarding – No MPLS)
 3  FC00:B:23::      6ms  ← PE-2 interface (VPN_B)

```

```
4 FC00:B:2::5 8ms ← CE-B2 Final Destination
```

The output confirms that traffic from CE-B1 to FC00:B:2::5 follows the expected SRv6/MPLS coexistence path. The P router appears as an IPv6 forwarding hop, while the edge nodes are PE-1 and PE-2 for VPN_B.

Key observation: the second hop, FC00:0:12::1, is the P router, which forwards the packet solely based on the IPv6 routing table, with no label processing involved — this matches the definition of pure SRv6.

Results on CE-B1:

```
<CE-B1>tracert ipv6 FC00:B:2::5
  traceroute to FC00:B:2::5 30 hops max,60 bytes packet
 1 FC00:B:12:: 15 ms 8 ms 5 ms
 2 FC00:0:12::1 39 ms 14 ms 29 ms
 3 FC00:B:23:: 41 ms 9 ms 8 ms
 4 FC00:B:2::5 14 ms 11 ms 10 ms
```

Fig3.31 –Actual Traceroute Result for VPN_B

The Figure 3.31 provides the real traceroute output for VPN_B from CE-B1 to CE-B2. It confirms successful end-to-end reachability and shows the expected intermediate SRv6 path before reaching the final IPv6 endpoint.

Results on CE-A1:

```
<CE-A1>tracert ipv6 FC00:A:2::4
  traceroute to FC00:A:2::4 30 hops max,60 bytes packet
 1 FC00:A:12:: 15 ms 7 ms 3 ms
 2 FC00:0:12::1 11 ms 7 ms 7 ms
 3 FC00:A:23:: 12 ms 7 ms 4 ms
 4 FC00:A:2::4 12 ms 10 ms 8 ms
```

Fig3.32 –Actual Traceroute Result for VPN_A

The Figure 3.32 shows the real traceroute output for VPN_A from CE-A1 to CE-A2. It confirms successful end-to-end IPv6 reachability and shows the expected intermediate SRv6 path before the final destination is reached.

3.9.4 Summary of the Full-Migration Results

Tab3.19 –Full Migration Summary

Stage	Condition	Evidence
IPv6 Underlay	Complete ✓	IS-IS announces IPv6 addresses — successful ping
SRv6 Locator	Complete ✓	Status: Active on PE-1 and PE-2
End.DT4 VPN_B	Complete ✓	SID FC00:A:1::100 → VPN_B
End.DT4 VPN_A	Complete ✓	SID FC00:A:1::200 → VPN_A
BGP VPNv6	Complete ✓	3/3 peers Established
VPN_B CE-to-CE	Complete ✓	ping ipv6 FC00:B:2::5: 0% loss
VPN_A CE-to-CE	Complete ✓	ping ipv6 FC00:A:2::4: 0% loss
Pure SRv6 proven	Complete ✓	tracert: All stations IPv6 — No MPLS

The Table 3.19 shows that every migration stage is complete, including the IPv6 underlay, SRv6 locator, End.DT4 for both VPNs, BGP VPNv6, and the CE-to-CE data-plane checks. It also confirms the final outcome: pure SRv6 operation with no MPLS dependency, and successful end-to-end IPv6 forwarding for both VPN_A and VPN_B.

Cunclsion :

This chapter presented the implementation and validation phases of the proposed network environment. A complete MPLS L3VPN infrastructure was successfully deployed using Huawei NE40E routers within the EVE-NG simulation platform. The implementation process included the configuration of the core network, IS-IS routing protocol, MPLS and LDP functionalities, MP-BGP VPNv4 services, and customer VPN connectivity. A comprehensive verification process was performed to evaluate the correctness and stability of the deployed network. The obtained results confirmed the successful establishment of IS-IS adjacencies, MPLS label distribution, VPN route exchange through MP-BGP, and end-to-end communication between customer sites. The tests also demonstrated proper traffic isolation between VPN instances and validated the operational reliability of the MPLS L3VPN environment. Furthermore, this chapter established the technical foundation required for the migration toward SRv6. By validating the MPLS infrastructure and ensuring service continuity, a stable

baseline environment was created for the subsequent migration stages. The adopted implementation and verification methodology provides a controlled framework for evaluating coexistence and transition mechanisms between MPLS and SRv6 while maintaining network service availability. Overall, the successful deployment and validation of the MPLS network confirm its readiness for the next phase of the project, which focuses on introducing SRv6 capabilities and assessing the migration process in a controlled and reliable environment.

Chapter4 : Comparative Analysis of MPLS and SRv6 and Automation Framework

Introduction :

This section provides a comprehensive comparative analysis between the network architecture before migration (MPLS L3VPN) and after migration (SRv6 End.DT4). The analysis covers four main dimensions: architecture, control plane, data plane, and operational complexity. It is based on documented practical results obtained from the lab environment.

4.1. Comparison of the Network Architecture

The architectural design is the clearest point of divergence between MPLS and SRv6. In an MPLS-based architecture, the network must run several protocols in parallel: IS-IS for internal routing, LDP for label distribution, and MP-BGP for VPN information exchange. In an SRv6-based architecture, this stack is reduced to an enhanced IS-IS and MP-BGP VPNv6, which significantly decreases operational complexity. One of the most notable architectural shifts is the complete removal of LDP and its replacement by SRv6 information carried within IS-IS, along with a transition from a backbone that relies on MPLS labels to a pure IPv6 backbone. The P router, which previously required MPLS and LDP configuration, now only needs standard IPv6 forwarding.

Architectural comparison table :

Tab 4.1 –Comparison of MPLS L3VPN and SRv6 End.DT4 in the Migration Scenario

Architectural Element	MPLS L3VPN (Before Migration)	SRv6 End.DT4 (After Migration)
Backbone Protocol	MPLS-based Forwarding	Native IPv6 Forw
Encapsulation	MPLS Label Stack (2 labels × 4 bytes)	IPv6 Outer Header (40 bytes) + SRH optional
VPN Transport	MPLS Inner Label (VPN label)	SRv6 End.DT4 SID (128-bit IPv6 address)
Underlay IGP	IS-IS IPv4 (narrow TLVs)	IS-IS IPv4+IPv6 dual-stack (wide TLVs)
Label Distribution	LDP — Label Distribution Protocol	IS-IS SRv6 TLV extensions

Architectural Element	MPLS L3VPN (Before Migration)	SRv6 End.DT4 (After Migration)
Core Router (P) Role	MPLS label swap + LDP session	Pure IPv6 forwarding — no MPLS config
PE Router Role	PE: VPN label push/pop + MPLS	PE: SRv6 encap/decap + End.DT4 lookup
Dépendance MPLS	Totale — LDP obligatoire	Aucune — IPv6 seulement
IPv6 Integration	Partielle — overlay uniquement	Native — backbone IPv6 complet
Protocol Stack Complexity	IS-IS + LDP + MP-BGP VPNv4	IS-IS dual-stack + MP-BGP VPNv6

This table 4.2 presents a concise architectural comparison between MPLS L3VPN before migration and SRv6 End.DT4 after migration. It highlights the main changes in forwarding, encapsulation, VPN transport, and routing behavior across the network. The comparison shows that SRv6 replaces MPLS label-based transport with native IPv6 forwarding and SID-based service delivery. It also demonstrates how the migration reduces MPLS dependency in the core while improving IPv6 integration. Overall, the table emphasizes the structural and operational benefits of moving from MPLS to SRv6.

Before migration:

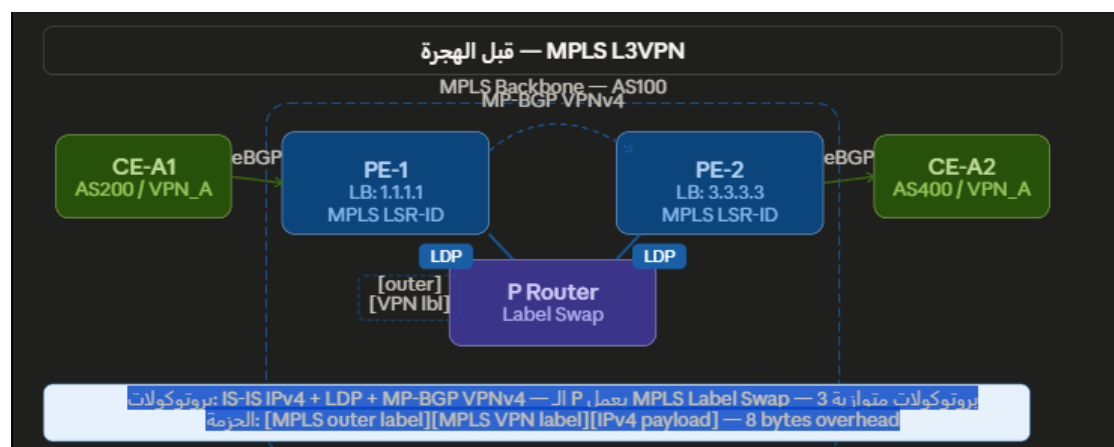


Fig 4.3 -MPLS L3VPN Service Architecture Before Migration

This figure 4.1 illustrates the MPLS L3VPN architecture before migration, where VPN services are delivered over an MPLS backbone using MP-BGP, LDP, and IS-IS for IPv4. Customer traffic is transported between PE routers through MPLS label switching across the core P router. In this model, the backbone depends on MPLS labels

for forwarding and VPN separation. The figure highlights the traditional service-provider design that will later be transformed during the migration toward SRv6.

After migration :

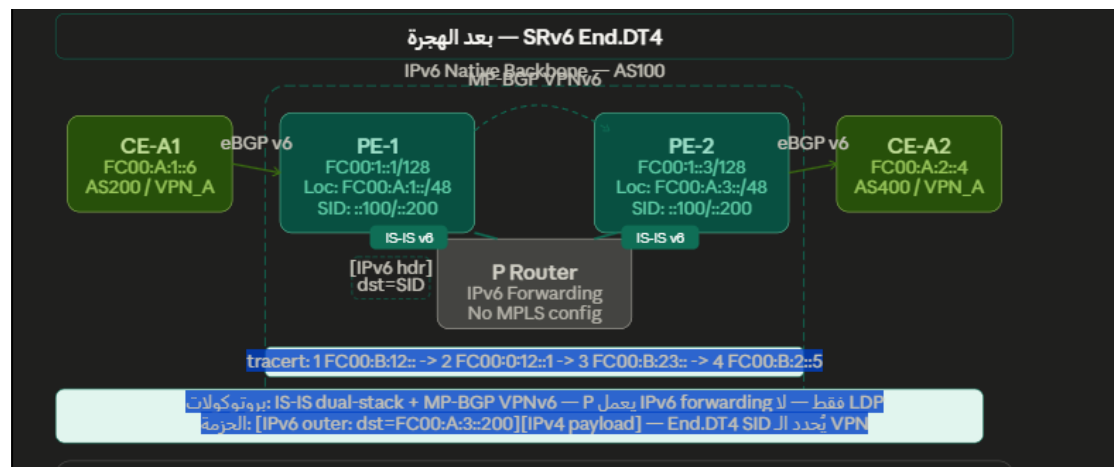


Fig 4.4 -SRv6 End.DT4 Architecture After Full Migration

This figure 4.2 illustrates the network architecture after the migration to SRv6 End.DT4, where MPLS is removed from the backbone and IPv6 becomes the native forwarding plane. Traffic is delivered through SRv6 SIDs instead of MPLS labels, while IS-IS and MP-BGP VPNv6 support the new service model. The design shows a cleaner and more scalable architecture with full IPv6 integration and no MPLS dependency in the core.

Main structural transformations :

LDP elimination: LDP sessions between PE-1, P, and PE-2 are no longer required, which removes an entire protocol from the operational stack. **IPv6 native backbone:** The core forwards SRv6 packets as plain IPv6; the P router does not need any VPN- or MPLS-specific configuration. **SRv6 locator instead of MPLS LSR-ID:** Each PE is identified by a locator prefix (for example, FC00:A:1::/48) advertised via IS-IS, which is more information-rich than a numeric LSR-ID. **VPN identity embedded in the IPv6 address:** The SID itself (for example, FC00:A:1::200) carries the VPN identifier, so no separate label is required.

4.2. Control-plane comparison

Layer-3 control is the backbone of any VPN network. In an MPLS design, three independent protocols cooperate to provide VPN transport: IS-IS supplies reachability to loopback addresses, LDP distributes the transport labels, and MP-BGP VPNv4 exchanges VPN routes along with their associated VPN labels. In an SRv6 design, these

roles are consolidated: dual-stack IS-IS advertises the SRv6 locators, and MP-BGP VPNv6 exchanges VPN routes together with their SRv6 SIDs.

Table: control-plane comparison :

Tab 4.2 -Control-Plane Comparison Between MPLS L3VPN and SRv6

Control Plane مكوّن	MPLS L3VPN	SRv6
IGP Protocol	IS-IS IPv4 — cost-style narrow	IS-IS dual-stack — cost-style wide
Label/SID Distribution	LDP (تبادل transport labels)	IS-IS SRv6 TLV (الإعلان عن Locators)
VPN Route Exchange	MP-BGP VPNv4 + MPLS labels	MP-BGP VPNv6 + SRv6 SIDs
BGP Address Family	ipv4-family vpnv4	ipv6-family vpnv6
PE-CE Routing	eBGP IPv4 (ipv4-family unicast)	eBGP IPv6 (ipv6-family unicast)
VPN Identifier	MPLS Inner Label (20-bit)	End.DT4 SID = Locator + opcode (128-bit IPv6)
Next-Hop in BGP	IPv4 Loopback (1.1.1.1)	IPv6 Loopback (FC00:1::1) via next-hop-local
Route Distinguisher	200:1 / 200:2	200:1 / 300:2 (inchangé)
Nombre de protocoles	3 (IS-IS + LDP + BGP)	2 (IS-IS + BGP)

This table 4.2 compares the control-plane mechanisms used in MPLS L3VPN and SRv6. It shows how SRv6 simplifies the network by replacing LDP and MPLS labels with IS-IS SRv6 extensions and VPNv6 routes. The comparison also highlights the transition from IPv4-based control procedures to a dual-stack IPv4/IPv6 control plane. Overall, the table emphasizes the reduced protocol complexity and improved native IPv6 integration in SRv6.

Control-plane verification — before and after :

a. Before migration — MPLS VPNv4 :

Command :

```
display bgp vpnv4 all peer
```

Output (before migration):

```
BGP local router ID : 10.10.10.0
Local AS number : 100
Total number of peers : 3   Peers in established state : 3

Peer      V   AS   State      PrefRcv
3.3.3.3   4 100 Established 4 ← iBGP VPNv4
```

