



Master Thesis

Verification of Segment Routing Traffic Engineering Using a Network Emulator

Submitted by

Muhammad Ibrahim Muhammad Salem Khidr

Matriculation Number: 757601

Carried out at

**Detecon International GmbH**

Frankfurt, August 2019

Under the Guidance of

**Industrial Supervisor**

Dr. Peter Gröschke

**Academic Supervisors**  
  
Prof. Dr.-Ing. Shun-Ping Chen  
Prof. Dr.-Ing. Herbert Krauß

# **ABSTRACT**

Network operators are facing a challenge to provide modern networks which are able to adapt to the increasing demands coming from the evolution of IP networks (e.g. huge bandwidth requirements, integration of millions of devices and hundreds of services in the cloud). Proposed in the early 2010’s, the Segment Routing (SR) architecture helps facing these challenging demands and it is being adopted and deployed. Traffic Engineering (TE) is used by ISPs to overcome this challenge and Segment Routing (SR) is a new Approach for TE.

Segment Routing (SR) is a flexible, scalable way of doing source routing. The source chooses a path and encodes it in the packet header as an ordered list of so called segments. Segments are identifier for any type of instruction. Each segment is identified by the segment ID (SID) consisting of a flat unsigned 32-bit integer [28]. Segment instruction can be:

• Go to node N using the shortest path

• Go to node N over the shortest path to node M and then follow links

With segment routing, the network no longer needs to maintain a per-application and per-flow state. Instead, it obeys the forwarding instructions provided in the packet.

Segment Routing relies on a small number of extensions to Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) protocols[28][29].

Segment Routing Traffic Engineering (SR-TE) provides a simple, automated, and scalable architecture to engineer traffic flows in a network. With SR, the traffic can be directed through certain segments to achieve more efficient routing. With SR-TE, the network no longer needs to maintain a per-application and per-flow state. Instead, it simply obeys the forwarding instructions provided in the packet.

Segment Routing Traffic Engineering (SR-TE) utilizes network bandwidth more effectively than traditional MPLS-TE networks. It uses a single intelligent source and relieves remaining routers from the task of calculating the required path through the network.

The Thesis will present the implementation of SR and SR-TE on a network emulator (GNS3) with different scenarios showing and proving SR benefits. Another goal was to produce clear instructions for setting up the testing environment and running tests in it, which can be used later by others to practice or continue investigating segment routing benefits.

# **Declaration**

I hereby declare that the work in the thesis entitled “**Verification of Segment Routing Traffic Engineering Using a Network Emulator**” has been written by me and it was not previously submitted for a degree or diploma in any university and it does not contain any material previously published by another person except where the reference is made in the text.

Any assistance that I have received during the research and the implementation of this work has been duly acknowledged.

City, Date : ………………………………   
Signature : ………………………………

# **ACKNOWLEDGMENT**

First and foremost, I would like to thank God Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible.

My sincere thanks goes to my university supervisors **Prof. Dr.-Ing. Shun-Ping Chen** and **Prof. Dr.-Ing. Herbert Krauß**, for their kind support and motivation during the course of my research work.

In my journey towards this degree, I have found a teacher, a friend, an inspiration, a role model and a pillar of support in my guide, **Dr. Peter Gröschke**. He has been there providing his heartfelt support and guidance at all times and has given me invaluable guidance, inspiration and suggestions in my quest for knowledge. He has given me all the freedom to pursue my research, while silently and non-obtrusively ensuring that I stay on course and do not deviate from the core of my research. Without his able guidance, this thesis would not have been possible and I shall eternally be grateful to him for his assistance.

I am so grateful to all my colleagues at Detecon with whom I have had the pleasure to work during this thesis. I would especially like to thank **Timmy Schüller** , **Pramadityo Tjondronegoro** and **Kseniya Manasova**.

Special thanks to all my previous teachers who gave me the basics to be able to start such thesis. . I would especially like to thank **Prof. Dr. Ahmed Zaki, Prof. Dr. Nesreen I. Ziedan and Eng. Ahmed Nabil.**

Many Thanks to all friends who always supported me whenever I asked for their support. . I would especially like to thank **Ahmed Dimah, Vahid Tavajjohi,Ali Gomaa ,Mohamed Abdelhady, Sherif Ibrahim, Ahmed Abdelsalam and Amr Moustafa.**

Nobody has been more important to me in the pursuit of this thesis than the members of my family.  
My deceased father, **Ibrahim Khidr**, who was a great inspiration to me, and his love for knowledge was the founding grounds for my love of knowledge. He is always my constant source of inspiration. I miss our interesting moments. He was always keen to know what I was doing and how I was proceeding, I will miss his reaction and his happy face during my graduation.  
  
My mother, **Faten Hassan**, she has been such a source of strength, support and encouragement throughout my journey and I thank her for her patience with me. She had such faith in me even during the times when I felt exhausted from the work that I had to do.

I am thankful to my wife, **Rawda Salah** ,for her unconditional love, support and patience and motivating me to write my Master thesis in the present form. It was her love that raised me up again when I got weary. Thank you for being with me and for your appreciated sacrifices.

My Brother, **Ahmed Khidr**, and My Sister, **Asmaa Khidr**, truly deserves my deepest appreciation for their continuous support for me. Their much needed encouragement, love, relentless patience, and sacrifices has been a pillar of strength for me during the time-consuming process of this thesis.

# **TABLE OF CONTENT**

Contents

[**ABSTRACT** 2](#_Toc17966533)

[**Declaration** 3](#_Toc17966534)

[**ACKNOWLEDGMENT** 4](#_Toc17966535)

[**TABLE OF CONTENT** 6](#_Toc17966536)

[**LIST OF FIGURES** 8](#_Toc17966537)

[**LIST OF Tables** 9](#_Toc17966538)

[**ABBREVIATIONS** 10](#_Toc17966539)

[**INTRODUCTION** 11](#_Toc17966540)

[**Chapter 1: Technology Overview** 12](#_Toc17966541)

[1.1 Source Routing 12](#_Toc17966542)

[1.2 Segment Routing 13](#_Toc17966543)

[1.3 How Segment Routing Works 14](#_Toc17966544)

[1.4 Market Needs 16](#_Toc17966545)

[1.5 Target Audience for Segment Routing 17](#_Toc17966546)

[1.6 Segment Routing Concepts 17](#_Toc17966547)

[1.7 Routing operation 19](#_Toc17966548)

[1.8 Different forwarding scenarios for Segment Routing 19](#_Toc17966549)

[1.9 Segment Routing Traffic Engineering 21](#_Toc17966550)

[1.10 Fast Reroute 22](#_Toc17966551)

[1.11 Segment Routing vs. MPLS 22](#_Toc17966552)

[**Chapter 2: Lab Setup** 24](#_Toc17966553)

[2.1 Network Emulator 24](#_Toc17966554)

[2.1.1 GNS3 24](#_Toc17966555)

[2.1.2 VMware Workstation Player 26](#_Toc17966556)

[2.2 Router operating system (Cisco XRv 6.1.3) 27](#_Toc17966557)

[2.3 Network analysis 28](#_Toc17966558)

[2.3.1 Wireshark 28](#_Toc17966559)

[2.3.2 SecureCRT 28](#_Toc17966560)

[2.3.3 Ostinato traffic generator 29](#_Toc17966561)

[2.4 Installation 29](#_Toc17966562)

[**Chapter 3: Implementation** 31](#_Toc17966563)

[3.1 Segment routing implementation with OSPF 31](#_Toc17966564)

[3.1.1 Implementing network diagram on GNS3 31](#_Toc17966565)

[3.1.2 Network Configuration 36](#_Toc17966566)

[**Chapter 4: Testing and Results** 50](#_Toc17966567)

[4.1 Segment routing ECMP capability 50](#_Toc17966568)

[4.2 Configuring Segment Routing with ISIS and OSPF 53](#_Toc17966569)

[4.3 Configuring TE policies and steer the traffic into different routes 53](#_Toc17966570)

[4.3.1 Steering Traffic into using dynamic SR-TE Policy 54](#_Toc17966571)

[4.3.2 Steering Traffic into using explicit SR-TE Policy 55](#_Toc17966572)

[4.3.3 Steering Traffic using explicit SR-TE Policy using Node SID and adj-SID 56](#_Toc17966573)

[4.5 Inter-domain TE tunnels with binding segment 58](#_Toc17966574)

[**Chapter 5: Conclusion and future work** 62](#_Toc17966575)

[5.1 Conclusion 62](#_Toc17966576)

[5.2 Future work 63](#_Toc17966577)

[**REFERENCES** 64](#_Toc17966578)

# **LIST OF FIGURES**

[Figure 1 Node-SID forwarding [13] 20](#_Toc17964616)

[Figure 2 Adjacency-SID forwarding 20](#_Toc17964617)

[Figure 3 Cisco IOS XRv Router Virtual Form Factor [56] 27](#_Toc17964618)

[Figure 4 GNS server type selection 29](#_Toc17964619)

[Figure 5 GNS virtual machine 31](#_Toc17964620)

[Figure 6 adding new appliance to GNS3 32](#_Toc17964621)

[Figure 7 Choosing Qemu virtual machine 32](#_Toc17964622)

[Figure 8 Choosing virtual machine serve type 33](#_Toc17964623)

[Figure 9 naming the new appliance 33](#_Toc17964624)

[Figure 10 allocate enough memory for the virtual machine 34](#_Toc17964625)

[Figure 11 allocating the virtual machine image 34](#_Toc17964626)

[Figure 12 the new appliance added successfully to the list 35](#_Toc17964627)

[Figure 13 Network Topology 35](#_Toc17964628)

[Figure 14 Routing Table (OSPF) 43](#_Toc17964629)

[Figure 15 Topology with Traffic generator, Clients, and Servers for testing 50](#_Toc17964630)

[Figure 16 MPLS forwarding Label towards PE-02 51](#_Toc17964631)

[Figure 17 Routes in CEF table for destination 200.200.200.200 51](#_Toc17964632)

[Figure 18 MPLS forwarding table after traffic generated 51](#_Toc17964633)

[Figure 19 interface 0/0/0/1 traffic monitoring using Wireshark 52](#_Toc17964634)

[Figure 20 interface 0/0/0/4 traffic monitoring using Wireshark 52](#_Toc17964635)

[Figure 21 Network topology for traffic engineering testing 53](#_Toc17964636)

[Figure 22 Head-end calculations results 54](#_Toc17964637)

[Figure 23 Traceroute from PE-01 to PE-02 54](#_Toc17964638)

[Figure 24 Monitoring Link between PE-01 and P-01 using Wireshark 55](#_Toc17964639)

[Figure 25 Explicit path List 55](#_Toc17964640)

[Figure 26 Router PE-01 uses the explicit List to reach router PE-02 56](#_Toc17964641)

[Figure 27 Packet Sample showing the use of the explicit list 56](#_Toc17964642)

[Figure 28 Explicit path List with Node SID and Adj-SID 57](#_Toc17964643)

[Figure 29 PE-01 is using node SID and adj-SID to reach PE-02 57](#_Toc17964644)

[Figure 30 Node SID and adj-SID imposed in the packet going from PE-01 to PE-02 57](#_Toc17964645)

[Figure 31 Two separate routing domains topology 58](#_Toc17964646)

[Figure 32 Border router tunnel details towards PE-02 59](#_Toc17964647)

[Figure 33 Border router tunnel details towards PE-01 60](#_Toc17964648)

[Figure 34 The use of binding-SID between PE-01 and PE-02 61](#_Toc17964649)

# **LIST OF Tables**

[Table 1 Segment Routing Vs. MPLS[13] 22](#_Toc17966579)

[Table 2 GNS3 recommended requirements 26](#_Toc17966580)

[Table 3 VMware recommended requirements 27](#_Toc17966581)

[Table 4 Enable segment routing MPLS and MPLS forwarding in OSPF[23] 45](#_Toc17966582)

[Table 5 Configure prefix segment identifier (SID)[23] 46](#_Toc17966583)

[Table 6 Configure a Local Dynamic SR-TE Policy [23] 47](#_Toc17966584)

[Table 7 Configure an Explicit SR-TE Policy [23] 49](#_Toc17966585)

# **ABBREVIATIONS**

SR Segment Routing

TE Traffic Engineering

ISP Internet Service Provider

SID Segment ID

RIB Routing Information Base

FIB Forwarding Information Base

SRB Source-Route Bridging

SDN Software Defined network

SD-WAN Software Defined Wide Area Network

SSRR Strict Source and Record Routing

LSRR Loose Source and Recorded Routing

VPN Virtual private Network

VPLS Virtual Private LAN Service

VPWS Virtual Private Wire Service

EVPN Ethernet Virtual private Network

FRR Fast Re-Route

LDP Label Distribution Protocol

RSVP-TE Resource Reservation Protocol - Traffic Engineering

SID Segment Identifier

LSP Label switched Path

SR-TE Segment Routing for Traffic Engineering

CEF Cisco Express Forwarding

LSP Label Switched Path

PEC Path Computation Element

AER Application Engineered Routing

OSPF Open Shortest Path First

IS-IS Intermediate System Intermediate System

IETF Internet Engineering Task Force

LFA Loop Free Alternative

TI-LFA Topology Independent Loop Free Alternative

# **INTRODUCTION**

This thesis is an experimental thesis showing the implementation of Segment Routing with OSPF and ISIS and implementing a simple traffic engineering technique using the Segment Routing.

The implementation will be done on a Real Router IOS (IOS XRV) which supports segment routing and uses a network emulator (GNS3) working with VMware workstation.

ISPs use as Interior Gateway Protocols OSPF and ISIS for their core network, so it was important to prove that segment routing can be suitable for both. I will start with implementing segment routing with OSPF then with ISIS and will use segment routing traffic engineering to steer traffic between two different routing domains one is using OSPF and the other is using ISIS

Segment Routing Traffic Engineering (SR-TE) provides simple and scalable method for engineering traffic flows. The source node encodes the path as segments and intermediary nodes forward the packet based on the segment instructions. This enables Segment Routing to scale much greater number of tunnels compared to RSVP as RSVP-TE is a soft-state protocol, which means that protocol messages (refresh messages) must be regularly exchanged between signaling neighbors in order to maintain the state for each LSP that runs between the neighbors which has the potential to be a significant constraint on the scaling of the network.

In my Thesis, I will implement a virtual network of routers. The virtual will be based on Cisco IOS XRV 6.1.3 and running on GNS3 environment. Using this virtual network, I will implement different use cases for SR and SR-TE using OSPF and ISIS, which are essential for the SR to be implemented, observing the traffic and emphasizing SR effects and benefits.

In the first chapter, I will introduce an overview about SR then in the second chapter, I will discuss the lab preparation. After that in the third chapter, I will introduce the implementation and the configuration required for SR and SR-TE. In the fourth chapter, I implement different uses cases showing the benefits of segment routing and discuss the obtained results. Finally, the conclusion and suggested future work that may follow this thesis would be in the fifth chapter.

# **Chapter 1: Technology Overview**

Computing and storage solutions have evolved rapidly during the past three decades; the number of internet-connected devices has also grown rapidly. Over the same period, computer networking core architecture has remained largely unchanged. The traditional networking approaches, where networking devices make independent routing decisions running several distributed protocols, have become a limiting factor in terms of innovation and growth in the industry[49].

A solution to this dilemma is the source routing paradigm, in which a sender of a packet can specify, either partially or completely, a route that the packet should take while travelling through the network. Source routing is the core design principle behind segment routing.

This chapter explains concepts and use cases of Segment Routing. Moreover, one can find analysis of all protocols and technologies necessary for Segment Routing deployment.

## 1.1 Source Routing

Routing is the process of selecting the best path through the network for incoming data flows. Usually, packet is guided by each hope based on the destination address and the limited knowledge about the network around it. Such routing method is called destination based routing and it’s commonly used in traditional IP networks. Once a router receives a packet, the router checks the packet’s destination IP address and consults its routing information base (RIB). Once it finds a suitable IP address match, a router forwards a packet to a proper port and packet is forwarded towards the destination IP address.

There are multiple scenarios in which a node may wish to determine a specific set of nodes that shall be traversed while delivering a packet to its destination, or even impose an explicit path through the network topology for reaching the destination. The strategy of imposing a partial or entire path on a packet is called source routing and can be a powerful mean towards efficient and programmable networks.  
  
In contrast of traditional destination based routing, source routing is a method where a sender of a packet specifies the route that a packet should take through the network. When the packet with source routing travels through the network, a transit router routes a packet by inspecting the path information encoded in packet by the source (router). A source router must be aware of network layout in order to specify the route to the destination. Source routing allows a router to determine a path partially or completely. When a sender determines the exact path that mechanism is called Strict Source and Record Routing (SSRR). This approach is rarely used. A common case of source routing is called Loose Source and Recorded Routing (LSRR), where a sender provides one or more intermediate hops that a packet must visit on its path to the destination.

Source routing idea is not a new idea; the source-route bridging (SRB) algorithm was developed by IBM and proposed to the IEEE 802.5 committee as the means to bridge between all local-area networks (LANs). The IEEE 802.5 committee subsequently adopted SRB into the IEEE 802.5 Token Ring LAN specification. SRBs are so named because they assume that the complete source-to-destination route is placed in all inter-LAN frames sent by the source [39].

Source routing is a useful technique for various reasons. In a nutshell, source routing can be particularly used for tunneling network traffic, offering resiliency enhancing possibilities. In addition, it allows for simplified traffic engineering. The main benefit of source routing is that intermediate nodes do not have to keep route information in RIB because the forwarding steps are specified in the data packet. Source routing enables easier network troubleshooting, enhances overall network performance due to automatic ECMP and other features. Software Defined Network can also be improved when source routing is used in the data plane. Source routing can minimize communication between SDN controller and switches when new flow is set up.

With todays extensive and increasing usage of network infrastructure, many ISPs and business analytics see segment routing as a key technology in the near future for large-scale networks in order to optimize traffic flows and implement traffic policies. Especially with regard to content delivery networks and the associated tremendous amount of data to be exchanged, source routing will more and more become an important mean for an efficient allocation of infrastructure resources[50][51].

## 1.2 Segment Routing

Segment routing is a new source routing mechanism developed by the IETF SPRING (Source Packet Routing In NetworkinG) working group. Segment Routing (SR) is a flexible, scalable way of doing source routing. The source chooses a path and encodes it in the packet header as an ordered list of segments. Segments are identifier for any type of instruction. Each segment is identified by the segment ID (SID).

Segment instruction can be:

• Go to node N using the shortest path

• Go to node N over the shortest path to node M and then follow links Layer 1, Layer 2,  
 and Layer 3

• Apply service S

Segment routing relies on a small number of extensions to Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) protocols. It can operate with an MPLS (Multiprotocol Label Switching) or an IPv6 data plane, and it integrates with the rich multi service capabilities of MPLS, including Layer 3 VPN (L3VPN), Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and Ethernet VPN (EVPN).

One of the key characteristics of segment routing is simplicity:

* From a configuration perspective, the number of lines required to enable segment routing on a router is minimum, usually three lines of configuration.
* From an operational perspective, it simplifies the operation of an MPLS network by making the label value constant across the core of the network.
* From a scale and simplicity perspective, segment routing is especially powerful in the era of SDN with application requirements programming the network behavior and where traffic differentiation and engineering are done at a finer granularity (for example, application-specific).

Segment routing can be directly applied to the Multiprotocol Label Switching (MPLS) architecture with no change in the forwarding plane [5]. Segment routing utilizes the network bandwidth more effectively than traditional MPLS networks and offers lower latency[52]. A segment is encoded as an MPLS label. An ordered list of segments is encoded as a stack of labels. The segment to process is on the top of the stack. The related label is popped from the stack, after the completion of a segment.

Segment routing can be applied to the IPv6 architecture with a new type of routing extension header. A segment is encoded as an IPv6 address. An ordered list of segments is encoded as an ordered list of IPv6 addresses in the routing extension header. The active segment is indicated by the Destination Address (DA) of the packet. The next active segment is indicated by a pointer in the new routing header. The pointer is incremented, after the completion of a segment[5].

Segment Routing provides automatic traffic protection without any topological restrictions. The network protects traffic against link and node failures without requiring additional signaling in the network. Existing IP fast re-route (FRR) technology, in combination with the explicit routing capabilities in Segment Routing guarantees full protection coverage with optimum backup paths. With simple configuration segment routing can protect the network without the need to impose any additional signaling requirements.

## 1.3 How Segment Routing Works

In order to provide stability and scalability, large networks should have as few protocols as possible running in the core, and they should keep state away from the core if possible.

Segment Routing is designed to simplify existing routing techniques in traffic-engineered networks. It enables efficient packet steering through a specific network path, rather than natural shortest path, that packet usually takes within a network. A source node directs incoming packet flow through the network by specifying a list of intermediate destinations that a packet must visit on its way to the final destination. In Segment Routing, labels called segments represent intermediate path points. The main benefit of this technology is its simplicity, easy implementation and scalability [6].

Today’s traffic engineering solutions, such as Resource Reservation Protocol – Traffic Engineering (RSVP-TE) require signaling for each path, and state of each path needs to be present on each node that traffic traverses. Segment routing can implement all these without the need of additional signaling protocol, making its architecture simpler and more scalable. Segment Routing using MPLS data plane, does not require Label Distribution Protocol (LDP) or RSVP-TE. Labels are distributed using Interior Gateway Protocol either Intermediate System-to-Intermediate System (ISIS) or Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). Running fewer protocols inside the network already makes network more stable and scalable. Segment Routing paths are protected with Fast Reroute (FRR) capability, that allows rerouting of traffic in under 50 milliseconds, in case of link or node failure [11][12].

MPLS-TE has drawbacks in terms of scalability, manageability, and uses computationally expensive signaling protocols such as RSVP-TE and LDP. Segment Routing overcomes these drawbacks and enables network service providers to change network behavior dynamically.

Segment Routing provides a tunneling mechanism that enables source routing in IP/MPLS networks. An SR path (SR tunnel) is encoded as a sequential list of sub-paths called segments, which are advertised to the SR domain using extensions to link-state routing protocols such as IS-IS or OSPF. An SR tunnel can contain a single segment that represents the destination node or it can contain a segment list that represents the set of segments that a given tunnel must traverse [26]. The SR tunnel can be established over an IPv4/IPv6 MPLS infrastructure or over an IPv6 infrastructure and is encoded as:

* A single MPLS label or an ordered list of hops represented as a stack of labels (with no change to the MPLS data-plane)
* A single IPv6 address or an ordered list of hops represented by a number of IPv6 addresses contained in an IPv6 extension header

The segments that an ingress router imposes for a particular tunnel act a set of instructions, such as “go to Node M using the shortest path” or “go to Node N using link/node/explicit-route L.”

When MPLS is used to instantiate SR tunnels, the MPLS forwarding plane does not change. Segment Routing uses extensions to the link-state IGP to flood SIDs in the form of MPLS labels. No LDP and/or RSVP control plane is required although it is acceptable to run these in conjunction with SR: because the LDP and RSVP label spaces do not overlap, they do not affect each other [26].

When SR is instantiated over the MPLS data plane, the following actions apply:

* A list of segments is represented as a stack of labels.
* The active segment is the top label.
* The CONTINUE operation is implemented as an MPLS swap operation.
* The NEXT operation is implemented as an MPLS pop operation.
* The PUSH operation is implemented as an MPLS push operation.

Unlike RSVP-TE based LSPs, in which the mid-points hold state, SR requires that only the ingress provider-edge (PE) hold state. For any transit or egress SR routers, any required state information is contained in the segment list [26].

Segment Routing can enable traffic engineering in three possible ways:

* By manually creating Segment Routing Label Switched Paths and explicitly defining route inside the network. This equivalent of MPLS-TE but without extra protocols
* By manually creating Segment Routing tunnels. Path is calculated by Path Computation Element (PCE) and later pushed by SDN Controller onto the network
* Dynamically create Segment Routing tunnels. Path is calculated by PCE using existing network information or SLA, such as delay, bandwidth, metric etc.

In addition, if ECMPs are present in the network, segment-routing traffic-engineering tunnels can use all the paths for load balancing flow across them, whereas if RSVP-TE is used to build a traffic-engineering tunnel, only one path is selected.

Segment Routing (SR) is not a new technology but only recently has it been embraced by all the major network equipment vendors. It is a packet forwarding technology where the source node defines the path for traffic, which is then sent through specific nodes and forwarding paths called segments. An SR path is not dependent on hop-by-hop signaling, Label Distribution Protocol (LDP) or RSVP. Instead, it uses segments for forwarding.

## 1.4 Market Needs

Strict service-level agreements (SLAs) in terms of packet loss, delay, jitter, and available bandwidth became a key business differentiator. Such new requirements pushed network evolution toward architectures that allow for steering traffic with more flexibility. In the 1990s, the MPLS architecture introduced a performant tunneling mechanism and a traffic steering function modeled on the ATM/Frame Relay architecture. The tunneling function was fundamental for the success of the MPLS technology, because it enabled the introduction of IP-/MPLS-based VPNs [15].

The RSVP-TE-based traffic-engineering function of MPLS has not enjoyed the same success and is rarely deployed compared to the VPN features, for three fundamental reasons. First, the poor balancing characteristics of the ATM/Frame Relay model do not fit the true nature of IP, which is based on networks offering abundant ECMPs. To overcome this fundamental incoherence, a notorious number of MPLS RSVP-TE tunnels would need to be replicated, thus elevating the difficulty of managing and monitoring the network. Second, control- and data-plane scalability problems are caused by the state required at each hop along any explicit path. Third, the RSVP-TE deployment model observed until now is based on distributed computation, leading to unpredictable placement of the traffic, non-optimal use of the resources, and slow re-optimization. From this experience, network operators are now requiring an evolution toward flexible, operationally friendly, incrementally deployable, yet scalable and simple network architectures. The emergence of application-centric networking and cloud-based services further highlights the need for such an evolution. Segment routing was created to address the needs described previously, as well as to evolve the network infrastructure to the SDN era [15].

## 1.5 Target Audience for Segment Routing

Segment routing supports use cases with applicability for service providers, over-the-top (OTT) providers, content and web providers, and large enterprises across the WAN, metropolitan-area network (MAN), and data center. Segment routing is defined and applicable for unicast traffic. The application of the source-route concept to IP Multicast is not in the scope of segment routing [15].

## 1.6 Segment Routing Concepts

According to the IETF, a segment is an instruction that node executes on the incoming packet. A segment is a 32-bit identification for either a topological instruction or a service instruction. A segment can be either global or local. The instruction associated with a global segment is recognized and executed by any segment-routing-capable node in the domain. The instruction associated with a local segment is supported by only the specific node that originates it [7][15].

A router in a Segment Routing network is capable of selecting any path to forward traffic, whether it is explicit or Interior Gateway Protocol (IGP) shortest path. Segments represent sub paths that a router can combine to form a complete route to a network destination[16].   
  
Each segment has an identifier (Segment Identifier) that is distributed throughout the network using new IGP extensions[28][29]. The extensions are equally applicable to IPv4 and IPv6 control planes. Unlike the case for traditional MPLS networks, routers in a Segment Router network do not require Label Distribution Protocol (LDP) and Resource Reservation Protocol - Traffic Engineering (RSVP-TE) to allocate or signal their segment identifiers and program their forwarding information.

Each router (node) and each link (adjacency) has an associated segment identifier (SID). Node segment

identifiers are globally unique and represent the shortest path to a router as determined by the IGP. The network administrator allocates a node ID to each router from a reserved block. On the other hand, an adjacency segment ID is locally significant and represents a specific adjacency, such as egress interface to a neighboring router. Routers automatically generate adjacency identifiers outside of the reserved block of node IDs.

In an MPLS network, a segment identifier is encoded as an MPLS label stack entry. Segment IDs direct the data along a specified path.

**Global Segment**

Any node in the segment-routing domain understands the instruction associated with a global segment. Any node in the domain installs the related instruction in its Forwarding Information Base (FIB). Global segments fall in a subspace of the segment (or label) space called the Segment Routing Global Block (SRGB). The SRGB is usually defined as the range 16000 to 23999, and all the nodes in a network are allocated the same SRGB; this stipulation is important to fulfill the requirement for operational simplification. Note that the use of a common SRGB in all nodes is not a requirement; a different SRGB at every node can be used, if needed [7][15], but this is discouraged [SR Architecture, RFC8402?].

**Local Segment**

The instruction associated with a local segment is supported by only the node originating it. No other node installs a remote local segment in its FIB. Local segments take a value outside of SRGB range. Since it has only local significance, its value is related only to local router FIB. A router is not aware of local segments of the other routers in a domain. Moreover, the local SID values could be reused within an IGP domain, since a local SID value has local meaning for each single router[13].

**IGP Segment Identifiers – IGP-SIDs**

IGP (OSPF or IS-IS) can allocate segments for different purposes. Link state protocols have an important role in Segment Routing. Global and local segments are distributed throughout the domain using IGP [7][15]. Both Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) support Segment Routing thanks to well-defined protocol extensions.

Segment Identifiers distributed by an IGP have two main types :

• **Prefix SID** Prefix-SID is a segment that refers to a specific network prefix. Prefix-SID is always global within an IGP domain and it refers to the shortest path computed by IGP to the related prefix. A packet that enters an IGP area with an active Prefix-SID will be forwarded along the ECMP-aware shortest path to the prefix. Since a prefix could represent a node or a group of nodes within an IGP domain, Prefix-SIDs are further divided into Node-SIDs and Anycast-SID s:

* + **Node-SID:**  is a Prefix-SID and it refers to a specific node
  + **Anycast-SID:**  is an ID which identifies a set of routers. A packet with Anycast-SID will be forwarded towards the closest node of anycast set.

• **Adjacency SID:** A segment ID that contains an advertising router’s adjacency to a neighbor. An adjacency SID is a link between two routers. Since the adjacency SID is relative to a specific router, it is locally unique.

**Binding segment**

Binding Segment is a segment for traffic engineering, also identified as BSID, which is fundamental for SRTE and brings scalability and service independence to segment routing [50].BSID can be used to steer traffic into theSR-TE policy and across domain borders, creating seamless end-to-end inter-domain SR-TE policies. Each domain controls its local SR-TE policies[40].

A BSID may either be a local or a global SID. If local, a BSID should be allocated from the SRLB. If global, a BSID must be allocated from the SRGB[5].

## 1.7 Routing operation

A source node steers the incoming traffic flow by attaching an ordered list of SIDs to a packet header. The top segment is the first one that will be executed. Once the segment is executed (packet reaches an intermediate destination), next segment is going to be processed and so on. When last segment is executed, a flow either reaches its destination, or it just exits a SR domain and continues to be routed according to destination IP address.

There are three actions that could be performed on segments by SR-capable nodes. They are closely related to operations performed on MPLS labels in MPLS networks.

Segment Routing operations are:

1. **PUSH (MPLS PUSH)** – a segment is pushed on the top of segment stack
2. **NEXT (MPLS POP)** – an active segment is completed, and it is removed from the stack
3. **CONTINUE (MPLS SWAP)** – active segment is not completed yet and it remains active. Naturally, this operation exists only for global segments, since their execution could include multi-hops. Local segments (adjacencies) are executed in a single hop the below figure explains how a packet is forwarded through a SR domain.

Figure 1 explains how a packet is forwarded through a SR domain

## 1.8 Different forwarding scenarios for Segment Routing

* **Forwarding using Node-SID :**

As aforementioned Node-SID refers to a specific node. As shown in *figure 1,* 16004 is a Node-SID of the R4. R1 wants to send a packet to R4 and it pushes the Node-SID on top of the packet’s header. Since the shortest path to the R4 is through R2 and R3, R4 forwards packet to R2. When the packet arrives to R2, the router checks its FIB and passes the packet towards R3. Since R3 is not the destination of the label 16004, R2 does not remove the label. Since R3 is the last router towards destination, it passes the packet to R4 and removes the label out of stack.

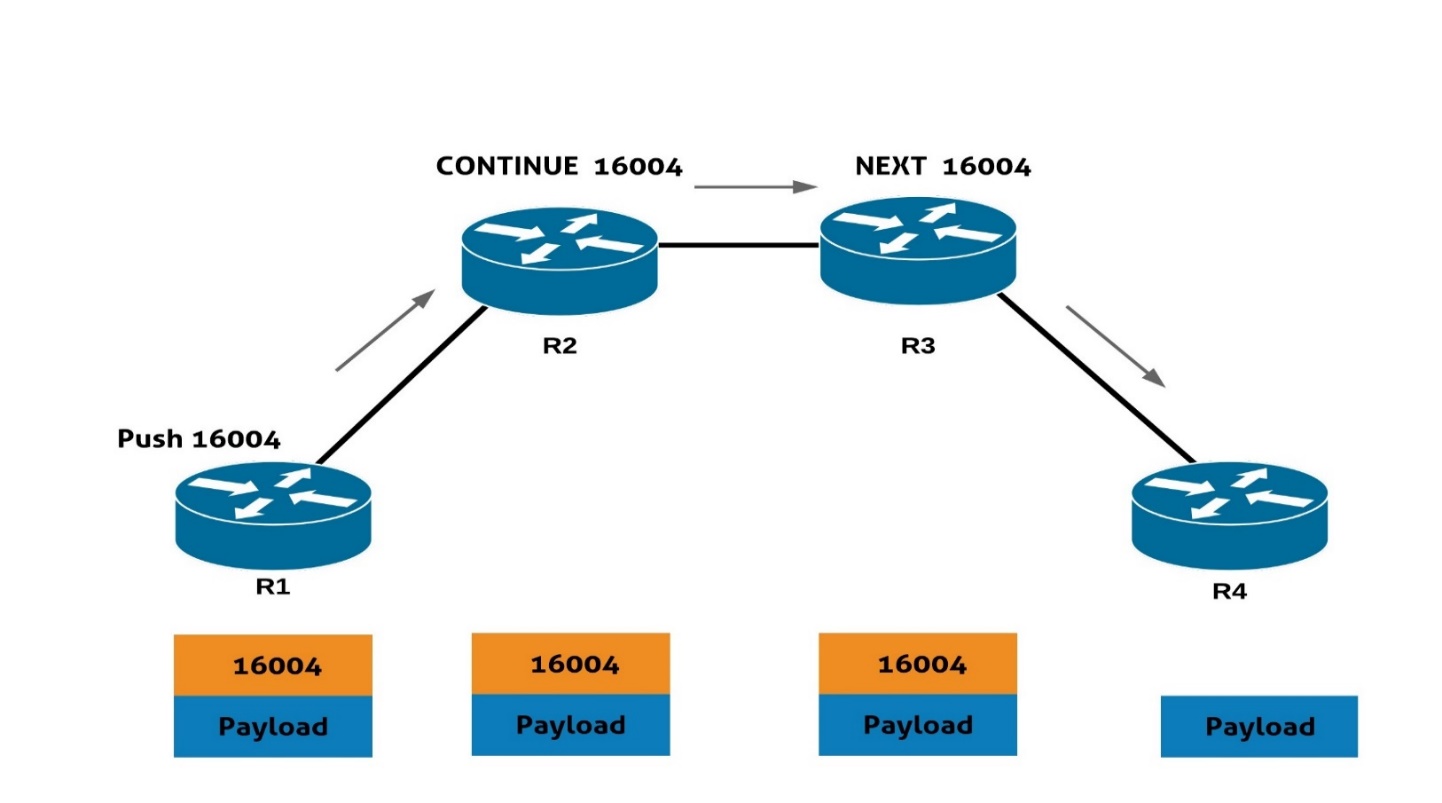


Figure 1 Node-SID forwarding [13]

* **Forwarding using Adjacency-SID :**

As aforementioned Adj-SID is IGP-SID that points on a specific link that belongs to the same IGP domain. Adj-SID has local significance, which means that a router maintains Adj-SIDs only for its neighbors.   
Since Adjacency SID’s has local significance, they do not have to be unique in the SR domain. Adj-SID is very useful in case steering traffic flow through a specific interface. Figure 2 illustrates how Adjacency-SID forwarding work.  
Let us assume the R4. Adj-SIDs are assigned automatically for its three interfaces 24001, 24002 and 24003. If one wants to use a link between R4 and R5 it is enough just to push the local label (24002) and packet will be forwarded to the next hop.

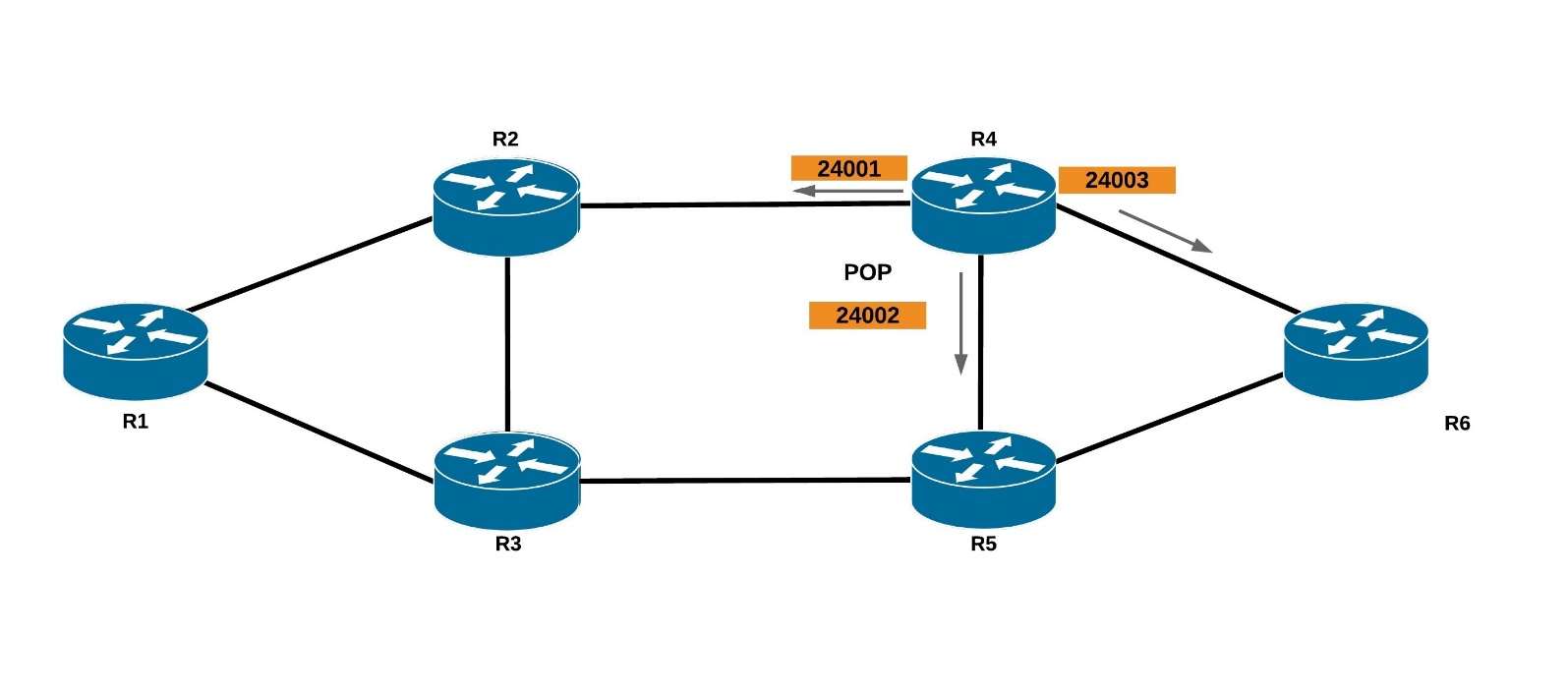


Figure 2 Adjacency-SID forwarding

## 1.9 Segment Routing Traffic Engineering

Traffic engineering (TE) is a mechanism for controlling the path that the traffic takes through the network to the destination. On plain IP network, traffic is by default routed based on the IGP calculated best path. The path that the traffic takes can be influenced by changing metric values on individual links, but this affects all traffic transiting the link and can lead to unexpected results.

MPLS-TE enables establishment of LSPs that take specific path across the network. These paths can be specified manually or calculated dynamically based on constraints, such as available bandwidth, link attributes or path disjointness. Defining constraints on LSPs enable network operators to steer different types of traffic through different paths based on the traffic bandwidth/latency, redundancy or other business requirements[53][54][56].

RSVP has been traditionally used as a control protocol for signaling the MPLS-TE LSPs. RSVP however has issues related to complexity of the protocol and scalability. Each node on the network has to maintain state for each LSP traversing through it. The LSP state also has to be periodically refreshed. This becomes a problem when the number of LSPs grows to thousands. Another issue with RSVP is that it doesn’t scale easily across IGP domains. End-to-end inter-domain LSP requires stitching or nesting of multiple tunnels. The configuration can also become complex.