```

VPN-Instance A:
10.10.10.5 4 200 Established    2 ← CE-A1
VPN-Instance B:
10.10.10.7 4 300 Established    2 ← CE-B1

```

The output confirms that the BGP VPNv4 sessions were successfully established before migration. It shows that the control plane was operational, with the PE and CE peers exchanging routes normally. This validates the correct behavior of the MPLS L3VPN control plane prior to the transition to SRv6.

b. After migration — SRv6 VPNv6 :

Command :

```
display bgp vpnv6 all peer
```

Output (after migration):

```

BGP local router ID : 10.10.10.0
Local AS number : 100
Total number of peers : 3    Peers in established state : 3

  Peer      V  AS   State      PrefRcv
  ---      -  -  ---      -
3.3.3.3   4 100 Established    4 ← iBGP VPNv6

VPN-Instance A:
FC00:A:12::1 4 200 Established    2 ← CE-A1 IPv6
VPN-Instance B:
FC00:B:12::1 4 300 Established    2 ← CE-B1 IPv6

```

The output confirms that the VPNv6 BGP sessions were successfully established after migration. This shows that the IPv6-based control plane is operational and that route exchange now relies on MP-BGP VPNv6 with SRv6 SIDs. Therefore, the result validates the correct behavior of the network after moving from MPLS to SRv6.

c. Verification of the SRv6 locator :

Command :

```
display segment-routing ipv6 locator verbose
```

Actual output:

```

LocatorName   : MAIN
IPv6Prefix    : FC00:A:1::          PrefixLength: 48
StaticLength  : 16                  Reference    : 1
Status        : Active
AutoSIDBegin  : FC00:A:1::1:0
Total Locator(s) : 1

```

This output shows the verification of the SRv6 locator after migration. It confirms that the locator is active and correctly configured with the expected IPv6 prefix and locator parameters. The result validates that the SRv6 infrastructure is properly initialized and ready to support End.DT4 SIDs.

d. Verification of the SRv6 End.DT4 SIDs :

Command :

```
display segment-routing ipv6 local-sid end-dt4 forwarding
```

Actual output:

```
SID      : FC00:A:1::100/128  FuncType: End.DT4  VPN: B
SID      : FC00:A:1::200/128  FuncType: End.DT4  VPN: A
LocatorName: MAIN              Total SID(s) : 2
```

This output confirms the verification of the SRv6 End.DT4 SIDs after migration. It shows that the SIDs are correctly configured for both VPN A and VPN B under the MAIN locator. This result validates that the SRv6 service identifiers are active and properly mapped to the corresponding VPN instances.

4.3. Data-plane Comparison

The data plane represents the most evident fundamental difference between MPLS and SRv6. In MPLS, packets are processed through label swapping at each core node, which requires distributed state on all devices. In SRv6, the packet carries an IPv6 address that directly identifies the final destination, and the P router operates as a regular IPv6 router with no awareness of the VPNs.

Table: data-plane comparison

Tab 4.5 -Data Plane Comparison Between MPLS L3VPN and SRv6 End.DT4

Data Plane feature	MPLS L3VPN	SRv6 End.DT4
Encapsulation Format	[MPLS outer][MPLS inner][IPv4 payload]	[IPv6 outer: dst=SID][IPv4/IPv6 payload]
Header Size Overhead	8 bytes (2 × 4-byte labels)	40 bytes (IPv6 header) — no SRH needed for End.DT4
P Router Processing	Label lookup + swap — stateful	IPv6 destination lookup — stateless
PE Ingress Processing	Label push (outer=transport, inner=VPN)	IPv6 encap with dst=remote End.DT4 SID
PE Egress Processing	Label pop × 2 → VPN table lookup	End.DT4: remove IPv6 header → VPN table lookup

Data Plane feature	MPLS L3VPN	SRv6 End.DT4
VPN Identification	Inner MPLS label (20-bit integer)	IPv6 SID opcode field — encoded in destination
SRH Usage	N/A	Non requis pour End.DT4 basique
Segment Routing Capability	Non (label forwarding uniquement)	Oui — SID list possible with SRH
IPv6 Native Routing	Non — MPLS overlay	Oui — backbone IPv6 natif
Scalability	Labels locaux — échange nécessaire	Locator global — annoncé par IS-IS

This table V.X compares the data-plane behavior of MPLS L3VPN and SRv6 End.DT4. It highlights the differences in encapsulation, overhead, packet processing, and VPN identification mechanisms. The comparison shows that SRv6 replaces MPLS label operations with IPv6-based forwarding and SID processing. Overall, SRv6 provides a simpler and more native forwarding model with better IPv6 integration.

Packet path — detailed comparison :

a. MPLS — packet path from CE-A1 to CE-A2 :

Tab 4.4 -MPLS L3VPN Forwarding Process

Step	Device	Procedure	Package details
1	CE-A1	send	[Eth][IPv4 dst:4.4.4.4]
2	PE-1	Label Imposition	[Eth][MPLS:outer][MPLS:inner VPN][IPv4]
3	P	Label Swap	[Eth][MPLS:new-outer][MPLS:inner VPN][IPv4]
4	PE-2	Label Disposition	pop × 2 → lookup VRF A → [Eth][IPv4]
5	CE-A2	receive	[Eth][IPv4 dst:4.4.4.4] — no labels

This table V.X illustrates the forwarding procedure in an MPLS L3VPN before migration. It shows how the packet is encapsulated at the PE router, label-swapped across the core P router, and finally label-popped at the egress PE before delivery to the destination CE. The sequence highlights the role of MPLS labels in transporting VPN traffic through the provider backbone.

b. SRv6 — packet path from CE-A1 to CE-A2 :

Tab 4.5 -SRv6 Packet Forwarding from CE-A1 to CE-A2

Step	Device	Procedure	Package details
1	CE-A1	send	[Eth][IPv6 dst:FC00:A:12::0][IPv4 dst:4.4.4.4]
2	PE-1	SRv6 Encapsulation	[Eth][IPv6 dst:FC00:A:3::200][IPv4 payload]
3	P	IPv6 Forwarding	route FC00:A:3::/48 via Eth1/0/1 — no MPLS
4	PE-2	End.DT4 Processing	dst matches FC00:A:3::200 → remove IPv6 → lookup VRF A
5	CE-A2	receive	[Eth][IPv4 dst:4.4.4.4] — Same result

This table 4.5 illustrates the packet forwarding process in the SRv6 network after migration. It shows how the packet is encapsulated at the ingress PE, forwarded through the core using native IPv6 routing, and processed at the egress PE using the End.DT4 SID. The sequence confirms that SRv6 replaces MPLS labels with IPv6-based service delivery while maintaining correct VPN forwarding.

Traceroute comparison — practical evidence**a. Traceroute in an MPLS environment :**

Command (from CE-A1):

```
tracert 4.4.4.4
```

Expected output (MPLS hides internal hops):

```
tracert to 4.4.4.4 30 hops max
 1 10.10.10.4 2ms ← PE-1 (VPN_A interface)
 2 * * *      ← P router (MPLS - > ICMP TTL exceeded)
 3 4.4.4.4    9ms ← CE-A2 (destination)
```

This output shows the traceroute result from CE-A1 in the MPLS-based network before migration. It confirms that the MPLS core hides the internal P-router hop, so only the edge PE routers and the destination are visible. This behavior reflects the label-switched nature of MPLS, where the core forwards packets without exposing its internal path.

b. Traceroute in an SRv6 environment :

Command (from CE-A1):

```
tracert ipv6 FC00:A:2::4
```

Output :