MPLS-TE, however, establishes a logical full mesh between all edge points in the network, and this is where the scaling problems arise since the structure of the network tends to focus a large number of LSPs within the core of the network [17].

Another practical concern for the scalability of large MPLS-TE networks is the ability to manage the network. This may be constrained by the available tools, the practicality of managing large numbers of LSPs, and the management protocols in use [17].

With segment routing addresses multiple RSVP scalability problems that mentioned before can be solved. Unlike RSVP, SR doesn’t need to maintain state for LSPs in the transit routers. This is because the forwarding is based on the SID list on the individual packets. The head-end router just pushes the SID list to packets and the transit routers forward them according to the segment instructions. The state is held on the packets instead of devices.

Segment routing for traffic engineering (SR-TE) takes place through a tunnel between a source and destination pair. Segment routing for traffic engineering uses the concept of source routing, where the source calculates the path and encodes it in the packet header as a segment. Each segment is an end-to-end path from the source to the destination, and instructs the routers in the provider core network to follow the specified path instead of the shortest path calculated by the IGP. The destination is unaware of the presence of the tunnel.

With segment routing for traffic engineering, the network no longer needs to maintain a per-application and per-flow state. Instead, it simply obeys the forwarding instructions provided in the packet. SR-TE utilizes network bandwidth more effectively than traditional MPLS-TE networks by using ECMP at every segment level as this is implemented automatically, without explicit configuration in the nodes. It uses a single intelligent source and relieves remaining routers from the task of calculating the required path through the network.

## 1.10 Fast Reroute

Segment Routing supports services with tight SLA. To do so, Topology Independent Loop-Free Alternate (TI-LFA) is used within Segment Routing network. TI-LFA guarantees 50 msec of convergence time in any IGP network. TI-LFA is easy to implement because for the protection path it uses one that is automatically pre-computed by IGP. As a protection path, it uses post convergence path, which is the optimum path in case of primary path failure [15][20][55].

Post-convergence path is typically planned by network architects to support traffic rerouting in the case of failure. If failure happens in SR domain, the only node that keeps state is one that suffered from failure – it reroutes packets by attaching backup segments.

## 1.11 Segment Routing vs. MPLS

The following table presents the short comparison between Segment Routing and MPLS[15][21].

|  |  |  |
| --- | --- | --- |
| **Technology Feature** | **Segment routing** | **MPLS** |
| Label Signaling | IGP | LDP + RSVP-TE |
| IGP/LDP SYNC | Not required | Required |
| Fast Reroute FRR 50ms | IGP | IGP + RSVP-TE |
| TI-LFA | Supported | Not Supported |
| Extra states for FRR | No | Yes |
| Optimum backup | Yes | No |
| ECMP capability | Inbuilt | No |
| SDN support | Yes | No |
| Routing type | Source based | Destination based |
| Scalability | High | Low |
| Operation and Troubleshooting | Low | High |

Table 1 Segment Routing Vs. MPLS[13]

In MPLS Label signaling and resource reservation is done with the help of other signaling protocols, LDP and RSVP-TE. However, while using segment routing IGP will be able to distribute the labels within the domain and no need for any signaling protocols, which is one of the major benefits of segment routing in terms of bandwidth and simplicity.

Furthermore, LDP has many drawbacks regarding synchronization after link failure. LDP has to synchronize with the IGP which will recalculate the new best path after the failure and LSP will need some time to be stable again which may cause in some cases packet losses. In segment routing only IGP is used and no need for synchronization with other protocols.

Having a path protection is very important for critical applications. In MPLS in some cases, it is possible to have path protection by calculating both primary and backup path. For both paths, resources must be reserved using RSVP-TE protocol. All routers in both paths must maintain the state of the tunnels. This guarantees QoS and no traffic losses in case of failure and the path is fully protected. However double resource reservation is not efficient in terms of network resource utilization, especially in busy core networks. On the other hand, Segment Routing uses post convergent path that is automatically calculated by IGP upon link failure and guarantees optimal path in new situation. There are no extra states that should be maintained to protect the path. The FRR mechanism in Segment Routing is called Topology Independent Loop-Free Alternate (TI-LFA) and it guarantees <50ms convergence.

Equal Cost Multipath (ECMP) enables traffic balancing among equal cost paths between source and destination. In Segment Routing by default if there are two flows between the same source and destination and multiple routes already exist with the same metric between the source and destination, the two flows will be forwarded through two different routes. ECMP is not supported in MPLS which means network maybe more stable,resilient and efficient in case of using Segment Routing.

Segment routing is highly scalable compared to MPLS. First, it eliminates need for signaling protocols, which simplifies overall architecture and leads to simplified and cheaper hardware. Moreover, number of FIB entries is highly reduced by applying Segment Routing – which is more scalable especially for huge core networks.

At the end, Segment Routing simplifies overall operation and reduces need for network maintenance. The data plane is simplified since there are no signaling protocols. Furthermore, it enables easy operation by making labels constant over the network.

# **Chapter 2: Lab Setup**

This chapter will discuss the following software which were used to build the virtual lab, Monitor and analysis the results

* GNS3 all-in-one (.exe)
* GNS3 VM (.ova)
* VMware Workstation Player
* Wireshark
* Cisco XRV 6.1.3
* SecureCRT
* Ostinato traffic generator
* IPERF 3 Test

## Network Emulator

### 2.1.1 GNS3 [www.gns3.com]

Graphical Network Simulator-3 (shortened to GNS3) is an open source, free software to emulate, configure, test and troubleshoot virtual and real networks. It allows the combination of virtual and real devices, used to simulate complex networks.

GNS3 consists of two software components:

* The GNS3-all-in-one software (GUI)
* The GNS3 virtual machine (VM)

**GNS3-all-in-one:**

This is the client part of GNS3 and is in essence the graphical user interface (GUI). The all-in-one software is installed on the local PC (Windows, MAC, Linux) and then topologies will be created using this software.

**Server options:**

When topologies are created in GNS3 using the all-in-one software GUI client, the devices created need to be hosted and run by a server process, there are a few options for the server part of the software:

* Local GNS3 server
* Local GNS3 VM
* Remote GNS3 VM

For all experiments conducted during this thesis, the local GNS3 server runs locally as a GNS3 VM on the same PC where the GNS3 all-in-one software is installed.

Tthe GNS3 VM can run locally using virtualization software such as VMware Workstation (will be more explained in this chapter) or Virtualbox; or it can be run remotely on a server using VMware ESXi or even in the cloud.

**Advantages and disadvantages of GNS3**

Before choosing GNS3 as the network emulation software to build the setup and to run the experiments, the following was considered:

**Advantages:**

* Free software
* Open Source software
* No monthly or yearly license fees
* No limitation on number of devices (only limitation is the user hardware: CPU and memory)
* Supports multiple switching options (ESW16 Etherswitch, IOU/IOL Layer 2 images, VIRL IOSvL2):
* Supports all VIRL images (IOSv, IOSvL2, IOS-XRv, CSR1000v, NX-OSv, ASAv)
* Supports multi-vendor environments
* Can be run with or without hypervisors
* Supports both free and commercial hypervisors (Virtualbox, VMware workstation, VMware player, ESXi, Fusion)
* Downloadable, free, pre-configured and optimized appliances available to simplify deployment
* Native support for Linux without the need for need for additional virtualization software
* Software from multiple vendors freely available
* Large and active community (800,000+ members)

**Disadvantages:**

* Cisco as well as other software images need to be supplied by user (download from Cisco.com, or purchase VIRL license, or copy from physical device).
* Not a self contained package, but requires a local installation of software (GUI).
* GNS3 can be affected by PC’s setup and limitations because of local installation (firewall and security settings, company laptop policies etc.).

**Supported operating systems**

GNS3 supports the following operating systems:

* Windows 7 (64 bit)
* Windows 8 (64 bit)
* Windows 10 (64 bit)
* Windows Server 2012 (64 bit)
* Windows Server 2016 (64 bit)
* Mac OS X Mavericks (version 10.9) and later.
* Linux

Additional platforms that can run the GNS3 VM:

* ESXi
* Bare Metal Cloud based providers such as Packet.net

**Hardware requirements**

For this thesis GNS3 2.1.14 was used and the following are the recommended requirements for a Windows GNS3 environment:

|  |  |
| --- | --- |
| **ITEM** | **REQUIREMENT** |
| Processor | 4 or more Logical cores - AMD-V / RVI Series or Intel VT-X / EPT |
| Virtualization | Virtualization extensions required. You may need to enable this via your computer's BIOS. |
| Memory | 8 GB RAM |
| Storage | Solid-state Drive (SDD)  35 GB available space |
| Additional Notes | Virtualizing devices is processor and memory intensive. More is better but properly configured device trumps RAM and Processing power. |

Table 2 GNS3 recommended requirements

### 2.1.2 VMware Workstation Player [www.vmware.com]

Workstation Player is a desktop application that allows creation, configuration, and running of virtual machines. Computers that run Workstation Player must meet specific hardware and software requirements. Virtual machines that run in Workstation Player support specific devices and provide certain features. For this thesis VMware Workstation 12 build 7535481 was used and the following is the recommended requirements

|  |  |
| --- | --- |
| **ITEM** | **REQUIREMENT** |
| Processor | 64-bit x86 CPU with 1.3 GHz or faster core speed. Multiprocessor systems are supported. |
| Operating system | Windows and Linux are supported. |
| Memory | The minimum memory required on the host system is 1 GB. 2 GB and above is recommended. |
| Storage | At least 1 GB free disk space is recommended for each guest operating system  and the application software used with it.  For installation, approximately 200 MB free disk space is required on Linux and  250 MB free disk space is required on Windows (the installer can be deleted after installation). |
| Display | 16-bit or 32-bit display adapter. |

Table 3 VMware recommended requirements

## Router operating system (Cisco XRv 6.1.3)

Cisco IOS XRv Router is a Virtual Machine (VM) based platform running 32-bit Cisco IOS XR software with the QNX microkernel. This VM contains a single Route Processor (RP) with control plane functionality and Line Card (LC) network interfaces with their associated functionality. XRv represents Cisco IOS XR software and operating systems that are running on actual Cisco hardware. It gives the user the possibility to work on Cisco routers without having a hardware router. However, IOS XRv is not complete emulation of any physical cisco router or module[56].

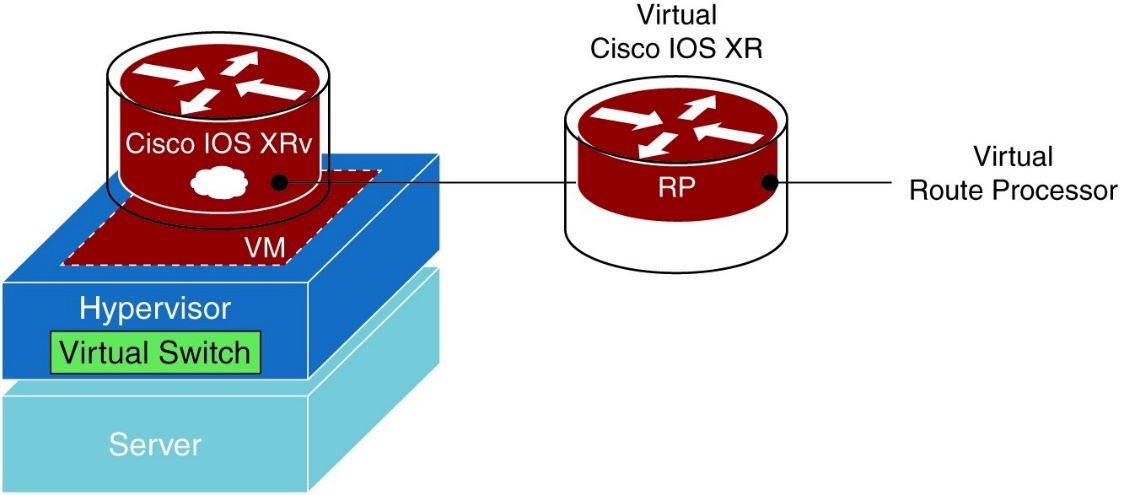


Figure 3 Cisco IOS XRv Router Virtual Form Factor [56]

When running Cisco IOS XRv, the following features and services are available:

* IP features – Cisco XRv supports most of IPv4 and IPv6 services: unicast, multicast, ECMP, load balancing, addressing and ICMP.
* Layer 3 protocols supported - BGPv4. OSPFv2, OSPFv3, IS-IS
* MPLS features – essential protocols supported LDP and RSVP as well as Diffserv Aware TE, MPLS control plane, MPLS forwarding and balancing
* Network Management features – enhanced CLI, XML interface and simple network management protocol support

There were no limitations affecting the results delivered in this thesis, however, number of routers used were limited to the Laptop hardware capabilities which also limited the used IOS version to 6.1.3 not a higher one with more segment routing features.

## Network analysis

### 2.3.1 Wireshark

Wireshark is a widely-used network protocol analyzer. It allows to see what’s happening on the network at a microscopic level. It has a rich feature set which includes the following:

* Deep inspection of hundreds of protocols.
* Live capture and offline analysis
* Multi-platform: Runs on Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, and many others.
* Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
* A configurable display filter.
* Read/write many different capture file formats which allows to read different formats for import and further analysis.
* Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform).
* Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2.
* Output can be exported to XML, PostScript®, CSV, or plain text.

### 2.3.2 SecureCRT

SecureCRT is a software which provides encrypted login, terminal sessions, and data transfer. SecureCRT is highly customizable and easy to use.

During the thesis, SecureCRT was used to access and configure the routers

### 2.3.3 Ostinato traffic generator

Ostinato is a packet generator and network traffic generator with a GUI. It allows to craft and send packets of several streams with different protocols at different rates. We can think of it as “Wireshark in Reverse“. In some use cases it was needed to generate packets to be sent across the network and Ostinato enabled exactly what I wanted to do.

It is an open source and easily integrated with GNS3

## Installation

First, we need to install the GNS3-all-in-one software and VMware Workstation on the PC. Then, after successfully installing VMware Workstation Player, it is time to create a GNS3 VM in VMware Workstation Player.

1. Start VMware Workstation Player.
2. Click “Open a Virtual Machine”
3. Point to the .ova file for GNS3 VM that downloaded earlier, and click Open.

After successfully importing the GNS3 VM, the next step will be integrating GNS3 with the GNS3 VM. After successful installation, the GNS3 Setup Wizard is displayed when GNS3 starts up for the first time. This provides an easy way to initially configure GNS3 options

I chose the first option as shown in the below figure to allow external IOS installation. And now environment is ready for creating the virtual lab

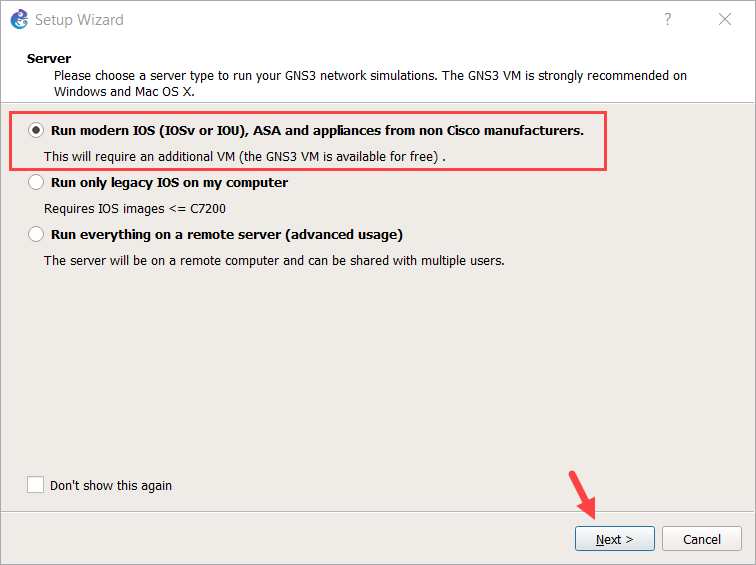


Figure 4 GNS server type selection

Once the environment is ready, we can create the topology in GNS3 by using the GUI to create the virtual router network. in the project area.

While not shown, the different network topologies were created and the virtual connectivity was tested in order to assure proper and reliable operation.

# **Chapter 3: Implementation**

In this chapter, implementation of this thesis work will be explained. Different scenarios were created to test the features of Segment Routing and Segment Routing Traffic Engineering.

All through this chapter, it is assumed that the reader has the knowledge about the syntax used for the configuration of Cisco Routers using IOS XRV.

The short summary of implementation is:

* Segment routing implementation with OSPF.
* Segment routing implementation with IS-IS.
* Segment routing traffic engineering implementation.
* Performance analysis and proving segment routing benefits.

## 3.1 Segment routing implementation with OSPF

### 3.1.1 Implementing network diagram on GNS3

After successful installation of GNS3 Wizard and integrating it with the VM, there two main steps before creating the lab with IOS XRV 6.1.3

* Customizing the virtual machine to have 16GB RAM as 8 routers will be used with 2 GB RAM for each router. The virtual machine will be as shown in the below figure.

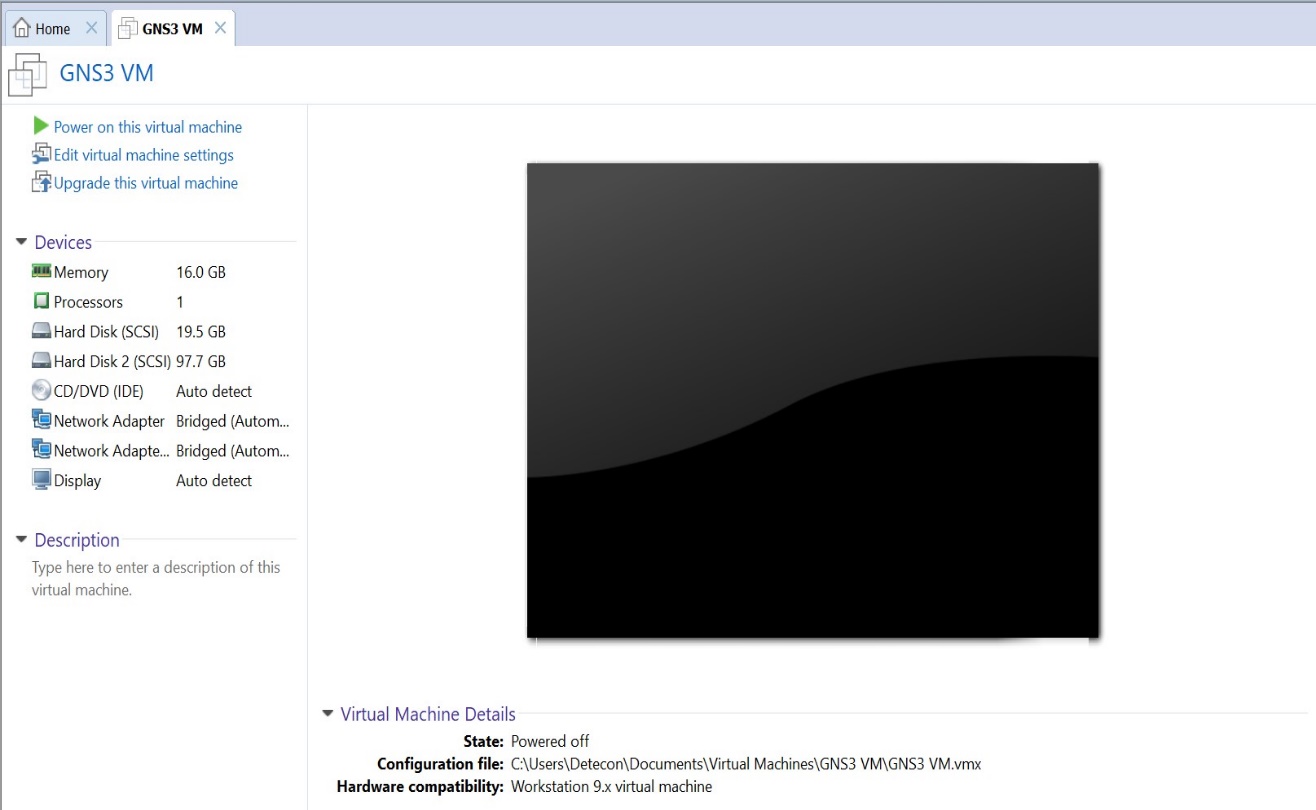


Figure 5 GNS virtual machine

* Importing the IOS XRV appliance into the GNS3 by following the below steps
* In the GNS3 user interface, click the installed appliances icon then click new appliance template as shown in the figure below.

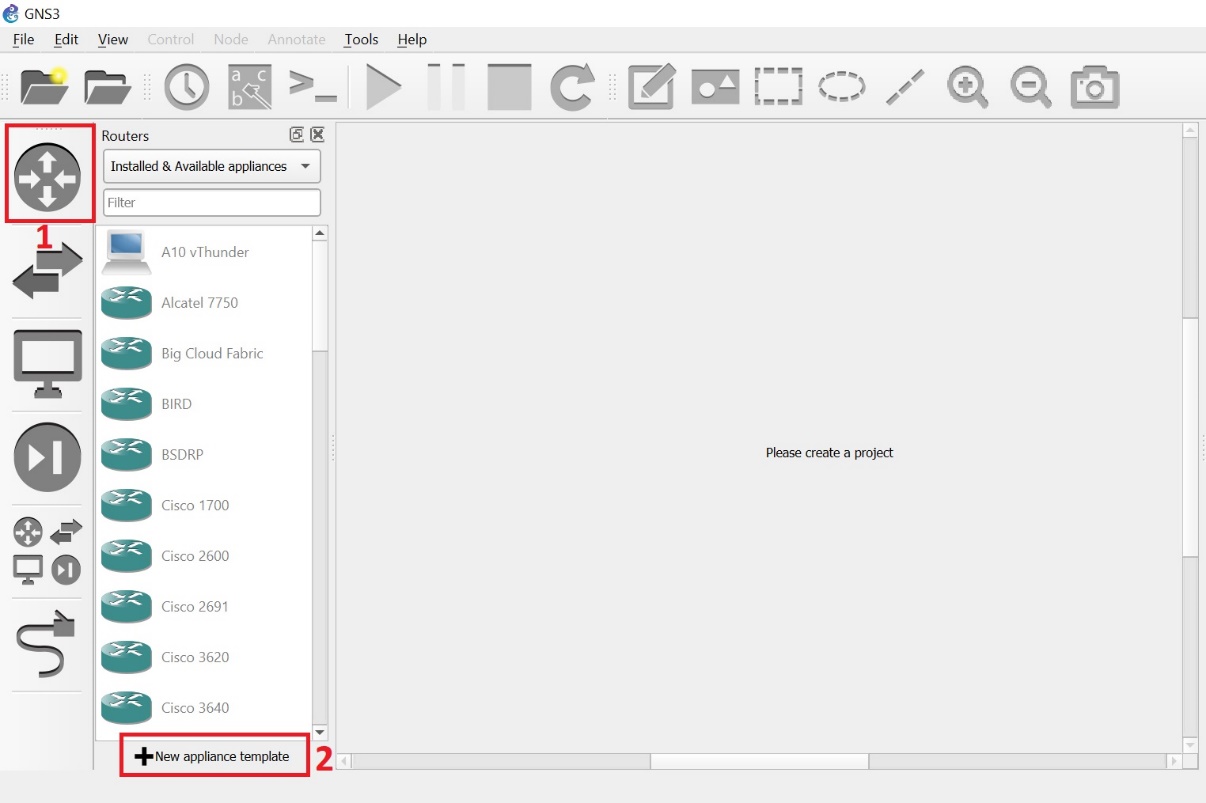


Figure 6 adding new appliance to GNS3

* Choose add a Qemu virtual machine.

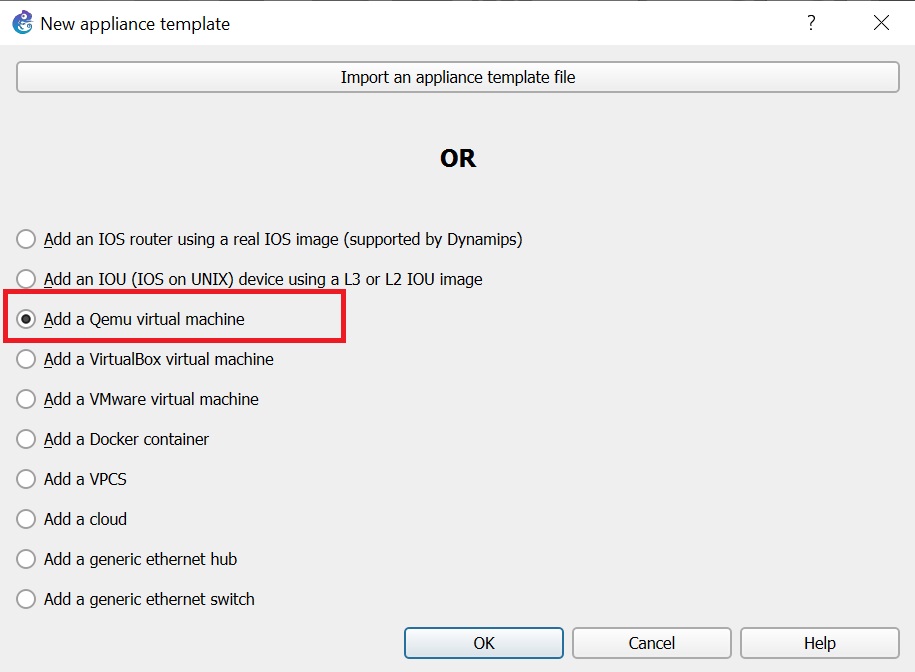


Figure 7 Choosing Qemu virtual machine

* Choose to run the virtual machine on the GNS3 virtual machine.



Figure 8 Choosing virtual machine serve type

* Give the new appliance a name.

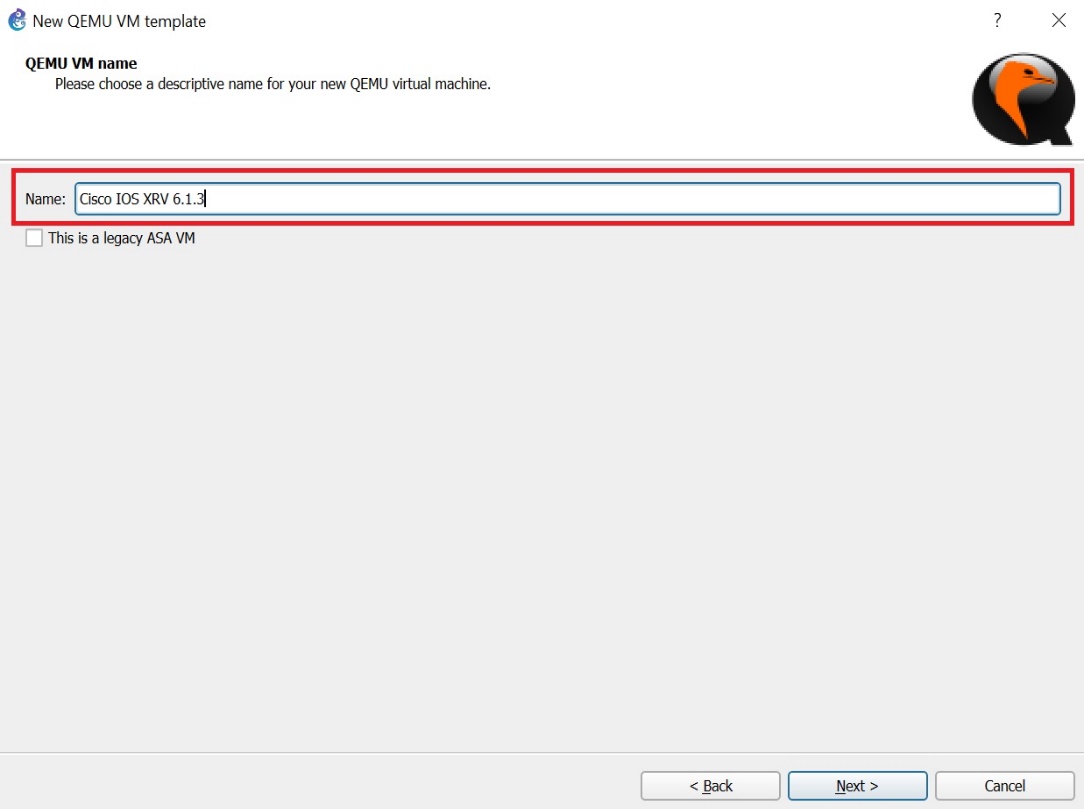


Figure 9 naming the new appliance

* Make sure to have enough memory for the virtual machine ( 2GB is enough for IOS XRV 6.1.3).

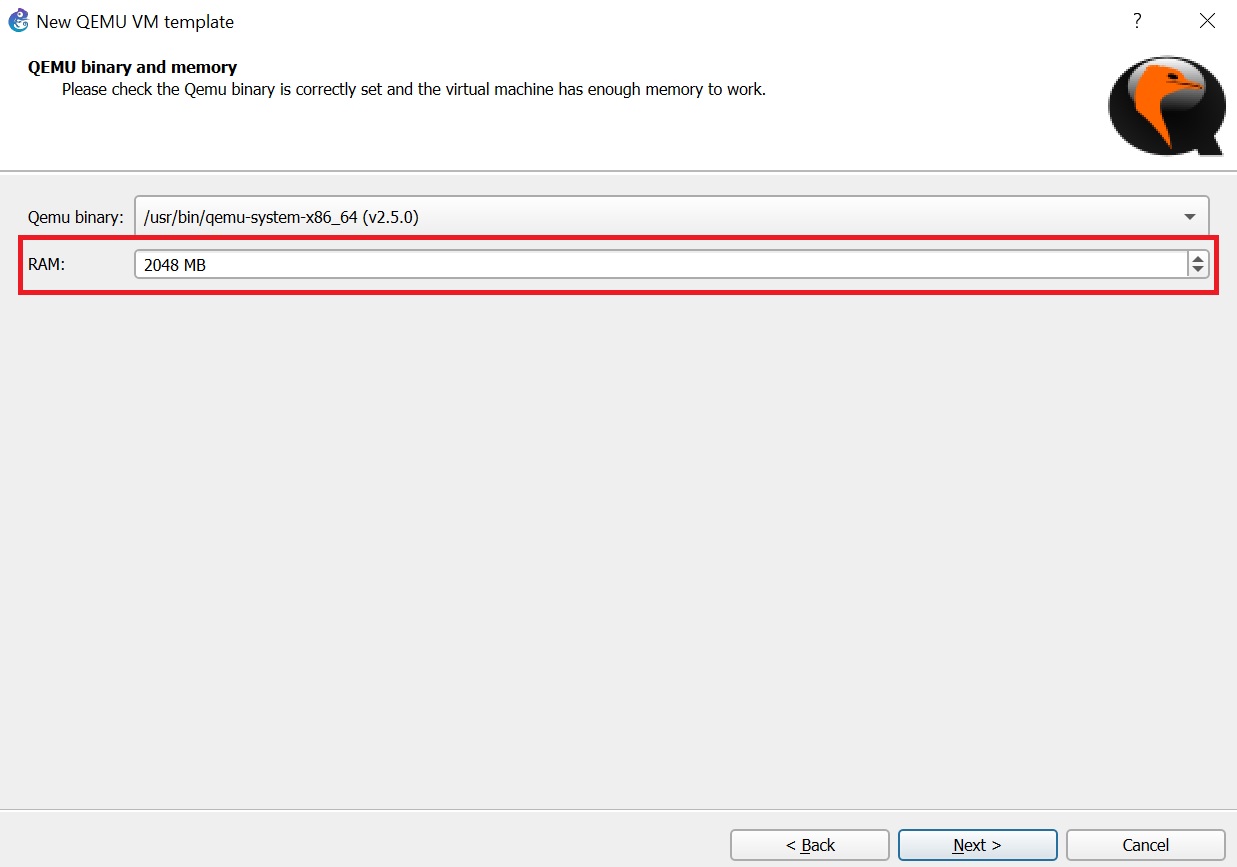


Figure 10 allocate enough memory for the virtual machine

* Choose to add a new image and browse to locate the virtual machine image.

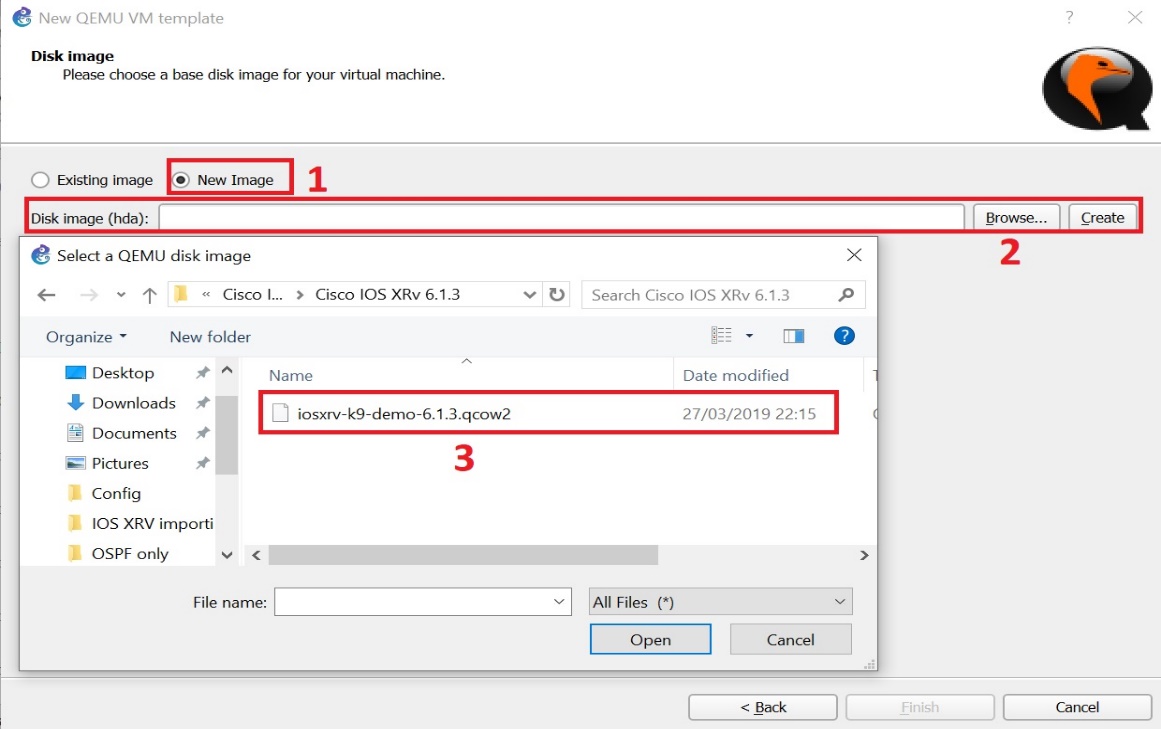


Figure 11 allocating the virtual machine image

* The new appliance should appear in the installed appliances list as shown in the figure below.

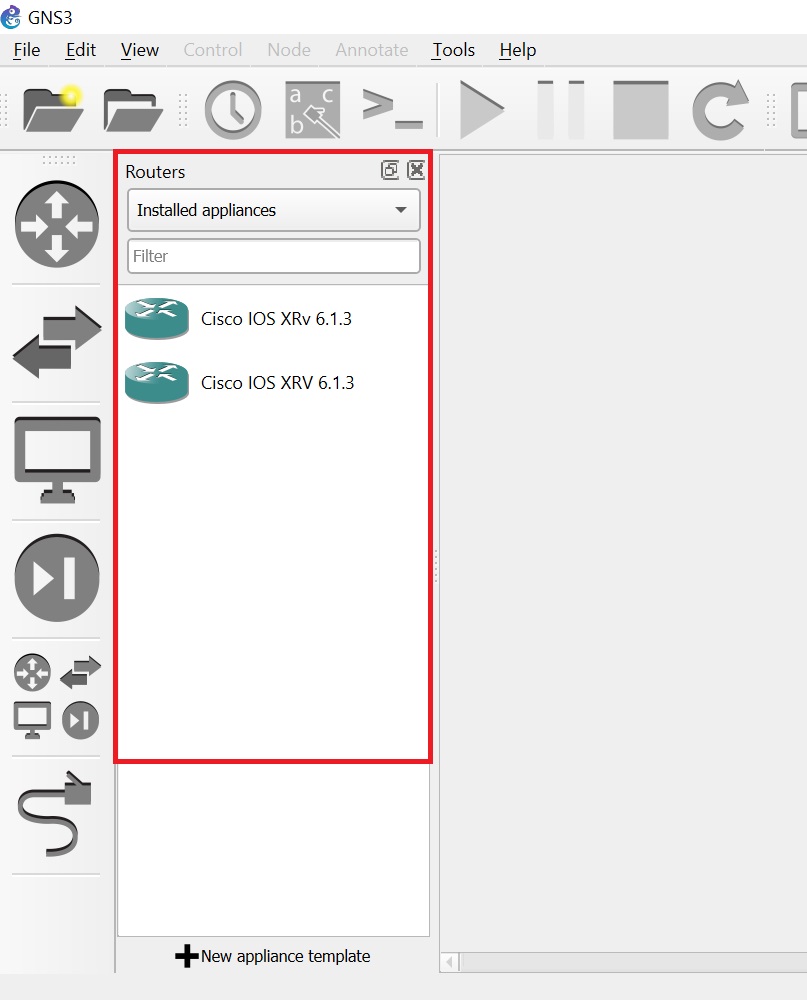


Figure 12 the new appliance added successfully to the list

Now the appliance is successfully imported to the GNS3 and is ready to be used.

* **Network diagram**

After successfully importing the IOS XRV 6.1.3 appliance to the GNS3 we now are able to implement our topology as shown in figure 12

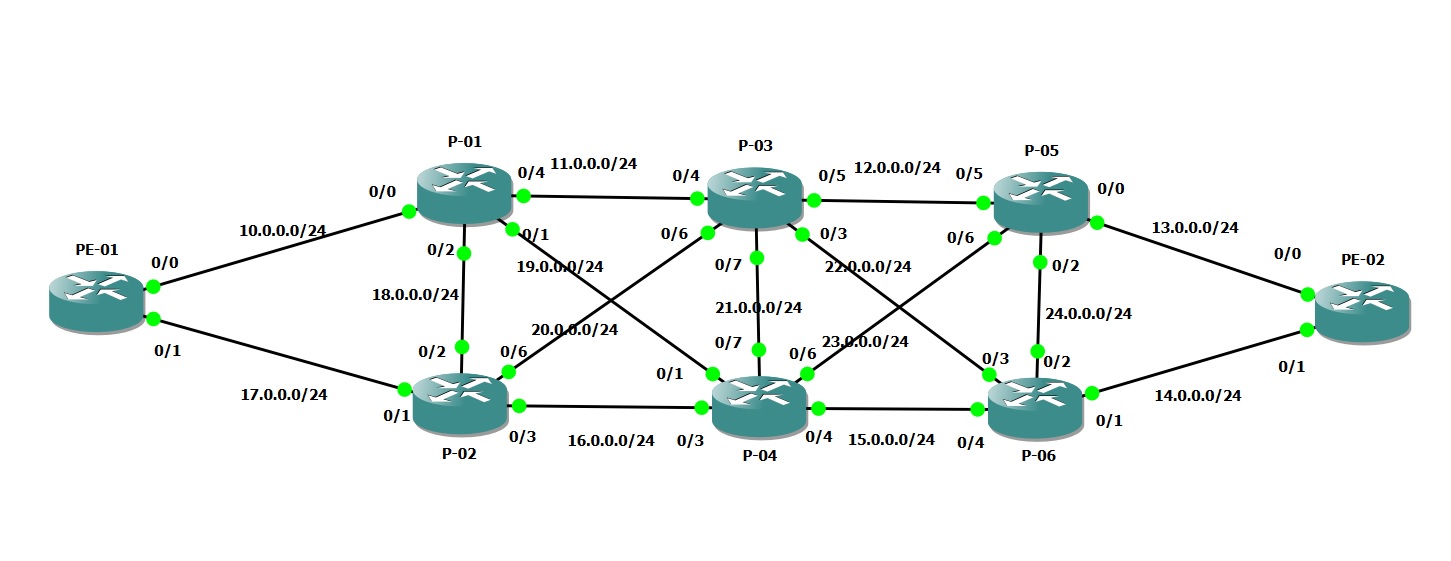


Figure 13 Network Topology

### 3.1.2 Network Configuration

Network is configured through CLI by entering the console of each router. There were three main steps through the configuration process

* Initial Router configuring
* IGP ( OSPF ) configuration
* Segment Routing configuration

#### 3.1.2.1 Initial Router configuration:

The purpose was to assign name for the router and IP addresses for the interfaces and I configured the routers as below:

**Router PE01 :**

hostname PE-01

interface loopback 0

ipv4 address 100.100.100.100/32

interface gigabitEthernet 0/0/0/0

ipv4 address 10.0.0.1/24

ipv4 point-to-point

no shutdown

interface gigabitEthernet 0/0/0/1

ipv4 point-to-point

ipv4 address 17.0.0.1/24

no shutdown

**Router P-01 :**  
hostname P-01

interface loopback 0

ipv4 address 1.1.1.1/32

interface gigabitEthernet 0/0/0/0

ipv4 address 10.0.0.2/24

ipv4 point-to-point

no shutdown

interface gigabitEthernet 0/0/0/2

ipv4 point-to-point

ipv4 address 18.0.0.1/24

no shutdown

interface gigabitEthernet 0/0/0/1

ipv4 point-to-point

ipv4 address 19.0.0.1/24

no shutdown

interface gigabitEthernet 0/0/0/4

ipv4 point-to-point

ipv4 address 11.0.0.1/24

no shutdown

**Router P-02 :**

hostname P-02

interface loopback 0

ipv4 address 2.2.2.2/32

interface gigabitEthernet 0/0/0/1

ipv4 address 17.0.0.2/24

ipv4 point-to-point

no shutdown

interface gigabitEthernet 0/0/0/2

ipv4 point-to-point

ipv4 address 18.0.0.2/24

no shutdown

interface gigabitEthernet 0/0/0/3

ipv4 point-to-point

ipv4 address 16.0.0.1/24

no shutdown

interface gigabitEthernet 0/0/0/6

ipv4 point-to-point

ipv4 address 20.0.0.1/24

no shutdown

**Router P-03 :**

hostname P-03

interface loopback 0

ipv4 address 3.3.3.3/32

interface gigabitEthernet 0/0/0/4

ipv4 address 11.0.0.2/24

ipv4 point-to-point

no shutdown

interface gigabitEthernet 0/0/0/6

ipv4 point-to-point

ipv4 address 20.0.0.2/24

no shutdown

interface gigabitEthernet 0/0/0/7

ipv4 point-to-point

ipv4 address 21.0.0.1/24

no shutdown

interface gigabitEthernet 0/0/0/3

ipv4 point-to-point

ipv4 address 22.0.0.1/24

no shutdown

interface gigabitEthernet 0/0/0/5

ipv4 point-to-point

ipv4 address 12.0.0.1/24

no shutdown

**Router P-04 :**

hostname P-04

interface loopback 0

ipv4 address 4.4.4.4/32

interface gigabitEthernet 0/0/0/3

ipv4 address 16.0.0.2/24

ipv4 point-to-point

no shutdown

interface gigabitEthernet 0/0/0/1

ipv4 point-to-point

ipv4 address 19.0.0.2/24

no shutdown

interface gigabitEthernet 0/0/0/7

ipv4 point-to-point

ipv4 address 21.0.0.2/24

no shutdown

interface gigabitEthernet 0/0/0/6

ipv4 point-to-point

ipv4 address 23.0.0.1/24

no shutdown

interface gigabitEthernet 0/0/0/4

ipv4 point-to-point

ipv4 address 15.0.0.1/24

no shutdown

**Router P-05 :**

hostname P-05

interface loopback 0

ipv4 address 5.5.5.5/32

interface gigabitEthernet 0/0/0/5

ipv4 address 12.0.0.2/24

ipv4 point-to-point

no shutdown

interface gigabitEthernet 0/0/0/6

ipv4 point-to-point

ipv4 address 23.0.0.2/24

no shutdown

interface gigabitEthernet 0/0/0/2

ipv4 point-to-point

ipv4 address 24.0.0.1/24

no shutdown

interface gigabitEthernet 0/0/0/0

ipv4 point-to-point

ipv4 address 13.0.0.1/24

no shutdown

**Router P-06 :**

hostname P-06

interface loopback 0

ipv4 address 6.6.6.6/32

interface gigabitEthernet 0/0/0/4

ipv4 address 15.0.0.2/24

ipv4 point-to-point

no shutdown

interface gigabitEthernet 0/0/0/3

ipv4 point-to-point

ipv4 address 22.0.0.2/24

no shutdown

interface gigabitEthernet 0/0/0/2

ipv4 point-to-point

ipv4 address 24.0.0.2/24

no shutdown

interface gigabitEthernet 0/0/0/1

ipv4 point-to-point

ipv4 address 14.0.0.1/24

no shutdown

**Router PE-02 :**

hostname PE-02

interface loopback 0

ipv4 address 200.200.200.200/32

interface gigabitEthernet 0/0/0/0

ipv4 address 13.0.0.2/24

ipv4 point-to-point

no shutdown

interface gigabitEthernet 0/0/0/1

ipv4 point-to-point

ipv4 address 14.0.0.2/24

no shutdown

interface gigabitEthernet 0/0/0/1

ipv4 point-to-point

ipv4 address 40.0.0.1/24

no shutdown

#### 3.1.2.2 IGP configuration:

After initial configuration for all routers, it is the time for the IGP configuration and with our case now we will configure OSPF on each router.

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). Designed expressly for IP networks, OSPF supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

This section provides the configuration information to enable OSPF.

The OSPF configurations will be as below

**Router PE-01 :**

router ospf 1

address-family ipv4 unicast

area 0

interface Loopback0

interface GigabitEthernet0/0/0/0

network point-to-point

interface GigabitEthernet0/0/0/1

network point-to-point

**Router P-01 :**

router ospf 1

address-family ipv4 unicast

area 0

interface Loopback0

interface GigabitEthernet0/0/0/0

network point-to-point

interface GigabitEthernet0/0/0/1

network point-to-point

interface GigabitEthernet0/0/0/2

network point-to-point

interface GigabitEthernet0/0/0/4

network point-to-point

**Router P-02 :**

router ospf 1

address-family ipv4 unicast

area 0

interface Loopback0

interface GigabitEthernet0/0/0/6

network point-to-point

interface GigabitEthernet0/0/0/1

network point-to-point

interface GigabitEthernet0/0/0/2

network point-to-point

interface GigabitEthernet0/0/0/3

network point-to-point

**Router P-03 :**

router ospf 1

address-family ipv4 unicast

area 0

interface Loopback0

interface GigabitEthernet0/0/0/6

network point-to-point

interface GigabitEthernet0/0/0/4

network point-to-point

interface GigabitEthernet0/0/0/7

network point-to-point

interface GigabitEthernet0/0/0/3

network point-to-point

interface GigabitEthernet0/0/0/5

network point-to-point

**Router P-04 :**

router ospf 1

address-family ipv4 unicast

area 0

interface Loopback0

interface GigabitEthernet0/0/0/6

network point-to-point

interface GigabitEthernet0/0/0/4

network point-to-point

interface GigabitEthernet0/0/0/7

network point-to-point

interface GigabitEthernet0/0/0/3

network point-to-point

interface GigabitEthernet0/0/0/1

network point-to-point

**Router P-05 :**

router ospf 1

address-family ipv4 unicast

area 0

interface Loopback0

interface GigabitEthernet0/0/0/6

network point-to-point

interface GigabitEthernet0/0/0/5

network point-to-point

interface GigabitEthernet0/0/0/0

network point-to-point

interface GigabitEthernet0/0/0/2

network point-to-point

**Router P-06 :**

router ospf 1

address-family ipv4 unicast

area 0

interface Loopback0

interface GigabitEthernet0/0/0/1

network point-to-point

interface GigabitEthernet0/0/0/4

network point-to-point

interface GigabitEthernet0/0/0/2

network point-to-point

interface GigabitEthernet0/0/0/3

network point-to-point

**Router PE-02 :**

router ospf 1

address-family ipv4 unicast

area 0

interface Loopback0

interface GigabitEthernet0/0/0/0

network point-to-point

interface GigabitEthernet0/0/0/1

network point-to-point

interface GigabitEthernet0/0/0/5

network point-to-point

Before moving to the next step and configuring segment routing we can check now OSPF is up and running by checking the routing table for Router PE-01 and see if it can reach all networks in our topology.  
And as shown in the below figure OSPF in up and running.

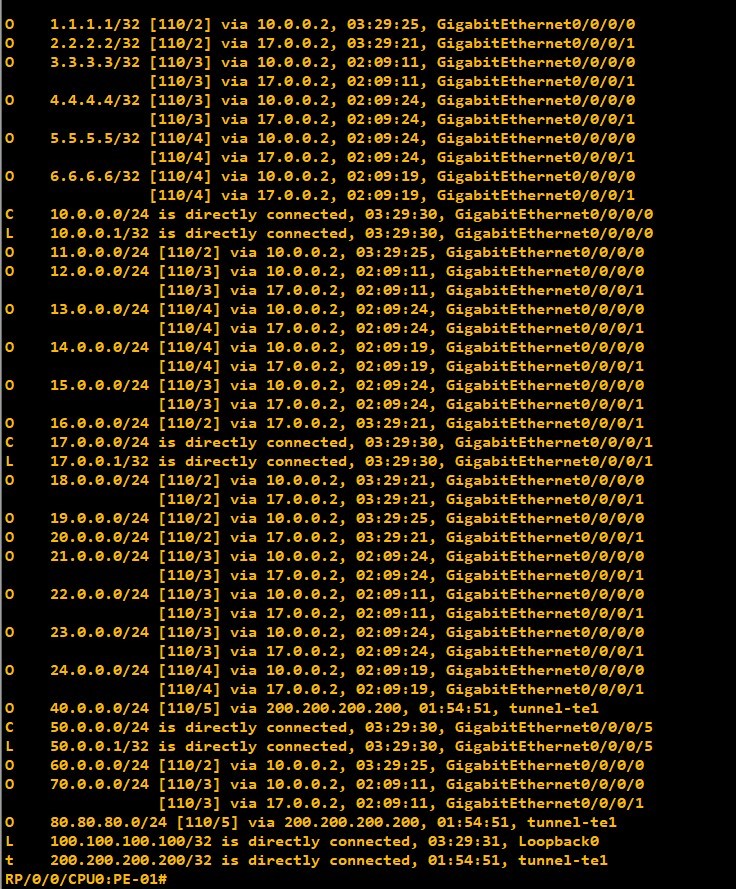
**

Figure 14 Routing Table (OSPF)

#### 3.1.2.3 Segment Routing configuration:

* **Enable segment routing MPLS and MPLS forwarding in OSPF**

This section describes how to enable segment routing MPLS and MPLS forwarding in OSPF. Segment routing can be configured at the instance, area, or interface level.

Segment routing on the OSPF control plane supports the following:

• OSPFv2 control plane

• Multi-area

• IPv4 prefix SIDs for host prefixes on loopback interfaces

• Adjacency SIDs for adjacencies

The below steps summarize how to enable segment routing MPLS and MPLS forwarding in OSPF :  
  
1. configure

2. router ospf process-name

3. segment-routing mpls

4. area 0

5. mpls traffic-eng area

6. mpls traffic-eng router-id interface

7. segment-routing mpls

8. exit

9. mpls traffic-eng

10. commit

And the detailed steps will be as shown in the below table

|  |  |  |
| --- | --- | --- |
| Step | Command or Action | Purpose |
| Step 1 | configure |  |
| Step 2 | router ospf process-name  Example:  RP/0/RSP0/CPU0:PE-01(config)# router ospf 1 | Enables OSPF routing for the specified routing process and places the router in router configuration mode. |
| Step 3 | segment-routing mpls  Example:  RP/0/RSP0/CPU0: PE-01 (config-ospf)#  segment-routing mpls | Enables segment routing using the MPLS data plane on the routing process and all areas and interfaces in the routing process. Enables segment routing forwarding on all interfaces in the routing process and installs the SIDs received by OSPF in the forwarding table. |
| Step 4 | area 0  Example:  RP/0/RSP0/CPU0: PE-01 (config-ospf)# area 0 | Enters area configuration mode. |
| Step 5 | mpls traffic-eng area  Example:  RP/0/RSP0/CPU0: PE-01 (config-ospf-ar)# mpls  traffic-eng area 0 | Enables IGP traffic engineering functionality. |
| Step 6 | mpls traffic-eng router-id interface  Example:  RP/0/RSP0/CPU0: PE-01 (config-ospf-ar)# mpls  traffic-eng router-id Loopback0 | Sets the traffic engineering loopback interface. |
| Step 7 | segment-routing mpls  Example:  RP/0/RSP0/CPU0: PE-01 (config-ospf-ar)#  segment-routing mpls  Step 7 | (Optional) Enables segment routing using the MPLS data plane on the area and all interfaces in the area. Enables segment routing forwarding on all interfaces in the area and installs the SIDs received by OSPF in the forwarding table. |
| Step 8 | exit  Example:  RP/0/RSP0/CPU0: PE-01 (config-ospf-ar)# exit  RP/0/RSP0/CPU0: PE-01 (config-ospf)# exit |  |
| Step 9 | mpls traffic-eng  Example:  RP/0/RSP0/CPU0: PE-01 (config)# mpls  traffic-eng | Enables traffic engineering functionality on the node. The node advertises the traffic-engineering link attributes in IGP, which populates the traffic-engineering database (TED) on the head-end. The SR-TE head-end requires the TED to calculate  and validate the path of the SR-TE policy. |
| Step 10 | commit |  |

Table 4 Enable segment routing MPLS and MPLS forwarding in OSPF[23]

* **Configuring a Prefix-SID on the OSPF-Enabled Loopback Interface**

A prefix SID is associated with an IP prefix. The prefix SID is manually configured from the segment routing global block (SRGB) range of labels. The prefix segment steers the traffic along the shortest path to its

Destination. A node SID is a special type of prefix SID that identifies a specific node. It is configured under the loopback interface with the loopback address of the node as the prefix.

The prefix SID is globally unique within the segment routing domain.

The below steps summarize how to configure prefix segment identifier (SID) index or absolute value on the

OSPF-enabled Loopback interface.

1. configure

2. router ospf process-name

3. area value

4. interface Loopback interface-instance

5. prefix-sid {index SID-index | absolute SID-value } [n-flag-clear ] [ explicit-null ]

6. commit

And the detailed steps will be as shown in the below table

|  |  |  |
| --- | --- | --- |
| Step | Command or Action | Purpose |
| Step 1 | configure |  |
| Step 2 | router ospf *process-name*  Example:  RP/0/RSP0/CPU0:PE-01(config)# router ospf 1 | Enables OSPF routing for the specified routing process and places the router in router configuration mode. |
| Step 3 | area value  Example:  RP/0/RSP0/CPU0: PE-01 (config-ospf)# area | Enters area configuration mode. |
| Step 4 | interface Loopback *interface-instance* Example:  RP/0/RSP0/CPU0:router(config-ospf-ar)#  interface Loopback0 passive | Specifies the loopback interface and instance. |
| Step 5 | prefix-sid {index SID-index | absolute SID-value } [n-flag-clear ] [ explicit-null ] **Example:** RP/0/RSP0/CPU0:router(config-ospf-ar)#  prefix-sid index 100  RP/0/RSP0/CPU0:router(config-ospf-ar)#  prefix-sid absolute 17001 | Configures the prefix-SID index or absolute value for the interface.  Specify index SID-index for each node to create a prefix SID based  on the lower boundary of the SRGB + the index.  Specify absolute SID-value for each node to create a specific prefix  SID within the SRGB.  By default, the n-flag is set on the prefix-SID, indicating that it is  a node SID. For specific prefix-SID (for example, Anycast  prefix-SID), enter the n-flag-clear keyword. OSPF does not set  the N flag in the prefix-SID sub Type Length Value (TLV).  To disable penultimate-hop-popping (PHP) and add an explicit-Null  label, enter the explicit-null keyword. OSPF sets the E flag in  the prefix-SID sub TLV |
| Step 6 | Commit |  |

Table 5 Configure prefix segment identifier (SID)[23]

#### 3.1.2.4 Segment Routing Traffic Engineering Configuration :

This section provides information about segment routing for traffic engineering (SR-TE) policies, how to configure SR-TE policies, and how to steer traffic into an SR-TE policy.

Segment routing for traffic engineering (SR-TE) uses a “policy” to steer traffic through the network. An SR-TE policy path is expressed as a list of segments that specifies the path, called a segment ID (SID) list. Each

Segment is an end-to-end path from the source to the destination, and instructs the routers in the network to follow the specified path instead of the shortest path calculated by the IGP. If a packet is steered into an SR-TE policy, the SID list is pushed on the packet by the head-end. The rest of the network executes the instructions embedded in the SID list.

There are two types of SR-TE policies: dynamic and explicit.

**Local Dynamic SR-TE Policy**

When you configure local dynamic SR-TE, the head-end locally calculates the path to the destination address. Dynamic path calculation results in a list of interface IP addresses that traffic engineering (TE) maps to adj-SID labels. Routes are learned by way of forwarding adjacencies over the TE tunnel.

**Configure Local Dynamic SR-TE Policy**

The below steps summarize how to configure a Local Dynamic SR-TE Policy

1. configure

2. interface tunnel-te tunnel-id

3. ipv4 unnumbered type interface-path-id

4. destination ip-address

5. path-option preference-priority dynamic segment-routing

6. path-protection

7. commit

And the detailed steps will be as the below table

|  |  |  |
| --- | --- | --- |
| Step | Command or Action | Purpose |
| Step 1 | configure |  |
| Step 2 | **interface** **tunnel-te** *tunnel-id* **Example:**  RP/0/RSP0/CPU0:PE-01(config)# interface tunnel-te 1 | Configures the tunnel interface |
| Step 3 | **ipv4 unnumbered** *type interface-path-id*  **Example:**  RP/0/RSP0/CPU0: PE-01(config-if)# ipv4 unnumbered loopback0 | Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type. |
| Step 4 | **destination** *ip-address*  **Example:**  RP/0/RSP0/CPU0:PE-01(config-if)# destination 200.200.200.200 | Assigns a destination address on the new tunnel. |
| Step 5 | **path-option** *preference-priority* **dynamic segment-routing**  **Example:**  RP/0/RSP0/CPU0:PE-01(config-if)# path-option 1 dynamic segment-routing | Sets the path option to dynamic and assigns the path ID. |
| Step 6 | **path-protection**  **Example:**  RP/0/RSP0/CPU0:PE-01(config-if)# path-protection | Enables path protection on the tunnel-te interface. |
| Step 7 | Commit |  |

Table 6 Configure a Local Dynamic SR-TE Policy [23]

**Explicit SR-TE Policy**

An explicit path is a list of IP addresses or labels, each representing a node or link in the explicit path. This feature is enabled through the **explicit-path** command that allows to create an explicit path and enter a configuration sub-mode for specifying the path.

**Configure Explicit SR-TE Policy**

The below steps summarize how to configure an Explicit SR-TE Policy

1. **configure**

2. **explicit-path name** *path-name*

3. **index** *index* {**next-address** *ip-address* | **next-label** *label*}

4. **exit**

5. **interface tunnel-te** *tunnel-id*

6. **ipv4 unnumbered** *type interface-path-id*

7. **destination** *ip-address* **[verbatim]**

8. **path-option** *preference-priority* **explicit name** *path-name* **segment-routing**

9. **commit**

And the detailed steps will be as the below table

|  |  |  |
| --- | --- | --- |
| Step | Command or Action | Purpose |
| Step 1 | configure |  |
| Step 2 | **explicit-path name** *path-name*  **Example:**  RP/0/RSP0/CPU0:PE-01(config)# explicit-path name 01-04-PE02 | Enters a name for the explicit path and enters the explicit path configuration mode. |
| Step 3 | **index** *index* {**next-address** *ip-address* | **next-label** *label*}  **Example:**  RP/0/RSP0/CPU0: PE-01 (config-expl-path)#  index 1 next-label 16004  RP/0/RSP0/CPU0: PE-01 (config-expl-path)#  index 2 next-label 16200 | Specifies a label or an address in an explicit path of a tunnel. Note:   • List can include multiple addresses, labels, or both. However, it is not allowed to configure addresses  After configuring labels. Once starting  Configuring labels, all the coming will be also labels.  • Each entry must have a unique index.  • If the first hop is specified as next-label, that label must be an Adj-SID of the head-end or a prefix-SID label value known by the head-end. |
| Step 4 | **exit** |  |
| Step 5 | **interface tunnel-te** *tunnel-id* **Example:**  RP/0/RSP0/CPU0:PE-01(config)# interface tunnel-te 1 | Configures the tunnel interface. |
| Step 6 | **ipv4 unnumbered** *type interface-path-id*  **Example:**  RP/0/RSP0/CPU0:PE-01(config-if)# ipv4 unnumbered loopback0 | Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type. |
| Step 7 |  |  |
| Step 8 |  |  |
| Step 9 | Commit |  |

Table 7 Configure an Explicit SR-TE Policy [23]

#### 3.1.2.5 Steering Traffic into an SR-TE Policies

There are three options to steer a traffic into an SR-TE Policy

* **Static Routes**

Static routes can use the segment routing tunnel as a next-hop interface. Both IPv4 and IPv6 prefixes can be routed through the tunnel. A static route to a destination with a prefix-SID removes the IGP-installed SR-forwarding entry of that prefix.