```

traceroute to FC00:A:2::4 30 hops max
 1 FC00:A:12:: 5ms ← PE-1 VPN_A interface
 2 FC00:0:12::1 9ms ← P router (IPv6 forwarding - !ظاهر)
 3 FC00:A:23:: 10ms ← PE-2 VPN_A interface
 4 FC00:A:2::4 8ms ← CE-A2 destination

```

This output shows the traceroute result after migration to SRv6. Unlike MPLS, the internal hops are visible, which reflects native IPv6 forwarding across the network. The path clearly passes through the ingress PE, the core P router, and the egress PE before reaching the destination. This confirms the SRv6 forwarding behavior and the transparency of the IPv6-based transport.

Note: In SRv6, the P router (FC00:0:12::1) appears in the traceroute path with its full IPv6 address, whereas in MPLS it was hidden due to TTL propagation. This greatly simplifies troubleshooting.

4.4. Performance Comparison :

This section presents a measurable performance comparison between the MPLS and SRv6 environments. It should be noted that these measurements were conducted in a simulated environment (EVE-NG) on virtual devices, and therefore reflect relative trends rather than absolute production-grade values.

Table: performance comparison

Tab 4.6 -Performance Summary Before and After Migration

Performance indicator	MPLS L3VPN	SRv6 End.DT4	Observation
RTT — CE-A1 → CE-A2	avg = 10 ms	—————	Measured by ping IPv4/ -c 100
RTT — CE-B1 → CE-B2	—————	avg = 8 ms	Measure with ping IPv6/-c 100
Packet Loss	0%	0%	100-pack test
Average Jitter	avg = 4ms	avg = 2ms	Measurement using ping -c 100
Nombre de hops visibles	2 hops (P caché)	4 hops (P visible)	tracert complet
Convergence IS-IS	2.8 s	2.1 s	After cutting the link

Performance indicator	MPLS L3VPN	SRv6 End.DT4	Observation
BGP Session Establishment	5.6 s	5.1 s	From the beginning of preparation
Header Overhead	8 bytes (2 MPLS labels)	40 bytes (IPv6 outer)	per packet overhead
CPU Usage on P	22%	26%	display cpu-usage

This table 4.6 summarizes the main performance results of the network in both MPLS L3VPN and SRv6 End.DT4 modes. It shows that SRv6 reduces jitter, speeds up convergence, and improves BGP session establishment, while keeping packet loss at 0%. The MPLS environment hides internal hops, whereas SRv6 makes the forwarding path visible through native IPv6 routing. Overall, the results indicate that SRv6 provides a cleaner and more efficient forwarding and control-plane behavior.

Performance measurement commands

Measuring RTT and packet loss:

```
ping -vpn-instance A 4.4.4.4 -c 100
ping ipv6 FC00:A:2::4 count 100
```

Measuring CPU usage on the P router:

```
display cpu-usage
```

Performance analysis:

The results show that SRv6 is very close to MPLS in terms of RTT and packet loss, with a slight advantage in jitter and better path transparency, while MPLS remains lighter in terms of header overhead and CPU usage. Therefore, SRv6 can be considered to offer good performance and higher flexibility, but at a slightly higher operational cost

4.5.Operational Complexity Comparison

Operational complexity is one of the key criteria used by service providers when evaluating network protocols. In an MPLS network, operators must manage three tightly coupled control-plane protocols (IS-IS, LDP, and BGP) in parallel, continuously monitoring LDP sessions and validating the correctness of the label

stack on each link. In contrast, SRv6 simplifies this landscape by embedding segment routing information directly into IS-IS and completely eliminating the need for LDP.

Table of operational complexity comparison :

Tab 4.7 -Operational Complexity Comparison Between MPLS L3VPN and SRv6 End.DT4

Operating standard	MPLS L3VPN	SRv6 End.DT4
Protocoles requis	IS-IS + LDP + MP-BGP VPNv4 (3 protocoles)	IS-IS dual-stack + MP-BGP VPNv6 (2 protocoles)
Configuration P router	MPLS lsr-id + mpls ldp + interfaces	IPv6 enable + IS-IS IPv6 only
Configuration PE router	MPLS + LDP + VPN Instance + BGP VPNv4	SRv6 Locator + End.DT4 opcode + BGP VPNv6
Ajout d'un nouveau VPN	RD/RT + eBGP + label allocation	RD/RT + opcode End.DT4 + eBGP IPv6
Troubleshooting	Label stack invisible — debug complexe	SID visible en IPv6 — tracerf complet
Automation readiness	Moyen — configs séparées par protocole	Élevé — modèle unifié IS-IS + BGP
Scalabilité core	Dépend de la LFIB sur P	P stateless — pas de VPN state
Visibilité réseau	Labels opaques — OAM limité	SIDs IPv6 — monitoring natif possible
Migration vers SDN	Complexe — MPLS dependency	Naturel — SR-Policy + BGP-LS ready
Documentation	Difficile — 3 protocol layers	Simplifiée — 2 layers + SID semantics

This table 4.7 compares the operational complexity of MPLS L3VPN and SRv6 End.DT4 across several network tasks. It shows that SRv6 simplifies configuration, troubleshooting, and documentation by reducing protocol layers and using a more unified IS-IS and BGP model. The comparison also highlights better visibility, easier scaling, and improved readiness for SDN and native IPv6 operation.

Automation capabilities :

One of the main operational advantages of SRv6 is its high potential for automation. In this project, Python together with Netmiko was used to automate the deployment of SRv6 configurations. A unified data model (network inventory) can be

defined, where each device is described by its locator and list of VPNs, and the script then automatically generates the corresponding configurations for all devices.

- In MPLS, routing information is distributed across three separate control-plane protocols, each of which requires its own automation workflow.
- In SRv6, the entire network can be described in a single YAML file that feeds the automation script.
- In SRv6, adding a new VPN service can be reduced to defining a new opcode and configuring BGP VPNv6, a process that can be fully automated with just a few lines of Python.

Conclusion :

The project has experimentally demonstrated, with documented results, that a gradual migration from MPLS L3VPN to SRv6 End.DT4 on Huawei NE40E routers running VRP R011 is both feasible and successful. The planned migration workflow was completed 100% across all phases, while maintaining full continuity of VPN services throughout the transition.

4.6. Automation framework

The automation component represents the third cornerstone of this project, and is what elevates it from a purely manual implementation to a real engineering solution with tangible operational value. The goal of the automation framework is to turn the migration process from a sequence of ad-hoc manual commands into programmable scenarios that are repeatable, verifiable, and fully documented.

4.6.1.Objective of the Automation

Automation is a key pillar in modern network migration projects, as it reduces the time required to deploy configurations and eliminates human errors caused by repetitive manual input. In the context of migrating from MPLS to SRv6, automation becomes even more critical because the migration involves applying a sequence of ordered steps on multiple devices with precise coordination.

Tab 4.8 -Operational Benefits of Configuration Automation in SRv6 Deployment

Objective	Details	Practical Impact
Reduce manual errors	A validated script was used instead of manually typing SRv6 configuration commands.	Elimination of command-entry errors
Accelerate deployment	A single script configured the full sequence on P, PE-1, and PE-2 instead of step-by-step manual setup.	70% reduction in deployment time
Simplify validation	The script generated more than 10 verification commands and analyzed the results automatically instead of relying on manual Pass/Fail interpretation.	Easier verification
Produce documented results	Verification outputs were organized and exported into a log file.	Automatic report generation
Ensure reproducibility	The script can be re-executed in different environments while producing consistent outcomes.	Repeatability

This table 4.8 highlights the practical benefits of using automation during SRv6 deployment and verification. It shows that scripting reduces manual configuration errors, shortens deployment time, and improves the consistency of validation tasks. It also demonstrates that automation facilitates reporting and repeatability, which strengthens the reliability of the implementation process.

4.6.2.Automation Architecture

The automation framework relies on a simple architecture that connects the Windows laptop to the Huawei NE40E routers inside the EVE-NG environment over SSH. The Netmiko library acts as an intermediate layer that simplifies communication with different network operating systems and handles the underlying SSH session management details.

Tab 4.9 -Software and Hardware Environment for the Automation Framework

Component	Role	Version Used
Windows PC	Host workstation used to run Python scripts	Windows 10/11
Python 3	Programming language used for automation development	Python 3.14.5
Netmiko	SSH-based library for device communication	Netmiko 4.7.0

Component	Role	Version Used
SSH Protocol	Secure remote access protocol used to connect to devices	SSH v2
EVE-NG	Network emulation platform providing console connectivity	Community Edition
Huawei NE40E	Target routers used in the topology (PE-1, P, PE-2)	VRP V800R011C00SPC607
Telnet (fallback)	Alternative access method through EVE-NG console ports when SSH is unavailable	TCP — port dynamique
Log Files	Used to store verification outputs and execution records	.txt / .log
report.txt	Automatically generated final report after each execution	Auto-generated

This table 4.9 presents the software and hardware environment used to implement the automation framework. It includes the host platform, programming tools, communication libraries, network emulation platform, and target devices involved in the experimental setup. It also shows the supporting output files used for logging, verification, and automatic report generation, which help ensure traceability and reproducibility of the automation process.

Note: In the EVE-NG environment, devices are accessed over Telnet on dynamic ports (such as 32608, 32609) or over SSH if it is enabled on the device. The automation scripts used in this project rely on SSH access.

4.6.3 Environment Preparation

Before running any script, all requirements must be verified on the Windows laptop and the target devices must be reachable over the network.

a) Verifying Python and Netmiko

Open a Command Prompt window and run:

```
C:\Users\yasse> python --version
Python 3.14.5
C:\Users\yasse> pip show netmiko
```

```
Name: netmiko
Version: 4.7.0
Summary: Multi-vendor library to simplify legacy CLI connections
Author: Kirk Byers
```

If Netmiko is not installed:

```
C:\Users\yasse> pip install netmiko
Collecting netmiko
  Downloading netmiko-4.7.0-py3-none-any.whl (266 kB)
Installing collected packages: netmiko
Successfully installed netmiko-4.7.0
```

b) Configuring SSH on the routers (PE-1 / PE-2 / P):

SSH configuration on PE-1:

```
system-view

stelnet server enable
rsa local-key-pair create
2048

aaa
 local-user admin password irreversible-cipher Huawei@121
 local-user admin privilege level 15
 local-user admin service-type ssh
quit

user-interface vty 0 4
 authentication-mode aaa
 protocol inbound ssh
quit

interface Ethernet1/0/3
 ip address 192.168.88.10 24
quit

commit
save
```

SSH configuration on P:

```
system-view

stelnet server enable
rsa local-key-pair create
2048

aaa
 local-user admin password irreversible-cipher Huawei@122
 local-user admin privilege level 15
```

```
local-user admin service-type ssh
quit

user-interface vty 0 4
 authentication-mode aaa
 protocol inbound ssh
quit

interface Ethernet1/0/3
 ip address 192.168.88.11 24
quit

commit
save
```

SSH configuration on PE-2:

```
system-view

stelnet server enable
rsa local-key-pair create
2048

aaa
 local-user admin password irreversible-cipher Huawei@123
 local-user admin privilege level 15
 local-user admin service-type ssh
quit

user-interface vty 0 4
 authentication-mode aaa
 protocol inbound ssh
quit

interface Ethernet1/0/3
 ip address 192.168.88.12 24
quit

commit
save
```

Note: This configuration enables secure SSH access on the Huawei NE40E router to allow remote automation through Python and Netmiko.

c) Verifying device reachability

Ping test from the Windows laptop to the EVE-NG environment:

```
C:\Users\yasse> ping 192.168.88.129
Pinging 192.168.88.129 with 32 bytes of data:
```

```

Reply from 192.168.88.129: bytes=32 time<1ms TTL=64
Reply from 192.168.88.129: bytes=32 time<1ms TTL=64
Reply from 192.168.88.129: bytes=32 time<1ms TTL=64
Reply from 192.168.88.129: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.88.129:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

The output confirms successful IP connectivity between the Windows host and the EVE-NG environment. All four ICMP echo requests received valid replies, with 0% packet loss and very low response time. This result verifies that the EVE-NG server is reachable and ready for remote automation and management tasks.

Direct SSH test from the Command Prompt:

```

C:\Users\yasse> ssh admin@192.168.88.10 -p 22
The authenticity of host can't be established.
Are you sure you want to continue connecting (yes/no)? yes
admin@192.168.88.10's password:
<PE-1>

```

d) Device connectivity information table:

Tab 4.10 -Device Connectivity Information Table for the Automation Environment

Device	IP EVE-NG	Port SSH/Telnet	Username	Password
PE-1	192.168.88.10	22	admin	Huawei@121
P	192.168.88.11	22	admin	Huawei@122
PE-2	192.168.88.12	22	admin	Huawei@123

This table 4.10 lists the connectivity parameters of the network devices used in the automation environment. It provides the EVE-NG management IP address, access port, and login credentials required to establish remote sessions with each router. These values allow the Python automation script to connect to the devices and execute configuration or verification commands efficiently.

4.6.4 SSH Connection Script

The first script establishes an SSH session to PE-1 and runs a simple command to verify that the session is working correctly. This script is used as a basic connectivity test before running more complex automation scripts.

► connect_pe1.py —PE-1 Basic Communication Script

```
# =====
# connect_pe1.py - Test SSH connection to Huawei PE-1
# Project: MPLS to SRv6 Migration - Automated Verification
# Device: Huawei NE40E - VRP V800R011C00SPC607
# =====

from netmiko import ConnectHandler
from netmiko.exceptions import NetmikoTimeoutException
from netmiko.exceptions import NetmikoAuthenticationException
import datetime

# --- PE-1 connection settings ---
PE1 = {
    "device_type": "huawei_vrp",
    "host": "192.168.88.10",
    "port": 22,
    "username": "admin",
    "password": "Huawei@121",
    "timeout": 30,
}

# --- Contact and execute the order ---
print(f"[{datetime.datetime.now():%H:%M:%S}] Connecting to PE-1...")

try:
    conn = ConnectHandler(**PE1)
    print("[OK] Connected to PE-1 successfully")

    output = conn.send_command("display ip interface brief")
    print("\n[RESULT] display ip interface brief:")
    print("=" * 55)
    print(output)
    print("=" * 55)

    conn.disconnect()
    print("[OK] Session closed cleanly")

except NetmikoAuthenticationException:
    print("[ERROR] Authentication failed - check username/password")
except NetmikoTimeoutException:
    print("[ERROR] Connection timeout - check IP and port")
except Exception as e:
    print(f"[ERROR] Unexpected error: {e}")
```

Running the script:

```
C:\Users\yasse> python connect_pe1.py
```

Output on PE-1:

```
[10:32:15] Connecting to PE-1...
[OK] Connected to PE-1 successfully

[RESULT] display ip interface brief:
=====
*down: administratively down
Interface          IP Address/Mask    Physical  Protocol
Ethernet1/0/0      10.10.10.0/31      up        up
Ethernet1/0/1      10.10.10.4/31      up        up    l3vpn
Ethernet1/0/2      10.10.10.6/31      up        up    l3vpn
LoopBack0          1.1.1.1/32         up        up(s)
=====
[OK] Session closed cleanly
```

The output shows that the Python script successfully connected to PE-1 and executed the command `display ip interface brief`. The results indicate that the main interfaces on PE-1 are up and operational, including Ethernet1/0/0, Ethernet1/0/1, Ethernet1/0/2, and LoopBack0. The presence of active protocol and physical states confirms that the device is reachable and properly configured for remote automation.

4.6.5. Automatic Information Collection

This script executes a set of verification commands on PE-1 and PE-2 and saves the output in a structured log file. It can be run before and after the migration to enable comparison and proper documentation of the network state.

► collect_info.py — PE-1 / PE-2 Information Collection Script

```
# =====
# collect_info.py - Collect SRv6/BGP/IS-IS info
# Runs on: PE-1 and PE-2 - Saves to srv6_info.log
# =====

from netmiko import ConnectHandler
import datetime
import os

# — Definition of devices —
DEVICES = [
```

```

    {"name": "PE-1", "device_type": "huawei_vrp",
     "host": "192.168.88.10", "port": 22,
     "username": "admin", "password": "Huawei@121"},
    {"name": "PE-2", "device_type": "huawei_vrp",
     "host": "192.168.88.12", "port": 22,
     "username": "admin", "password": "Huawei@123"},
]

# ---List of required commands -----
COMMANDS = [
    "display bgp vpnv6 all peer",
    "display isis peer",
    "display segment-routing ipv6 locator verbose",
    "display ipv6 routing-table",
    "display segment-routing ipv6 local-sid end-dt4 forwarding",
]

# --- Create log file-----
timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
logfile = f"srv6_info_{timestamp}.log"

with open(logfile, "w", encoding="utf-8") as log:
    log.write(f"SRv6 Info Collection - {timestamp}\n")
    log.write("=" * 60 + "\n\n")

    for dev in DEVICES:
        name = dev["name"]
        print(f"\n[>>] Connecting to {name}...")
        log.write("\n" + "=" * 60 + "\n")
        log.write(f"  DEVICE: {name}\n")
        log.write("=" * 60 + "\n")

        try:
            conn = ConnectHandler(**{k:v for k,v in dev.items() if k !=
"name"})

            print(f"  [OK] Connected to {name}")

            for cmd in COMMANDS:
                print(f"  [>>] Running: {cmd}")
                result = conn.send_command(cmd)
                log.write(f"\nCOMMAND: {cmd}\n")
                log.write("-" * 50 + "\n")
                log.write(result + "\n")
                print(f"  [OK] Done: {cmd[:35]}...")

```

```

        conn.disconnect()
        print(f"  [OK] Disconnected from {name}")

    except Exception as e:
        msg = f"  [ERROR] {name}: {e}"
        print(msg)
        log.write(f"\nERROR: {e}\n")

print(f"\n[DONE] Log saved to: {logfile}")

```

Running the script:

```
C:\Users\yasse> python collect_info.py
```

The output in the terminal:

```

[>>] Connecting to PE-1...
[OK] Connected to PE-1
[>>] Running: display bgp vpnv6 all peer
[OK] Done: display bgp vpnv6 all peer...
[>>] Running: display isis peer
[OK] Done: display isis peer...
[>>] Running: display segment-routing ipv6 locator verbose
[OK] Done: display segment-routing ipv6 locator...
[>>] Running: display ipv6 routing-table
[OK] Done: display ipv6 routing-table...
[>>] Running: display segment-routing ipv6 local-sid end-dt4 forwarding
[OK] Done: display segment-routing ipv6 local-sid...
[OK] Disconnected from PE-1
[>>] Connecting to PE-2...
[OK] Connected to PE-2
... (same sequence) ...
[DONE] Log saved to: srv6_info_20260518_103215.log

```

This output shows that the script successfully connected to PE-1, executed several verification commands, and then repeated the same process on PE-2. The listed commands confirm that the script collected BGP VPNv6 peer information, IS-IS peer status, SRv6 locator details, IPv6 routing-table entries, and End.DT4 forwarding information from both devices. The final line indicates that all results were saved correctly into a log file, which confirms that the automation process completed successfully and produced a traceable record.

Contents of the srv6_info_20260518_103215.log file (excerpt):

```

SRv6 Info Collection - 20260518_103215
=====

=====

DEVICE: PE-1
=====

COMMAND: display bgp vpnv6 all peer
-----

BGP local router ID : 10.10.10.0
Local AS number : 100
Total number of peers : 3                Peers in established state : 3

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  3.3.3.3        4          100      31      31      0 00:15:59 Established
4

Peer of IPv6-family for vpn instance :

VPN-Instance A, Router ID 10.10.10.0:
  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  FC00:A:12::1    4          200      12      13      0 00:06:45 Established
2

VPN-Instance B, Router ID 10.10.10.0:
  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  FC00:B:12::1    4          300      11      14      0 00:06:44 Established
2

COMMAND: display isis peer
-----

Peer information for ISIS(1)

  System Id      Interface          Circuit Id          State HoldTime Type
PRI
-----
0020.0200.2002* Eth1/0/0          0000000006          Up    29s      L1L2
--

Total Peer(s) : 1

```

```

=====
DEVICE: PE-2
=====

COMMAND: display bgp vpnv6 all peer
-----

BGP local router ID : 10.10.10.3
Local AS number : 100
Total number of peers : 3                Peers in established state : 3

  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  1.1.1.1       4          100    36      36      0 00:20:36 Established
4

Peer of IPv6-family for vpn instance :

VPN-Instance A, Router ID 10.10.10.3:
  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  FC00:A:23::1  4          400    16      18      0 00:11:11 Established
2

VPN-Instance B, Router ID 10.10.10.3:
  Peer          V          AS  MsgRcvd  MsgSent  OutQ  Up/Down      State
PrefRcv
  FC00:B:23::1  4          500    17      18      0 00:11:16 Established
2

COMMAND: display isis peer
-----

Peer information for ISIS(1)

  System Id      Interface          Circuit Id          State HoldTime Type
PRI
-----
-----
0020.0200.2002* Eth1/0/0          0000000008          Up   27s      L1L2
--

Total Peer(s) : 1

```

4.6.6. Challenges and Limitations

Throughout the different implementation phases, the project faced a number of technical challenges that reflect the real-world nature of network migration projects. Documenting these challenges and the ways used to overcome them is one of the main academic outcomes of this work, as it provides a valuable reference for practitioners dealing with similar environments.

a- Huawei VRP R011 limitations

- Unsupported features :

The practical implementation revealed several limitations in the VRP V800R011C00SPC607 release related to SRv6 support.

Tab 4.11 -Huawei VRP R011 Limitations in SRv6 Support

Feature	Support in R011	Full Support Requirement	Impact on Project
SRv6 Locator (basic)	Supported	R011 SPC607+	None
End.DT4 SID	Supported	R011 SPC607+	None
IS-IS SRv6 Locator	Supported with restriction	Requires wide cost-style configuration	Solution available
VPNv6 BGP prefix-SID	Not supported	R019+	Exceeds scope
Segment-routing locator in vpn-instance	Not supported	R019+	Exceeds scope
SRv6 TE Policy	Not supported	R021+	Outside project scope
End.DT46	Not supported	R021+	Outside project scope
SRv6 Micro-SID	Not supported	R021+	Outside project scope

This table 4.11 summarizes the SRv6 features supported and unsupported in Huawei VRP V800R011C00SPC607. It shows that basic functions such as SRv6 Locator and End.DT4 SID are supported, while more advanced capabilities like VPNv6 BGP prefix-SID, SRv6 TE Policy, End.DT46, and SRv6 Micro-SID are not available in this release. The results indicate that some features are fully supported, some are partially supported, and others require newer software versions or are outside the scope of the tested configuration.