* **AutoRoute Announce**

The SR-TE policy can be advertised into an IGP as a next hop by configuring the autoroute announce statement on the source router. The IGP then installs routes in the Routing Information Base (RIB) for shortest paths that involve the tunnel destination. Autoroute announcement of IPv4 prefixes can be carried through either OSPF or IS-IS.

* **AutoRoute Destination**

Autoroute destination allows to automatically route traffic through a segment routing tunnel instead of manually configuring static routes. Multiple autoroute destination addresses can be added in the routing information base (RIB) per tunnel.

# **Chapter 4: Testing and Results**

The implementation that was described in the previous chapter was used to build the virtual lab with different use cases to examine and verify Segment Routing capabilities and benefits.

The below segment routing capabilities were tested and verified as per of the thesis work

* Segment routing ECMP capability
* Working with both OSPF and IS-IS
* Configuring TE policies and steer the traffic into different routes
* Inter-domain TE tunnels with binding segment

## 4.1 Segment routing ECMP capability

Equal Cost Multipath is a strategy where the traffic can be forwarded from a source to the destination by using multiple best paths (equal cost). ECMP balances the overall traffic load and guarantees the efficient network utilization

I used the previous topology explained in the previous chapter and added a traffic generator ( Ostinato 9 ) for ECMP test and two clients and server which will be used for further tests.

ECMP capability was verified by two ways the first was exploring the Router forwarding tables like routing table, CEF table and MPLS forwarding table. I examined the available routes from P-01 ( where the traffic generator attached ) to PE-02 where as seen from the topology we have two equal routes to the destination through interface 0/1 and interface 0/4 and we can see that clearly on the below pictures taken from the router forwarding tables .

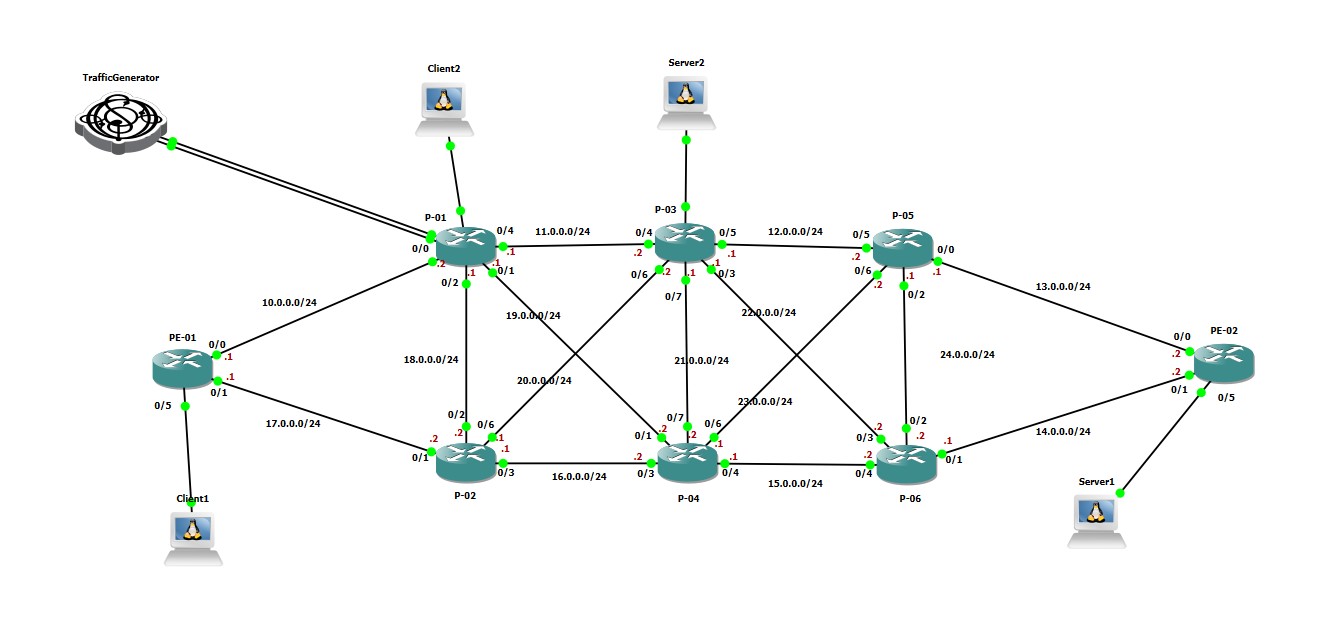


Figure 15 Topology with Traffic generator, Clients, and Servers for testing



Figure 16 MPLS forwarding Label towards PE-02

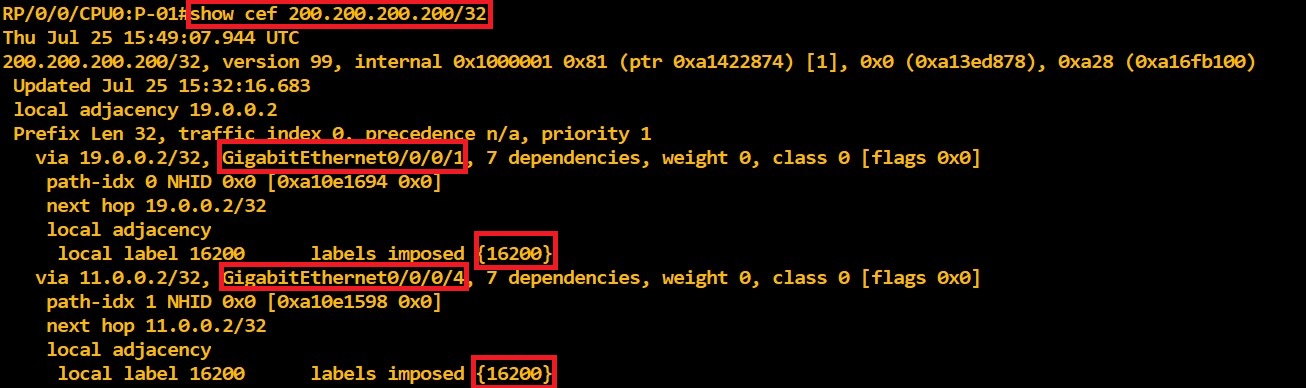


Figure 17 Routes in CEF table for destination 200.200.200.200

The other test done to confirm ECMP capability was using ostinato traffic generator and connect it to Router P-01 through two different interfaces then generate two different types of traffic (ICMP and TCP) ,one stream on each interface, for the same destination ( PE-02 ). And now we can observe the routes taken by each traffic and as we can see in figure 17 below the router sends traffic on both Links and number of bytes switched is changed for both interfaces while it was zero for both in the beginning as shown figure 15.

And we can also notice that on figure 18 and figure 19, which are traffic monitoring using Wireshark for interfaces 0/0/0/1 and 0/0/0/4, TCP traffic arrived on interface 0/0/0/5 with source IP 192.168.1.5 is forwarded to interface 0/0/0/1, while ICMP traffic received on interface 0/0/0/6 with source IP 192.168.2.2 is forwarded to interface 0/0/0/4. And both traffics are to the same destination and label number 16200 was used, as it is Node SID for PE-02.

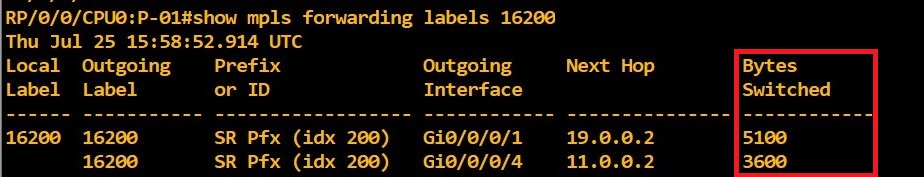


Figure 18 MPLS forwarding table after traffic generated

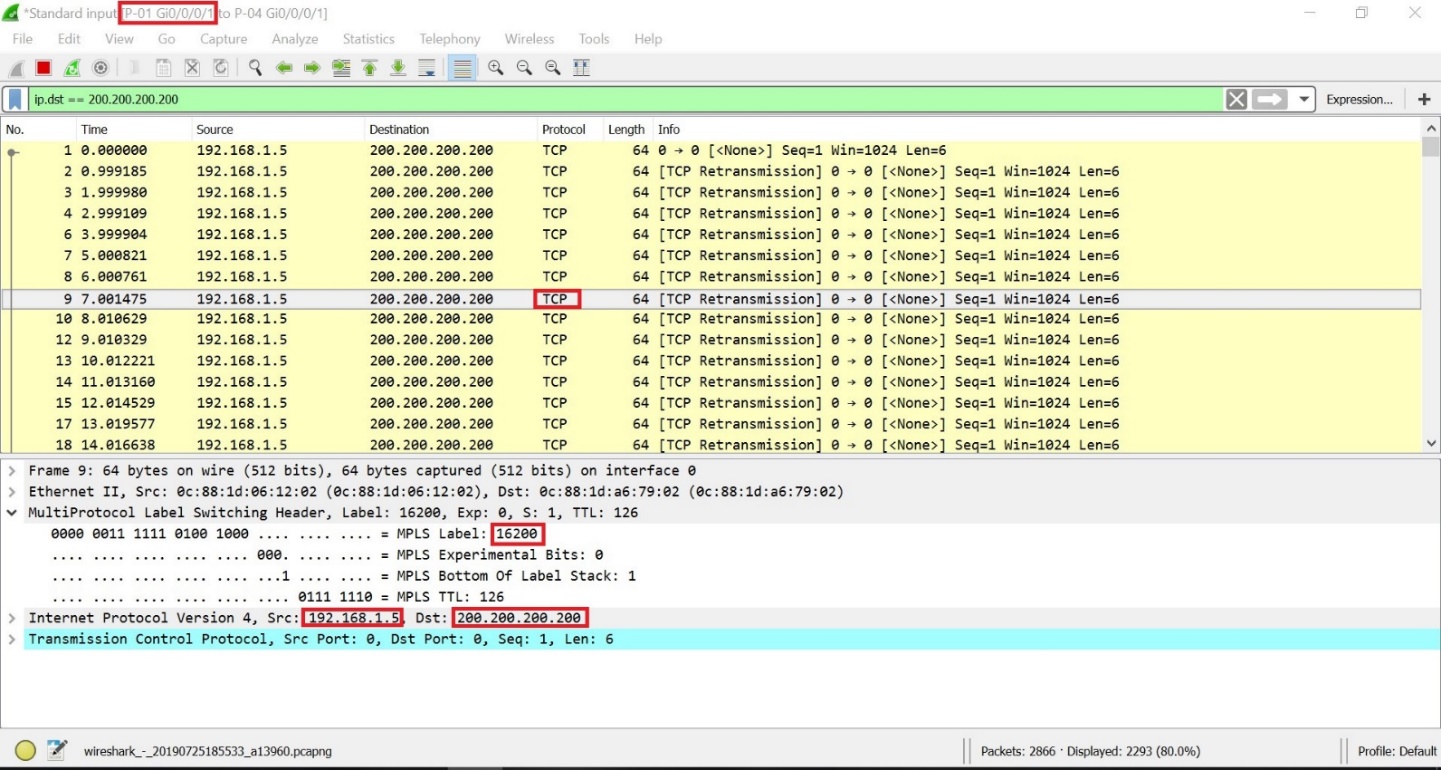


Figure 19 interface 0/0/0/1 traffic monitoring using Wireshark

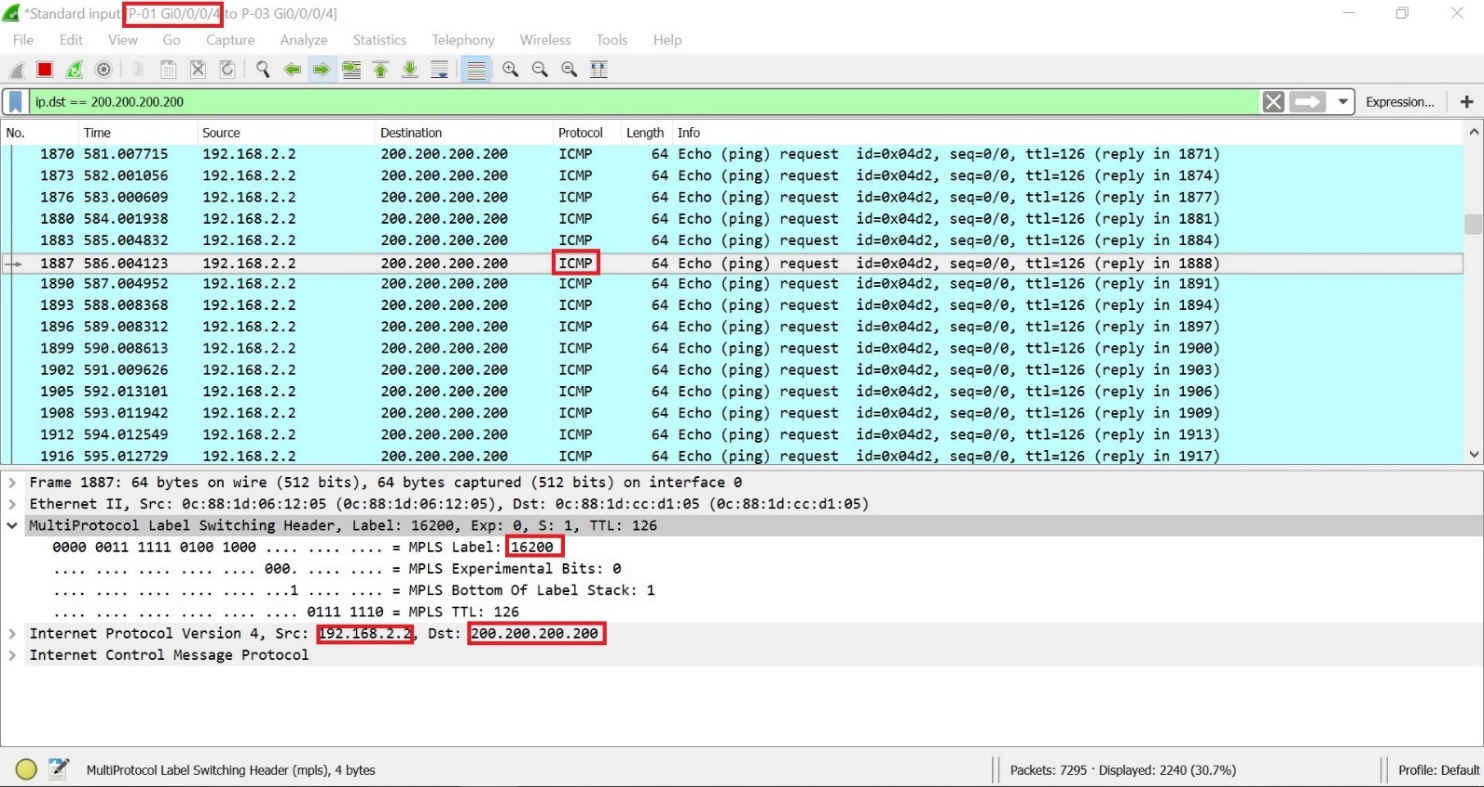


Figure 20 interface 0/0/0/4 traffic monitoring using Wireshark

## 4.2 Configuring Segment Routing with ISIS and OSPF

Segment routing depends mainly on the IGP protocol and as mentioned previously ISPs are using either ISIS or OSPF for their core network. I tested SR with ISIS and also with OSPF it worked perfectly with both protocols. In my test cases I didn’t find differences while using SR with ISIS or with OSPF.

In one of the test cases I combined two different networks one with OSPF and another one with ISIS and used segment routing traffic engineering to steer the traffic between the two different domains. (will explain the test case in details in section 4.5)

## 4.3 Configuring TE policies and steer the traffic into different routes

Traffic engineering is one of the most interesting use cases for Segment Routing as it promises to reduce overhead compared to RSVP. Traffic engineering has been a very hard and complex problem especially in large networks. In this section, I will apply different traffic engineering scenario that will show how SR traffic engineering can introduce a simple solution for traffic engineering purposes.

MPLS-TE uses RSVP to establish a Label switched path (LSP). However RSVP has many complexity and scalability issues. Each node on the network has to maintain state for each LSP traversing through it. The LSP state also has to be periodically refreshed. This will be a problem when the number of LSPs grows to thousands.

Unlike MPLS-TE, SR does not need to maintain the state of the LSP in the transit routers as the forwarding is depending only on the list of labels, which were imposed by the head-end router. The head-end router will impose the list of SIDs on the packet and the transit routers will forward the packet according to the associated instructions.

On the same topology I applied, different policies were used to steer the traffic from the head-end (Router PE-01) to the destination (PE-02) that will show how it is very simple to configure a SR-TE and steer the traffic into it using both dynamic and static SR-TE policies.

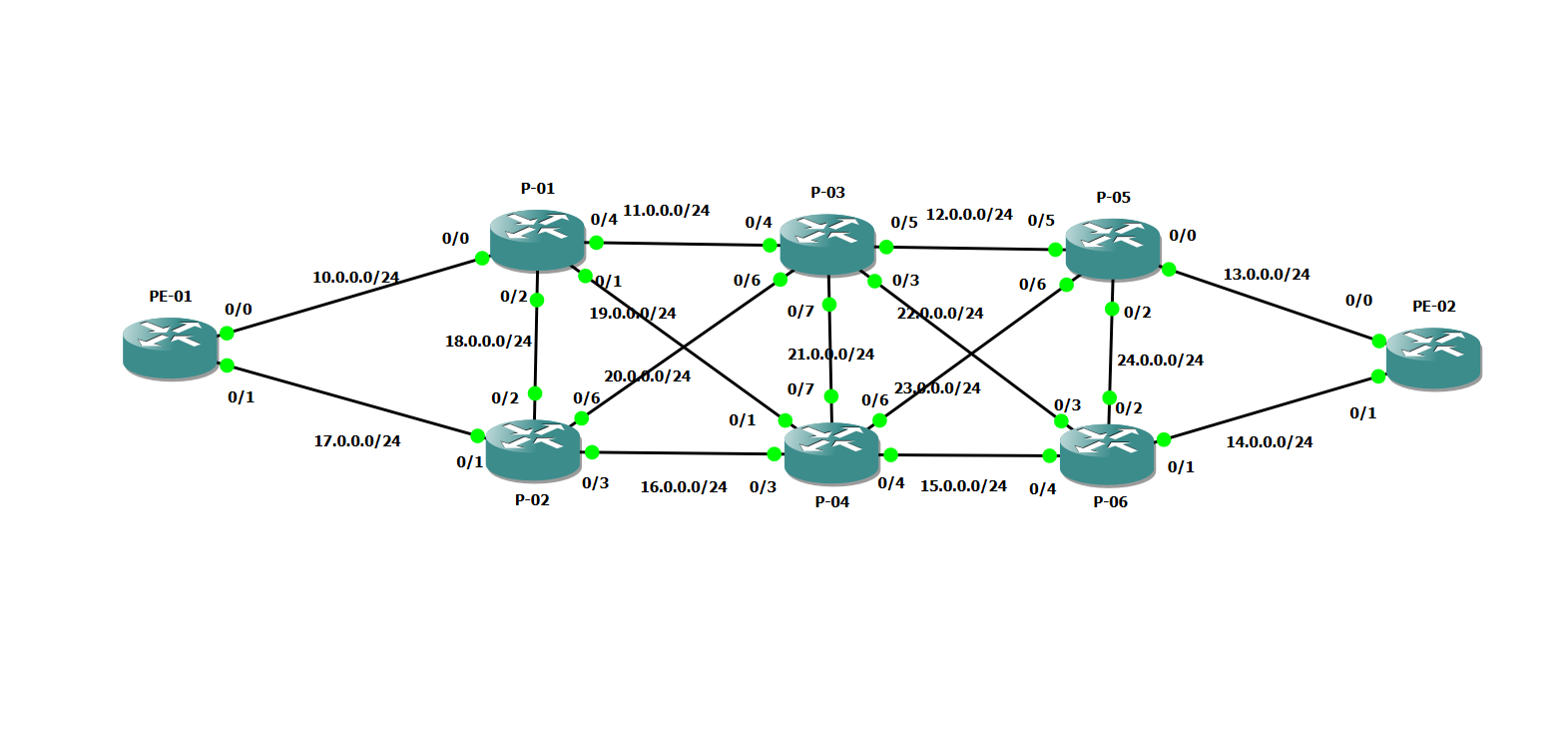


Figure 21 Network topology for traffic engineering testing

### 4.3.1 Steering Traffic into using dynamic SR-TE Policy

Two steps are required to steer the traffic into a dynamic tunnel.  
Firstly, configure the tunnel then configure a route for the traffic into the tunnel (here I used a static Route to steer the traffic into the tunnel). With Dynamic SR-TE Policy, the head-end calculates the route to the destination dynamically and the calculations results in a list of interface IP addresses that traffic engineering (TE) maps to adj-SID labels as shown in figure 21.