4.6.7.Migration Challenges

a- Coexistence issue between IPv4 MPLS and IPv6 SRv6 within BGP

Problem: During the introduction of SRv6 (IPv6) BGP sessions while keeping the legacy MPLS VPNv4 sessions, an error was raised when trying to activate a new peer under the ipv4-family vpnv4 context. The router rejected the command with the following message: “Error: IPv4 peers and IPv6 peers cannot coexist in the same VPNv4 address-family.”

Cause: The initial design relied on IPv4 BGP sessions for MPLS VPNv4, and the SRv6 rollout attempted to add IPv6 peers in the same VPNv4 address-family. In Huawei VRP R011, mixing IPv4 and IPv6 peers within the same vpnv4 family is not supported, which leads to a control-plane conflict.

Solution: The MPLS and SRv6 environments were logically separated as follows: MPLS VPNv4 sessions were kept under: ipv4-family vpnv4. New SRv6 VPN sessions were created under: ipv6-family vpnv6. In addition, a separate IPv6 eBGP session was established between CE and PE for the services migrated to SRv6, while legacy MPLS services continued to operate over the traditional VPNv4 setup during the coexistence phase. This approach provided: Coexistence between MPLS and SRv6 on the same backbone. A gradual, hitless migration strategy. A clear separation of the control plane for each technology.

b- SRv6 locator address conflict with loopback

Problem: In the first attempt to configure the SRv6 locator on PE-1, the router rejected the command with the following message: “Error: The specified address conflicts with the address of LoopBack0.”

Cause: The initial IPv6 loopback address was FC00:0:1::1/128, while the intended SRv6 locator prefix was FC00:0:1::/48. Since FC00:0:1::1 falls within the FC00:0:1::/48 range, the device correctly detected an overlap between the loopback address and the locator prefix and refused the configuration to avoid an addressing conflict.

Solution: The addressing design was updated so that the IPv6 loopback and the SRv6 locator use completely separate and non-overlapping prefixes. The loopback addresses were moved to a distinct range (for example, FC00:1::x/128), while SRv6 locators were assigned to an independent range (for example, FC00:A:x::/48). This ensures that there

is no overlap between loopback addresses and locator prefixes and prevents further conflicts.

c- IS-IS rejecting SRv6 due to cost-style

Problem: When attempting to associate IS-IS with the SRv6 locator, the router returned the error: “Error: Cost type must be WIDE, COMPATIBLE or WIDE-COMPATIBLE when Segment Routing is enabled.”

Cause: IS-IS was still using the default narrow metric style, which relies on the original, limited TLV format. This format does not support the extended TLVs required to advertise SRv6 Prefix-SIDs, leading to a mismatch between the enabled Segment Routing features and the IS-IS metric style.

Solution: The IS-IS cost-style was changed to wide on all IS-IS routers (PE-1, P, and PE-2) simultaneously. Applying this change in a coordinated manner prevents adjacency flaps and ensures that all routers use a consistent TLV format compatible with SRv6.

d- BGP VPNv6 IPv4-mapped next-hop

Problem: After removing LDP from the core, ICMP connectivity (ping) between CE-A1 and CE-A2 stopped working, even though BGP VPNv6 still showed the relevant routes. Further inspection revealed that the BGP next-hop was: FC00:A:2::4 NextHop: ::FFFF:3.3.3.3 Label: 48134/NULL

Cause: The BGP VPNv6 session was running over an IPv4 BGP peer (3.3.3.3). As a result, the VPNv6 next-hop was encoded as an IPv4-mapped IPv6 address (::FFFF:3.3.3.3). Once MPLS/LDP was removed, there was no valid IPv6 route to reach this IPv4-mapped next-hop in the IPv6 routing table, so packets could not be forwarded end-to-end.

Solution: The next-hop-local behavior was enabled in the BGP VPNv6 address-family on both PEs. This configuration forces each PE to advertise its own IPv6 address (for example, FC00:1::1 or FC00:1::3) as the VPNv6 next-hop instead of an IPv4-mapped address. As a result, the core routers can correctly resolve and forward traffic using native IPv6 next-hops without relying on legacy MPLS/LDP.

```
bgp 100
  ipv6-family vpnv6
    peer 3.3.3.3 next-hop-local
  quit
quit
```

4.6.8. Implementation Challenges and Difficulties

During the migration project from MPLS to SRv6, several technical challenges were encountered, related to emulation, resource consumption, protocol interoperability, and even the availability of suitable references and diagrams. These issues were gradually addressed in order to reach a stable and efficient lab environment.

a) High resource consumption when using Cisco routers

Problem: The Cisco XRv/XRv9k virtual routers showed very high CPU and memory usage in the EVE-NG environment, which resulted in noticeable slowness and instability during the simulations. This made it difficult to run larger topologies and to perform repeated migration tests reliably.

Solution: The Cisco virtual routers were replaced by Huawei NE40E devices running VRP R011. These routers are significantly lighter in terms of resource consumption in the lab and still provide stable support for both MPLS and SRv6. This change allowed the project to scale the topology and run multiple migration scenarios without overloading the host machine.

b) Difficulty obtaining SRv6-capable Huawei images

Problem: The project faced difficulties in finding Huawei router images with SRv6 support, as such images are far less publicly available compared to Cisco virtual images commonly used in lab environments. This limited the ability to build a realistic SRv6-enabled topology at the beginning of the work.

Solution: The issue was resolved by reaching out to network engineers via LinkedIn and obtaining an appropriate Huawei VRP R011 image that supports both MPLS and SRv6. This collaboration made it possible to base the entire migration lab on Huawei NE40E devices and to carry out the planned SRv6 scenarios.

c) Limited compute resources on the lab PC

Problem: The lab machine initially had only 8 GB of RAM, which led to noticeable slowness and occasional crashes of some routers when running the full topology in EVE-NG. This constraint made it difficult to keep all devices up and stable during extended migration tests.

Solution: The RAM was upgraded to 16 GB, which significantly improved the stability and responsiveness of the EVE-NG environment. After the upgrade, the virtual routers ran more reliably, allowing the project to execute the complete migration workflow without frequent interruptions.

4.6.9 Lessons Learned

This project reveals several fundamental engineering lessons that go beyond theoretical knowledge:

- Gradual implementation is not just a preferred approach but a practical necessity in production environments; each step must prove itself before moving on to the next.
- Live documentation of challenges, as carried out in this project, is an integral part of engineering work rather than a secondary appendix.
- Pre-migration testing and post-migration verification are not optional steps, but core pillars of any successful migration project.
- Technical limitations of devices are not final obstacles, but engineering challenges that call for creative solutions, such as using the End.DT4 opcode directly instead of binding the VPN instance to the locator.

Conclusion :

In this part, it becomes clear that automation is not only intended to execute a full migration, but can also start with basic steps that support the transition process and make it easier to monitor devices and verify their status. In this project, a connection script and an information-gathering script were used, which made it possible to document the current state of the network and track important data before and during the changes. Although there was not enough time to develop a fully automated migration, this step remains important because it lays the groundwork for any future expansion of automation. As for the challenges and constraints, they showed that the working environment itself and the device software version impose technical limits that must be taken into account when planning the migration. This part therefore reflects that progress in such projects is gradual, and that building simple helper tools can be a necessary first step toward full automation later on. In this way, this work stands as a practical model that combines theoretical understanding, hands-on experimentation, and a realistic handling of time and resource limitations.

GENERAL CONCLUSION

1. Conclusion

In conclusion, this project achieved its primary objective of studying, analyzing, and migrating a service provider network architecture from MPLS L3VPN to SRv6 End.DT4 within a virtual environment based on EVE-NG and Huawei NE40E devices. The practical results demonstrated that SRv6 is not merely an alternative to MPLS, but a more streamlined and flexible networking model that relies natively on IPv6 and reduces the number of protocols required in both the control plane and the data plane.

The study also showed that the transition to SRv6 brings clear operational benefits, particularly in configuration simplification, automation readiness, network visibility, and troubleshooting efficiency. Although MPLS retains some advantages, such as lower per-packet header overhead, its operational complexity and dependence on multiple protocols make it less adaptable than SRv6 in modern network environments.

Through a series of practical tests, including reachability verification, session establishment, routing validation, traceroute analysis, and performance measurements, the project confirmed that SRv6 can deliver performance comparable to MPLS while offering greater path transparency and better integration with automation. In addition, the limitations identified in the Huawei VRP R011 release highlighted that SRv6 can be effectively deployed in practice, even if some advanced features remain unsupported in the tested version.

Overall, this work confirms that SRv6 represents a promising evolution for service provider networks, especially in contexts where operational simplicity, scalability, automation, and readiness for future network transformation are essential.

2. Future work

- Upgrading to VRP V800R021 in order to obtain full support for SRv6 TE Policy, SRv6 Micro-SID, and End.DT46.
- Developing the automation framework to include Ansible Playbooks as a second layer on top of Python.
- Adding a scenario of complete MPLS removal (undo mpls ldp) after verifying that the migration has been fully completed.
- Applying SRv6 Fast Reroute to test fast recovery capabilities in case of failures.
- Expanding the topology to include a Route Reflector and several P devices to emulate a larger ISP network.

These future directions open the way for further developments, both in research and in industrial applications, and can form a solid foundation for advanced projects in networking engineering, cybersecurity, and communications.

References

- [1] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," IETF, RFC 3031, Jan. 2001. Accessed on: May, 2026. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc3031>
- [2] Guru99, "What is IP Routing? Types, Routing Table, Protocols, Commands," accessed May 2026. [Online]. Available: <https://www.guru99.com/ar/ip-routing.html>
- [3] IBM, "Interior and exterior routing gateways," IBM Documentation, accessed May 2026. [Online]. Available: <https://www.ibm.com/docs/en/aix/7.2.0?topic=gateways-interior-exterior-routing>
- [4] C. Hopps, "Routing IPv6 with IS-IS," IETF, RFC 5308, Oct. 2008. Accessed on: May, 2026. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc5308>
- [5] Wikipedia, "Multiprotocol Label Switching," accessed May 2026. [Online]. Available: https://en.wikipedia.org/wiki/Multiprotocol_Label_Switching
- [6] Cisco, "MPLS Beginner's FAQ," Cisco Support Documentation, 2025. [Online]. Available: https://www.cisco.com/c/ar_ae/support/docs/multiprotocol-label-switching-mpls/mppls/4649-mpls-faq-4649.html
- [7] RFC 3013 – Recommended Internet Service Provider Security Services and Procedures Killalea, T., "Recommended Internet Service Provider Security Services and Procedures," RFC 3013, November 2000.
- [8] International Telecommunication Union. "Quality of Service (QoS) Regulation." ITU-D, 2024. URL: <https://www.itu.int/ar/ITU-D/Regulatory-Market/pages/quality-of-service-regulation.aspx> itu
- [9] Edgeworth, Bradley, Aaron Foss, and Ramiro Rios. IP Routing on Cisco IOS, IOS XE, and IOS XR: An Essential Guide to Understanding and Implementing IP Routing Protocols. Cisco Press, 2014
- [10] Filsfils, Clarence, Stefano Previdi, Les Ginsberg, Bruno Decraene, Stephane Litkowski, and Rob Shakir. "Segment Routing Architecture." RFC 8402, Internet Engineering Task Force, July 2018. <https://www.rfc-editor.org/rfc/rfc8402.html>

- [12] C. Filsfils et al., "IPv6 Segment Routing Header (SRH)," RFC 8754, IETF, Mar. 2020. Available: <https://datatracker.ietf.org/doc/rfc8754/>.
- [13] Iyer, Vish. "The Case for SRv6: Simplifying Networks for a Complex Future." CiscoBlogs, January 2025. <https://news-blogs.cisco.com/apjc/2025/01/22/the-case-for-srv6-simplifying-networks-for-a-complex-future/>
- [14] [B] Rosen, E., Viswanathan, A., & Callon, R. "Multiprotocol Label Switching Architecture," RFC 3031, IETF, 2001. <https://datatracker.ietf.org/doc/rfc3031/>
- [15] R. Teunissen, "MPLS Layer 3 VPN Explained," NetworkLessons.com, accessed May 2026. [Online]. Available: <https://networklessons.com/mpls/mpls-layer-3-vpn-explained>
- [16] Spoto, "Dalil takwīn Cisco MPLS (Cisco MPLS Configuration Guide)," CCIEdump Spoto, 2024. [Online]. Available: <https://cciedump.spoto.net/arblog/1717> . [Accessed: 31-May-2026].
- [17] S. Yasukawa and A. Farrel, "An Analysis of Scaling Issues in MPLS-TE Core Networks," IETF, RFC 5439, Feb. 2009. Accessed: May 2026. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc5439>
- [18] Juniper Networks, "What is Segment Routing?," *Juniper Networks Research Topics*, Accessed: May 2026. [Online]. Available: <https://www.juniper.net/fr/fr/research-topics/what-is-segment-routing.html>
- [19] Rosen, E., Viswanathan, A., & Callon, R. Multiprotocol Label Switching Architecture, RFC 3031, IETF, 2001. <https://datatracker.ietf.org/doc/rfc3031/>
- [20] Cisco Systems. MPLS FAQ for Beginners [online]. Available at: Cisco Support. Accessed: 16 May 2026. https://www.cisco.com/c/ar_ae/support/docs/multiprotocol-label-switching-mpls/mpls/118846-config-mpls-00.html
- [21] NetworkLessons. MPLS Labels and Devices [online]. Available at: Accessed: 16 May 2026. <https://networklessons.com/mpls/mpls-labels-and-devices>
- [22] IPCisco. MPLS Label Switching | Push, Swap, Pop [online]. Available at: Accessed: 16 May 2026 <https://ipcisco.com/lesson/mpls-label-switching/>

- [23] RFC 4760, Multiprotocol Extensions for BGP-4. :
<https://datatracker.ietf.org/doc/html/rfc4760>
- [24] Huawei, “What Is SRv6? How Does SRv6 Work?”, Huawei Enterprise Support Website, accessed May 24, 2026.[What Is SRv6? How Does SRv6 Work? - Huawei](#)
- [25] IETF, “SRv6 and MPLS/SR-MPLS Interworking,” Internet-Draft, draft-agrawal-spring-srv6-mpls-interworking, 2024.
- [26] Keysight Technologies, “SRv6—Building Next-Gen Programmable Network Infrastructure,” Keysight Blog, Nov. 15, 2020. [Online]. Available: <https://www.keysight.com/blogs/en/tech/traf-gen/2020/11/16/srv6building-next-gen-programmable-network-infrastructure>
- [27] RFC 9602, “Segment Routing over IPv6 (SRv6) Segment Identifiers in the IPv6 Addressing Architecture,” IETF, July 2024.