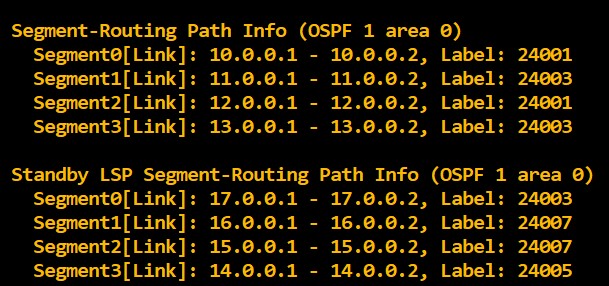


Figure 22 Head-end calculations results

After configuring the tunnel and configuring the router to steer the traffic into the tunnel if we traced the traffic from PE-01 to PE-02 we can notice as shown in figure 22 how will PE-01 will depend on the adj-SID labels to reach PE-02. To better illustrate the effect, I used Ostinato to generate traffic from PE-01 to PE-02. Figure 23 shows the results from Wireshark monitoring and the content of the packet.

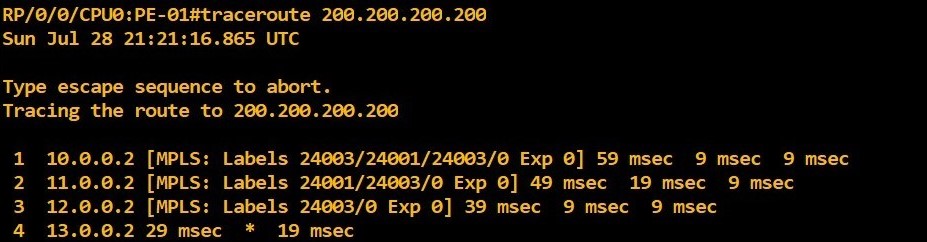


Figure 23 Traceroute from PE-01 to PE-02

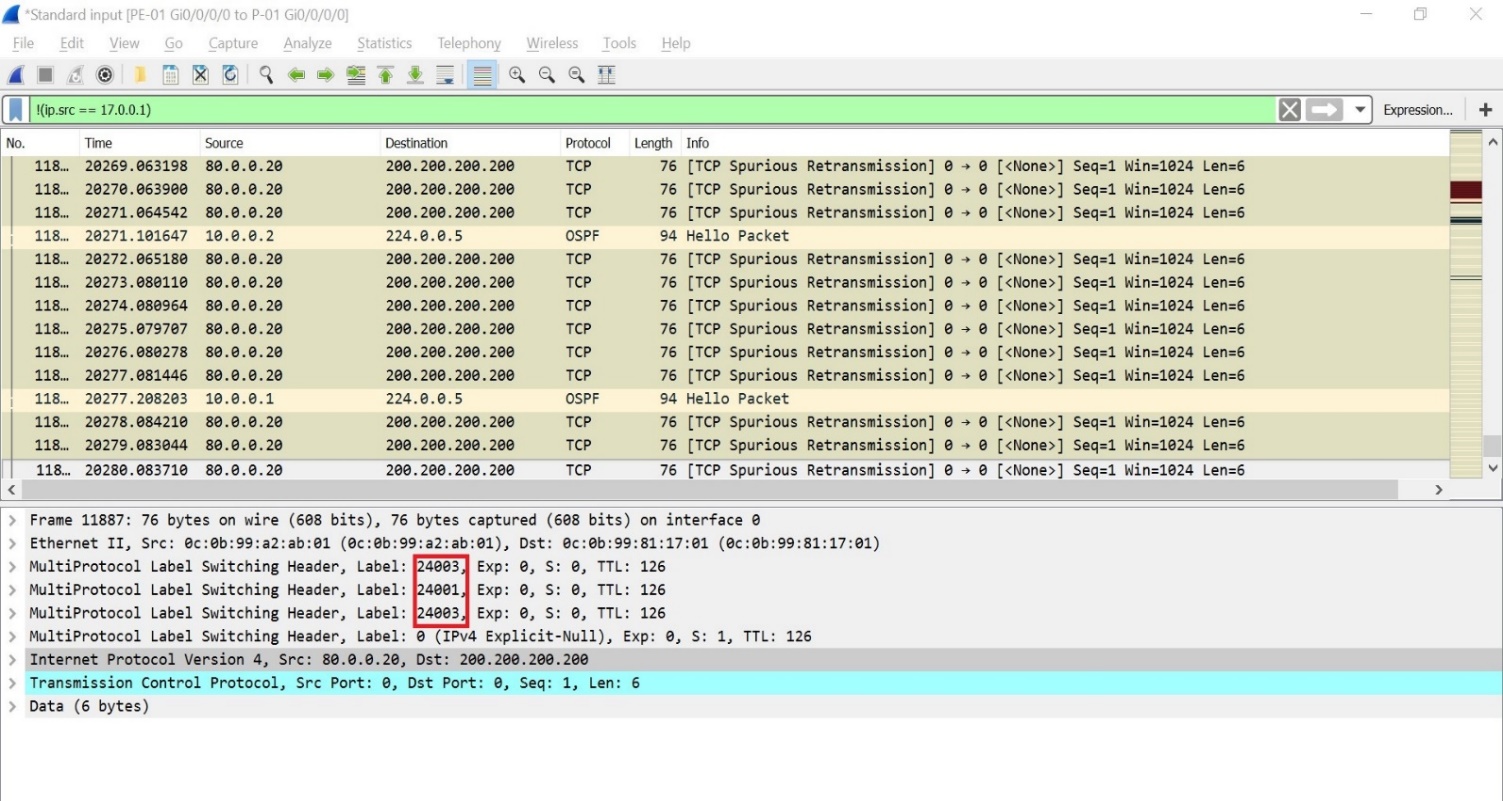


Figure 24 Monitoring Link between PE-01 and P-01 using Wireshark

### 4.3.2 Steering Traffic into using explicit SR-TE Policy

Three steps are required to steer the traffic using an explicit list.  
The first two steps are the same as the dynamic tunnel but will need a third step, which will be the setup of a defined explicit path for the traffic. This is a list of IP addresses or labels, each IP or label is representing a node or a link in the route from the source to the destination.

For testing, I configured a list to steer the traffic from PE-01 to PE-02 through P-01🡪P🡪02🡪P-03🡪P-04🡪P05🡪P-06 using the nodes SID figure 24 shows the list, figure 25 shows how will PE-01 depend on the pre-defined route to steer the traffic and figure 26 shows how will the packet looks like.

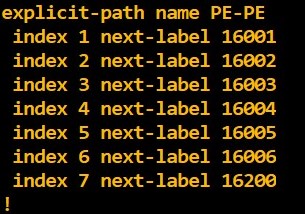


Figure 25 Explicit path List

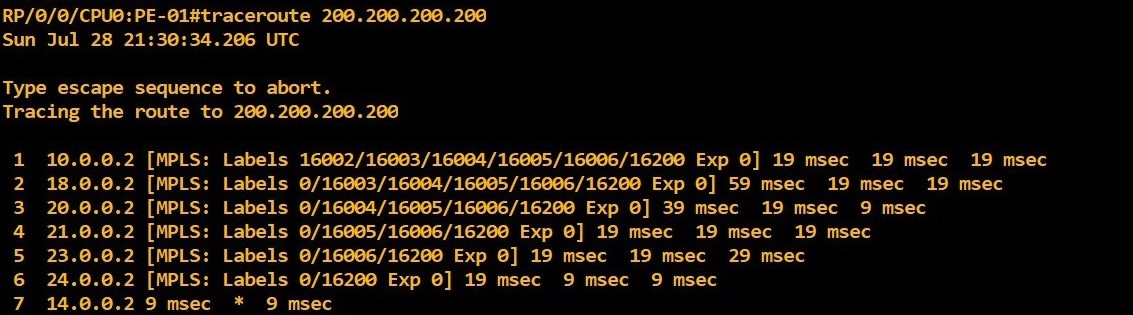


Figure 26 Router PE-01 uses the explicit List to reach router PE-02

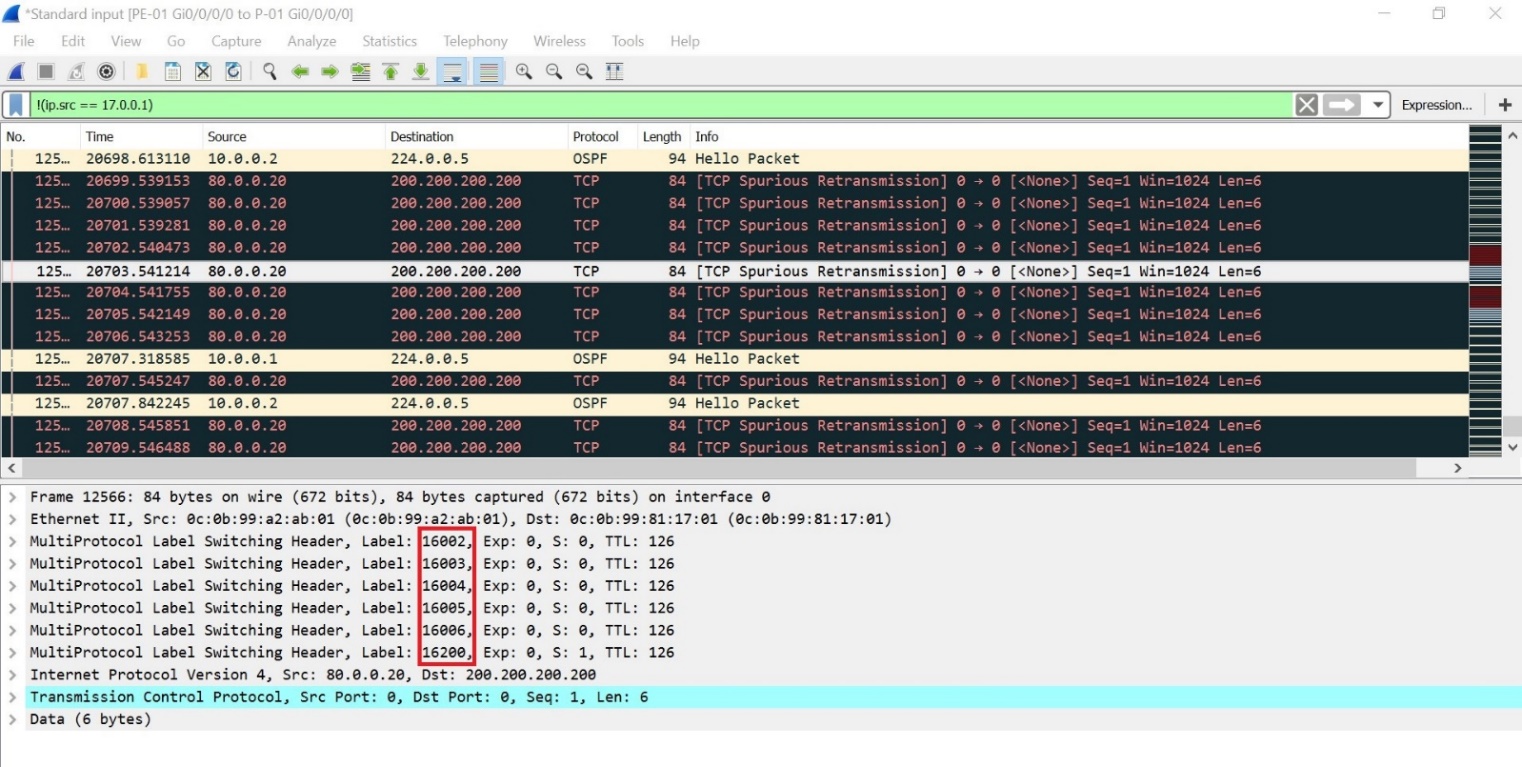


Figure 27 Packet Sample showing the use of the explicit list

### 4.3.3 Steering Traffic using explicit SR-TE Policy using Node SID and adj-SID

SR-TE is very flexible and gives the possibility to combine node and link SIDs in the same policy.

In the last uses case I combined a list defining the route through a specific node and a specific link. The list will steer the traffic to node P-03 and network 15.0.0.0 which has label number 24006 and the intermediate routes will be defined by the IGP. This scenario maybe useful to reduce the number of labels, which will help to reduce the packet size and may avoid the need to increase the MTU.

Figure 27 shows the list, figure 28 shows how will PE-01 depend on the pre-defined route to steer the traffic and figure 29 shows how will the packet looks like.

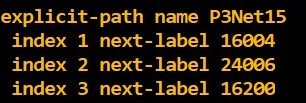


Figure 28 Explicit path List with Node SID and Adj-SID

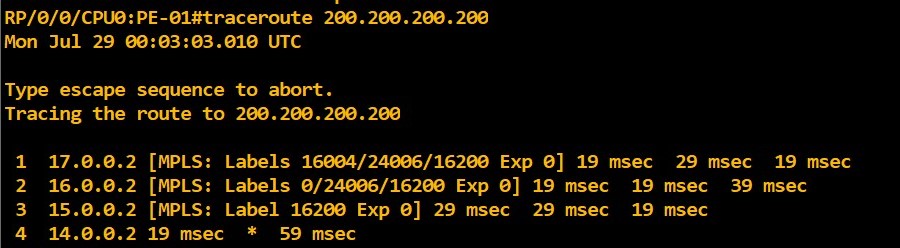


Figure 29 PE-01 is using node SID and adj-SID to reach PE-02

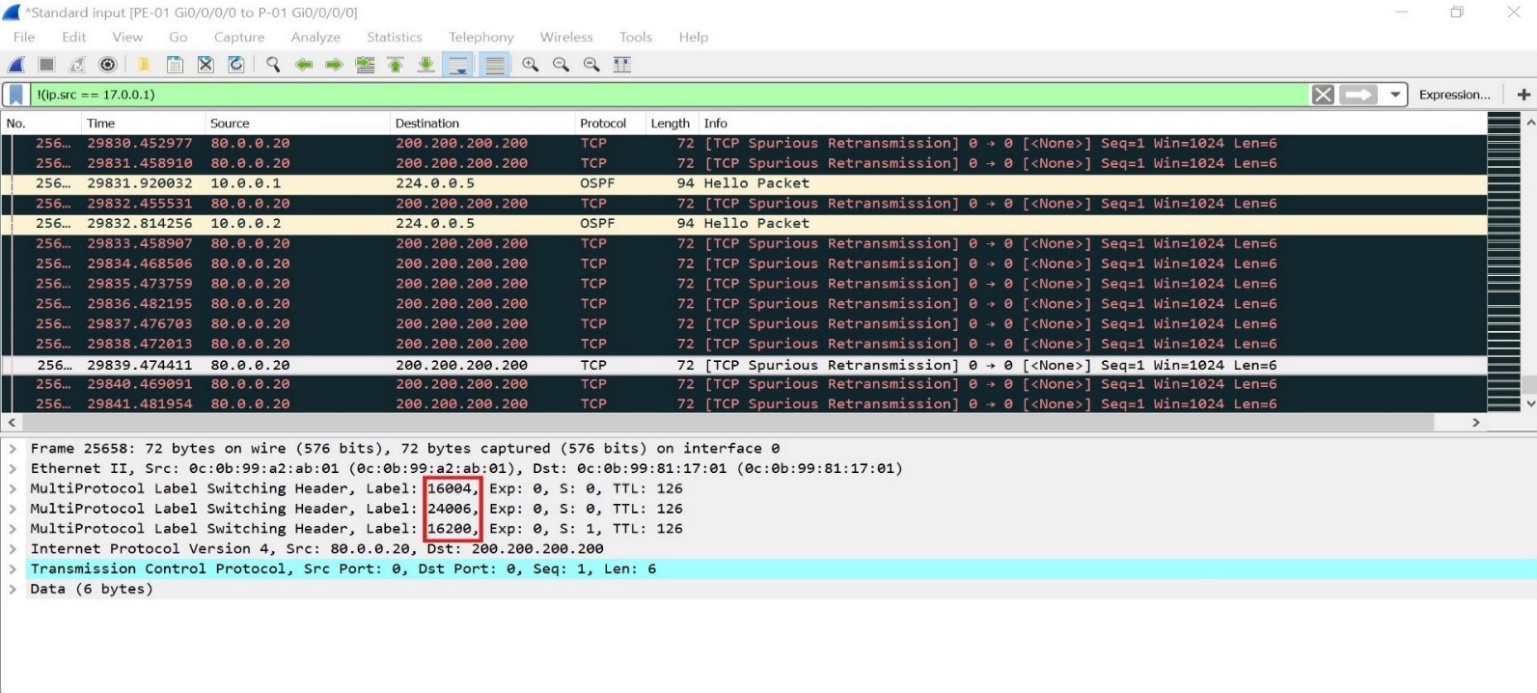


Figure 30 Node SID and adj-SID imposed in the packet going from PE-01 to PE-02

## 4.5 Inter-domain TE tunnels with binding segment

The binding segment is a local segment identifying an SR-TE policy. Each SR-TE policy is associated with a binding segment ID (BSID). The BSID is a local label that is automatically allocated for each SR-TE policy.  
BSID can be used to steer traffic into the SR-TE policy and across domain borders, creating seamless end-to-end inter-domain SR-TE policies. Each domain controls its local SR-TE policies independent from the remote domain’s head-end. Using binding segments isolates the head-end from topology changes in the remote domain.

In addition to inter-domain routing binding segment also can help to compress the label stack if the label stack size required for an SR-TE policy exceeds the platform capability, the SR-TE policy can be seamlessly combined with other SR-TE policies using a binding segment.

In this use case I will illustrate the inter-domain routing using binding segment. I have two separate routing domains with one running OSPF and the other IS-IS and the goal is to have a tunnel between PE-01 and PE-02 while each router is in different domain.

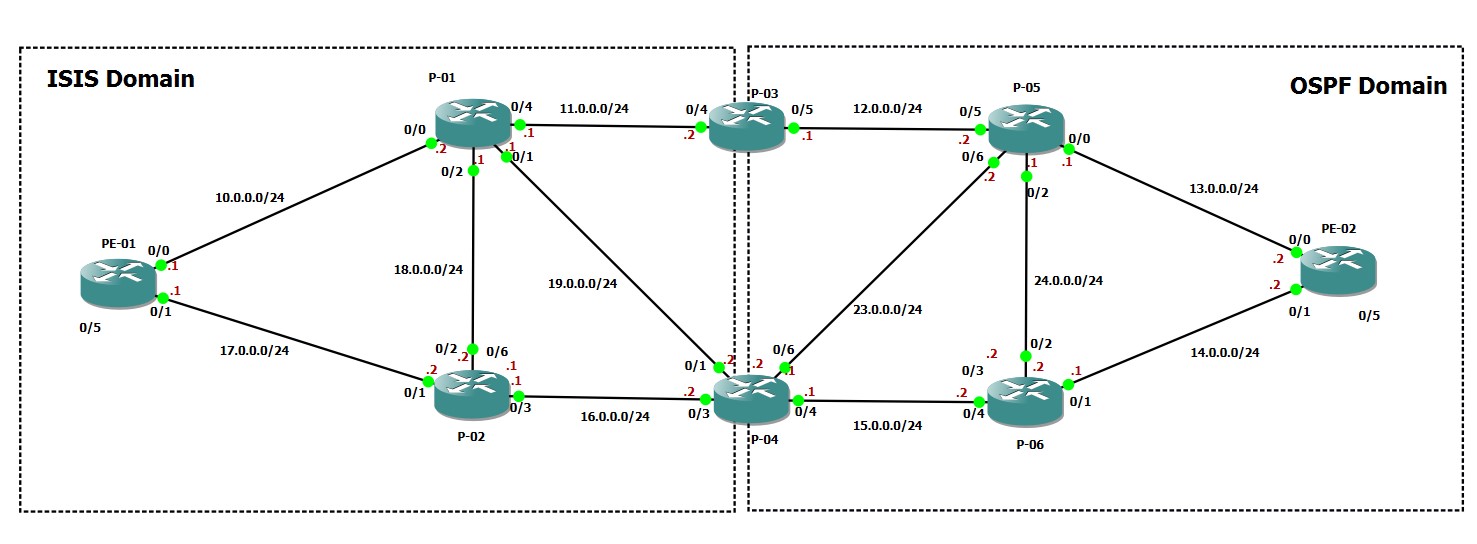


Figure 31 Two separate routing domains topology

Firstly, I will need to create two tunnels on the border router P-03 on tunnel to PE-02 and the other tunnel to PE-01. The IOS XRV automatically assigns binding-SID for SR-Tunnels, but it is also allowed to assign the binding-SID manually as I did here.  
  
Binding-SID can be verified from the tunnels details as shown in figure 33 and figure 34. Label 4000 is from the border router to PE-02 and Label 3000 is from the border router to PE-01.

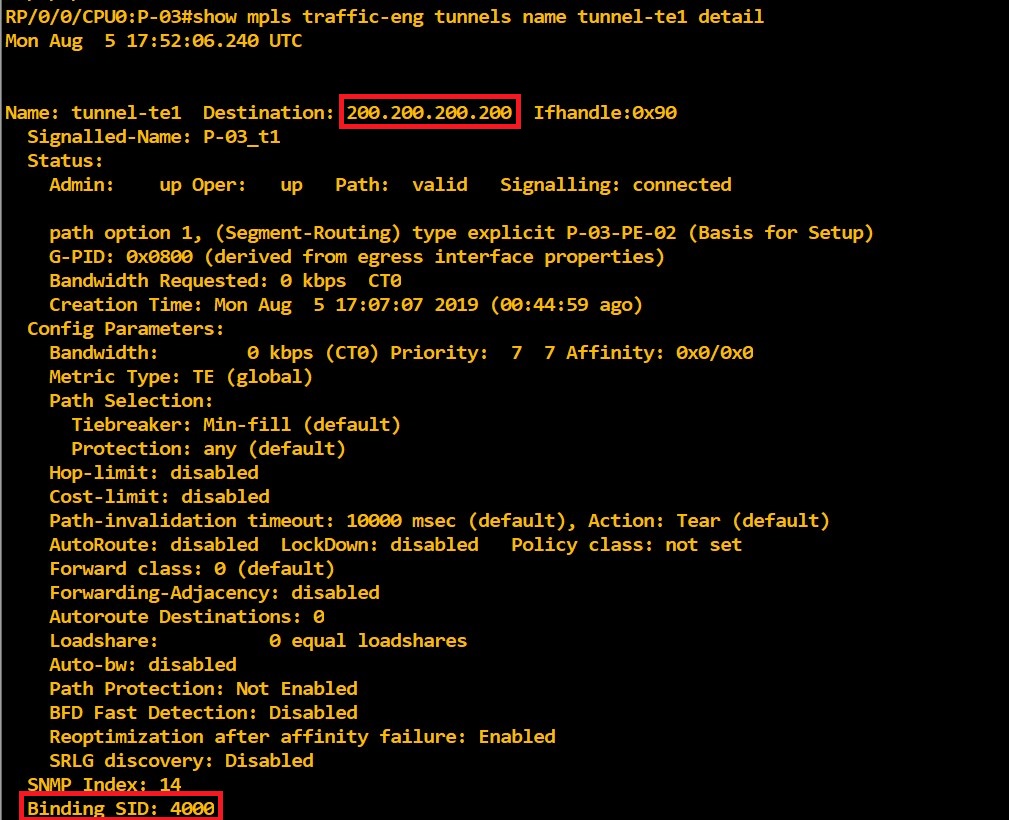


Figure 32 Border router tunnel details towards PE-02



Figure 33 Border router tunnel details towards PE-01

After that I defined a tunnel from PE-01 to the border router P-04 but adding the binding-SID 4000 at the end of the explicit list so once the border router receive the traffic will forward the traffic in its tunnel to PE-02. And did the same on PE-02 by defining Tunnel towards the border router but adding Label 3000 at the end of the List so the border router can forward the traffic towards PE-01

Now I am able to send traffic between the two routers, however each router in different routing domain and we can notice while tracing the packets from PE-01 to PE-02 it has two labels one label to reach the border router and the other one is the binding-SID 4000 which will steer the traffic from the border router to PE-02

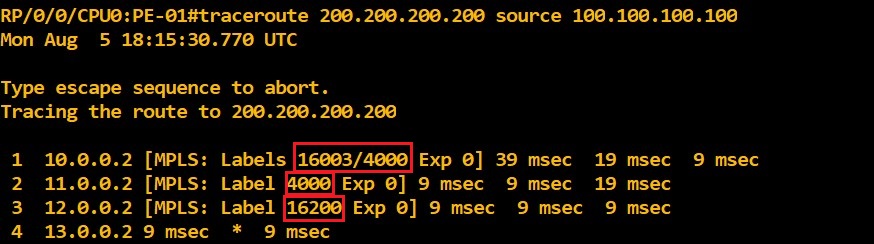


Figure 34 The use of binding-SID between PE-01 and PE-02

# **Chapter 5: Conclusion and future work**

### 5.1 Conclusion

This thesis aimed to investigate the new source routing mechanism called segment routing, focus on a number of core features and benefits segment routing offers and provide the instructions for building a virtualized laboratory that can be useful for others to continue investigating segment routing .

The Thesis started out with a comprehensive overview on segment routing, how it is working and showing its benefits and why segment routing is emerging.

The implementation of the virtualized environment was achieved. The lab was tested and then a number of relevant uses cases were implemented, conducte, and analyzed. Each case was briefly discussed.   
  
The first use case verified the use of the Equal Cost Multi-Path (ECMP) feature to balance the traffic between multiple available equal paths in the network. This feature results in a better bandwidth utilization and this flexibility does not exist in the current RSVP-TE, where the parallel use of paths will need complex manual configuration.

As Some ISPs are using OSPF but also others are using ISIS for their core network, it was important to verify that segment routing can work with both protocols and that was verified in the second uses case.

In the third use case segment routing traffic engineering was discussed with different scenarios that showed how it is easy to configure segment routing traffic engineering and steer the traffic using both dynamic and static SR-TE Policies and showing the flexibility in using Node SID or Adj-SID or combining between them to choose a specific route towards the destination.

The last uses case was about segment routing traffic engineering applications that can be used for connecting two different routing domains using a so-called binding segment. This involved the configuration of two tunnels on one border router and using a binding segment to help stitch the two tunnels together. The binding segment acted as an abstraction point hiding the internal topology from the respective other side and occurring internal changes. As an additional benefit, the Binding segment can also help reduce the depth of the label stack that the head-end has to push.

Segment Routing does not require a massive upgrade to the existing network and does not require huge investments to be implemented. Segment Routing will eliminate the need for signaling protocols Like RSVP-TE and LDP which will lead to increased network simplicity and thus better stability and manageability.

Network providers are moving towards SDN paradigm. All the tested segment routing features can be orchestrated from high-level application with help of SDN controller. Segment routing can be a powerful tool that allows the SDN controller to improve the existing network performance and ease of automation.

### 5.2 Future work

In this thesis, I tried to explore most of Segment Routing benefits. Due to hardware limitations I could not run IOS XRV versions newer than 6.1.3, which does not support SRV6. The verification of the functionality and ease of implementation should be tested with SRV6, as this would be the logical step for IPv6 core networks.

Having the virtual environment ready one can start testing and applying other uses cases and relevant features for network operators, such as Topology Independent Loop Free Alternative (TI-LFA) which is supported by Segment Routing and introduces new solutions more effective than traditional fast reroute (FRR). This effectiveness and the convergence time as the critical key parameter should be explored in future research, as well as detailed interoperability.

# **REFERENCES**

[1] “Segment Routing: a Comprehensive Survey of Research Activities, Standardization Efforts and Implementation Results“ [online]: <https://arxiv.org/pdf/1904.03471.pdf>

[2] S. Yasukawa, A. Farrel, O. Komolafe. “An Analysis of Scaling Issues in MPLS-TE Core Networks“. IETF RFC 5439, February 2009 [online]: <https://tools.ietf.org/html/rfc5439#section-2.3>

[3] S. Previdi, C. Filsfils. “Source Packet Routing in Networking (SPRING) Problem Statement and Requirements“. IETF RFC 7855, May 2016 [online]: <https://tools.ietf.org/html/rfc7855>

[4] Adrian Reuter , Technische Universität München Paper. “Source Packet Routing in Networking (SPRING) “. [online]: <https://pdfs.semanticscholar.org/d42a/6b5961b0078a0cd6195496262943f932fab6.pdf>

[5] S. Previdi, C. Filsfils L. Ginsberg. “Segment Routing Architecture“. IETF RFC 8402, July 2018 [online]: <https://tools.ietf.org/html/rfc8402>

[6] C. Filsfils ,S. Previdi. “Segment Routing Architecture”. IETF draft-ietfspring-segment-routing-07, December 2015 [online]: <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-07>

[7] S. Previdi, C. Filsfils. “Segment Routing with MPLS data plane” IETF draft-ietf-spring-segment-routing-mpls-03, February 2016 [online]: <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-mpls-03>

[8] C. Filsfils ,S. Previdi. “Segment Routing Architecture” IETF draft-filsfils-rtgwg-segment-routing-00, June 2013 [online]: <https://tools.ietf.org/html/draft-filsfils-rtgwg-segment-routing-00>

[9] C. Filsfils ,S. Previdi, A. Bashandy. “Segment Routing interoperability with LDP”, IETF draft-ietf-spring-segment-routing-ldp-interop-00, October 2015 [online]: <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-ldp-interop-00>

[10] C. Filsfils ,S. Previdi, P. Francois. “Segment Routing Use Cases”, IETF draft-filsfilsrtgwg-segment-routing-use-cases-02, October 2013 [online]: <https://tools.ietf.org/html/draft-filsfils-rtgwg-segment-routing-use-cases-02>

[11] C. Filsfils , P. Francois, M. Horneffer. “Loop-Free Alternate (LFA) Applicability in Service Provider (SP) Networks”. IETF RFC 6571 [online]: <https://tools.ietf.org/html/rfc6571>

[12] Clarence Filsfils, Kris Michielsen. “Introduction to the Topology Independent LFA technology”[online]: <https://www.segment-routing.net/tutorials/2016-09-27-topology-independent-lfa-ti-lfa/>

[13] Ana Kos. “Segment routing principles and applications for sdn”. Master Thesis, politecnico di Milano

[14] Sarvesh Bidkar, Ashwin Gumaste.”Scalable segment routing-a new paradigm for efficient service provider networking using carrier Ethernet advances”. IEEE/OSA Journal of Optical Communications and Networking [online]:

<https://ieeexplore.ieee.org/document/7107877>

[15] Cisco, “Segment Routing: Prepare Your Network for New Business Models” White Paper.

[16] Segment Routing Configuration Guide, Cisco IOS XE [online]: <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/seg_routing/configuration/xe-16-5/segrt-xe-16-5-book/intro-seg-routing.html>.

[17] S. Yasukawa, A. Farrel. “An Analysis of Scaling Issues in MPLS-TE Core Networks”. IETF RFC 5439, February 2009 [online]: <https://tools.ietf.org/html/rfc5439#section-2.3>

[18] M. Shand, S. Bryant. “IP Fast Reroute Framework”. IETF RFC 5714 , January 2010 [online]: <https://tools.ietf.org/html/rfc5714#section-5>

[19] A. Atlas, A. Zinin. “Basic Specification for IP Fast Reroute: Loop-Free Alternates”. IETF RFC 5286 , September 2008 [online]: <https://tools.ietf.org/html/rfc5286>

[20] Juniper “Topology-Independent Loop-Free Alternate with Segment Routing for OSPF” [online]: <https://www.juniper.net/documentation/en_US/junos/topics/topic-map/ti-lfa-segment-routing-ospf.html>

[21] <http://www.mplsvpn.info/2015/07/segment-routing-based-mpls-vs-classic.html>

[22] Cisco IS-IS Overview and Basic Configuration [online]: <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_isis/configuration/15-mt/irs-15-mt-book/is-is_overview_and_basic_configuration.html#GUID-F1498D99-0DD0-40C8-8978-16003E5B7E8F>

[23] Cisco Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 6.1.x [online]: <https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/segment-routing/configuration/guide/b-seg-routing-cg-asr9k.html>

[24] ] Juniper “Supported Standards for Segment Routing” [online]: <https://www.juniper.net/documentation/en_US/junos/topics/reference/standards/segment-routing.html>

[25] Pierre Francois, Clarence Filsfils, Ahmed Bashandy. “Topology Independent Fast Reroute using Segment Routing” IETF draft-francois-rtgwg-segment-routing-ti-lfa-00 ,August 2015 [online]: <https://tools.ietf.org/html/draft-francois-rtgwg-segment-routing-ti-lfa-00>.

[26] Alcatel-Lucent Technology White Paper “Segment Routing and Path Computation Element Using traffic engineering to optimize path placement and efficiency in IP/MPLS networks” [online]: <https://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2015/11525-segment-routing-path-computation-element-using-traffic-engineering.pdf>

[27] A. Farrel, P. Vasseur. “A Path Computation Element (PCE)-Based Architecture”. IETF RFC 4655 , August 2006 [online]: <https://tools.ietf.org/html/rfc4655>

[28] S. Previdi, L. Ginsberg, C. Filsfils.“IS-IS Extensions for Segment Routing“ draft-ietf-isis-segment-routing-extensions-19”, July 2018 [online]: <https://tools.ietf.org/html/draft-ietf-isis-segment-routing-extensions-19>

[29] P. Psenak, S. Previdi, C. Filsfils. “OSPF Extensions for Segment Routing” draft-ietf-ospf-segment-routing-extensions-24”, December 2017[online]: <https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-24>

[30] C. Filsfils, S. Sivabalan. “Segment Routing Policy for Traffic Engineering” IETF draft-filsfils-spring-segment-routing-policy-04”, December 2017[online]: <https://tools.ietf.org/html/draft-filsfils-spring-segment-routing-policy-04>

[31] Juniper “Understanding Static Segment Routing LSP in MPLS Networks” [online]: <https://www.juniper.net/documentation/en_US/junos/topics/concept/understanding-static-segment-routing-lsp.html#jd0e22>

[32] IETF draft “Segment Routing Policy for Traffic Engineering draft-filsfils-spring-segment-routing-policy-05”[online]: <https://tools.ietf.org/html/draft-filsfils-spring-segment-routing-policy-05>

[33]Juniper , “Understanding Topology-Independent Loop-Free Alternate with Segment Routing for IS-IS”.[online]: <https://www.juniper.net/documentation/en_US/junos/topics/concept/understanding-ti-lfa-for-is-is.html>

[34]Cisco , Configure Topology-Independent Loop-Free Alternate (TI-LFA). [online]: <https://www.cisco.com/c/en/us/td/docs/routers/ncs6000/software/segment-routing/configuration/guide/b-segment-routing-cg-ncs6k/b-segment-routing-cg-ncs6k_chapter_01000.pdf>

[35] Clarence Filsfils, Kris Michielsen. “Segment RoutingMPLS data plane”.[online]: <https://www.segment-routing.net/images/0040-SR-TOI-SR_IGP_control_plane_v11a.pdf>

[36] Cihat Cetinkaya, Erdem Karayer. “SDN for segment based flow routing of DASH”, 2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin).[online]: <https://ieeexplore.ieee.org/abstract/document/7034284/references#references>

[37] Randeep Bhatia, Fang Hao, Murali Kodialam, T.V. Lakshman.”Optimized Network Traffic Engineering usingSegment Routing”. 2015 IEEE Conference on Computer Communications (INFOCOM).

[38] Juniper , Understanding Source Packet Routing in Networking (SPRING). [online]: <https://www.juniper.net/documentation/en_US/junos/topics/concept/source-packet-routing.html>

[39] <http://wwwlehre.dhbw-stuttgart.de/~schulte/htme/55175.htm>

[40] Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers [online]<https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-5/segment-routing/configuration/guide/b-segment-routing-cg-asr9000-65x/b-segment-routing-cg-asr9000-65x_chapter_0111.html>

[41] Ilpo Litmanen, „Segment Routing Thesis “. [online] : <https://www.theseus.fi/bitstream/handle/10024/124568/Litmanen_Ilpo.pdf?sequence=1>

[42] Daniel Dib , Segment Routing: Potential For Far-Reaching Network Effects , June 2015[online]: <https://www.networkcomputing.com/networking/segment-routing-potential-far-reaching-network-effects>

[43] Juniper, “Understanding Adjacency Segments, Anycast Segments, and Configurable SRGB in SPRING”, March 2019.[online] : <https://www.juniper.net/documentation/en_US/junos/topics/concept/understanding-support-for-spring-anycast-and-prefix-segments.html>

[44] Cisco, “Segment Routing for Service Providers”.[online] :  
 <https://www.cisco.com/c/m/en_us/training-events/events-webinars/webinars/segment-routing.html>

[45] C. Hopps , “Analysis of an Equal-Cost Multi-Path Algorithm“, IETF RFC 2992, November 2000.[online]: <https://tools.ietf.org/html/rfc2992>

[46] R. Gandhi, C. Filsfils “Performance Measurement for Segment Routing Networks with MPLS Data Plane” IETF draft-gandhi-spring-rfc6374-srpm-mpls-01, May 2019.[online]: <https://tools.ietf.org/pdf/draft-gandhi-spring-rfc6374-srpm-mpls-01.pdf>

[47] D. Frost, S. Bryant. “Packet Loss and Delay Measurement for MPLS Networks”.IETF RFC 6347 , September 2011.[online] : <https://www.rfc-editor.org/rfc/rfc6374.txt>

[48] A. Bashandy, C. Filsfils.”Topology Independent Fast Reroute using Segment Routing”, IETF draft-bashandy-rtgwg-segment-routing-ti-lfa-05, October 2018.[online]: <https://www.ietf.org/archive/id/draft-bashandy-rtgwg-segment-routing-ti-lfa-05.txt>

[49] Klaus-Tycho Foerster, Stefan Schmid, and Stefano Vissicchio, “Survey of Consistent Software-Defined Network Updates” [online]: <https://net.t-labs.tu-berlin.de/~stefan/survey-network-update-sdn.pdf>

[50] Ray Mota ,”ACG research paper” [online]: <https://www.segment> routing.net/images/ACG\_Segment\_Routing\_201808.pdf

[51] Sterling Perrin , “Making networks sdn ready with segment routing”, white paper produced for Cisco Systems Inc [online]: <https://www.segment-routing.net/images/lightreading_report.pdf>

[52] Frederic Trate, “Equinix Segment Routing-powered network delivers increased value to its customers” , Cisco Blog [online]: <https://blogs.cisco.com/sp/equinix-segment-routing-powered-network-delivers-increased-value-to-its-customers>

[53] A. Farrel, D. Papadimitriou , “Encoding of Attributes for MPLS LSP Establishment Using

Resource Reservation Protocol Traffic Engineering (RSVP-TE)”, IETF RFC 5420, February 2009.[online]: <https://tools.ietf.org/html/rfc5420>

[54]Cisco White paper , “Advanced Topics in MPLS-TE Deployment”.[online]: <https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/multiprotocol-label-switching-traffic-engineering/whitepaper_c11-551235.pdf>

[55] S. Litkowski, A. Bashandy, C. Filsfils” Topology Independent Fast Reroute using Segment Routing IETF draft-ietf-rtgwg-segment-routing-ti-lfa-01”. March 2019.[online]: <https://www.ietf.org/id/draft-ietf-rtgwg-segment-routing-ti-lfa-01.txt>

[56] Cisco XR Virtual Router Installation and Configuration Guide.[online]: <https://www.cisco.com/en/US/docs/ios_xr_sw/ios_xrv/install_config/b_xrvr_432.pdf>

[57] <https://www.segment-routing.net/>

[58]<https://ostinato.org/>

[59]<https://howdoesinternetwork.com/2015/how-to-generate-network-packets>

[60]<https://iperf.fr/>

[61]<https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9_aLY8kkdhgaMB0wPCz8a38/index.html>

[62][https://docs.vmware.com](https://docs.vmware.com/)

[63][https://www.techspot.com](https://www.techspot.com/)

[64] [https://www.linkflapping.net](https://www.linkflapping.net/)