



Teemu Lius

Segment Routing

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

1 March 2021

Abstract

Author: Teemu Lius
Title: Segment Routing
Number of Pages: 39 pages + 0 appendices
Date: 1 March 2021

Degree: Bachelor of Engineering
Degree Programme: Information and Communication Technology
Professional Major: IoT and Cloud Computing
Supervisors: Janne Salonen, Head of School, School of ICT

Aim of this project was to study segment routing as a technology and compare it to its predecessors, as well as configuring a segment routing environment with a virtualized demonstration environment. This included personally creating the said environment.

Virtual demonstration consisted of GNS3 as a network software emulator software, with VirtualBox hypervisor installed on my personal PC. This included GNS3 VM inside the VirtualBox and as the routing devices, Cisco IOS-XRv and CSR1000v virtual routers were used.

Inside the virtual environment a topology of five routers was built. Using this topology segment routing was configured on the devices in four parts. With the main points explained in sub-heading titles and device configurations provided to the reader.

Results of the project showcased the simplicity of segment routing and highlighted the possible compatibility issues when using multiple different components for virtualized testing environment.

Keywords: Segment Routing, Networking, GNS3, MPLS

Tiivistelmä

Tekijä:	Teemu Lius
Otsikko:	Segment Routing
Sivumäärä:	39 sivua
Aika:	1.3.2021
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaina:	Verkko- ja pilvipalvelut
Ohjaajat:	Janne Salonen, Osaamisaluepäällikkö

Tämän insinööriyön päämääränä oli perehtyä Segment Routing -reititys teknologiaan ja sen edeltäjien vertailuun. Sekä kyseisen teknologian demonstroimiseen virtuaalisella testi alustalla. Projektiin kuului tämän alustan pystytys.

Virtuaalinen demonstraatio koostui GNS3 sovelluksesta, joka toimi verkko sovelluksen emulaatio sovelluksena, sekä VirtualBox sovelluksesta, joka toimi tässä tapauksessa hypervisorina. Tähän sisältyi myös GNS3 virtuaalikone, sekä pari Ciscon virtuaalireitintä, jotka olivat: Cisco IOS-XRV ja Cisco CSR1000v.

Virtuaalisen ympäristön sisällä oli viidestä reitittimestä koostuva topologia. Tätä topologiaa hyödyntäen, tehtiin demonstraatio segment routing -reititys teknologiasta, neljässä osassa. Nämä osat korostettiin ali-otsikkojen nimissä, joiden alla myös näytettiin tärkeimmät konfiguraatiot.

Projektin lopputulos korosti tämän teknologian yksinkertaisuutta, sekä mahdollisia yhteensopivuusongelmia, kun käytetään useita eri komponentteja virtuaalista ympäristöä tehdessä.

Keywords: Segment routing, Tietoverkot, GNS3, MPLS

Contents

List of Abbreviations

1	Introduction	1
2	Software Designed Networking	1
2.1	Architecture of Software Designed Networking	2
2.2	Centralized Management with SDN	4
2.3	SDN with Segment Routing	5
3	Segment Routing	6
3.1	MPLS Routing Technique	6
3.2	Segment Routing comparison to MPLS	7
3.3	Segment Routing Architecture	9
3.3.1	Segments	11
3.3.2	Advertisement of Segments	12
3.3.3	Local and Global Segments	12
3.3.4	IGP-SIDs – IGP Segment Identifiers	13
3.3.5	Segment Identifiers	13
3.3.6	Routing	15
4	Virtual Demonstration	16
4.1	Test Environment	16
4.1.1	GNS3	16
4.1.2	Oracle VirtualBox	17
4.1.3	Cisco IOS-XRV	17
4.2	Network Creation in GNS3	18
4.2.1	Setup of GNS3 GUI	19
4.2.2	VirtualBox and GNS3 VM	19
4.2.3	Importing Cisco IOS devices	22
4.3	Network Configuration	26
4.3.1	Basic Segment Routing configuration	26
4.3.2	Traffic Engineering configuration	29
4.3.3	Explicit and dynamic path configuration	32
4.3.4	Weighted ECMP configuration	34

List of Abbreviations

MPLS:	Multiprotocol Label Switching.
API:	Application Programming Interface
IP:	Internet Protocol.
SR:	Segment Routing.
SDN:	Software-defined networking.
LDP:	Label Distribution Protocol.
IGP:	Interior Gateway Protocol.
LSP:	Label-switched Path.
RSVP-TE:	Resource Reservation Protocol – Traffic Engineering.
TI-LFA:	Topology Independent Loop-Free Alternate.
ECMP:	Equal Cost Multipath.
FRR:	Fast Reroute.
FIB:	Forwarding Information Base.
OSPF:	Open Shortest Path First.
VPN:	Virtual Private Network.

GUI: Graphical User Interface.

FEC: Forwarding Equivalence Class.

VM: Virtual Machine.

KVM: Kernel-Based Virtual Machine.

SRH: Segment Routing Header.

1 Introduction

When new technology is being developed and after development, when it's being pitched the question asked is "Why, what is the benefit of this?"

Scalability and simplicity have always been major selling points when it comes to network architecture and technology in general. Segment Routing is a technology developed by IETF's SPRING Working Group that provides both of those aspects to a large network.

Segment routing is a modern take on Multiprotocol Label Switching (MPLS) and the major change between the two is simplification. [1.] [2.]

This Bachelor's Thesis is a personal project with the purpose of getting acquainted with Segment Routing technology and virtually demonstrating it. Tools used for the virtualization include: GNS3 as the network simulator, VirtualBox as the hypervisor, GNS3 VM as the virtual machine and Cisco IOS-XR and CSR1000v as the routers.

2 Software Designed Networking

There has been a swift growth in traditional IP networks during the past decades, as a result of the growth the networks have become less simple and harder to manage. Primary issue stems from the fact that modern networks use devices such as switches, routers and others, on these devices large amount of distributed protocols require implementation on top of using proprietary and closed interfaces. Therefore, traditional IP network leaves little if no room at all for third parties to innovate.

Modern networks have a significant lack of automation which makes them static and leads to them being unable to handle real time demands. Data and control plane are embedded in the same network node in a traditional IP network. This essentially means, that each network device has the responsibility of forwarding

decisions autonomously, due to control function being distributed over network devices. This was originally thought to be as a positive aspect due to it resulting in resilient network, but the issue is that whenever a change is made on the decentralized control plane, a manual change is required to be made on all of the network devices. [3, 1-2.]

2.1 Architecture of Software Designed Networking

In this segment we shall compare SDN-network to a traditional network and dive deeper into the SDN architecture. The figures below illustrate traditional network device (Figure 1) and software defined network (Figure 2).

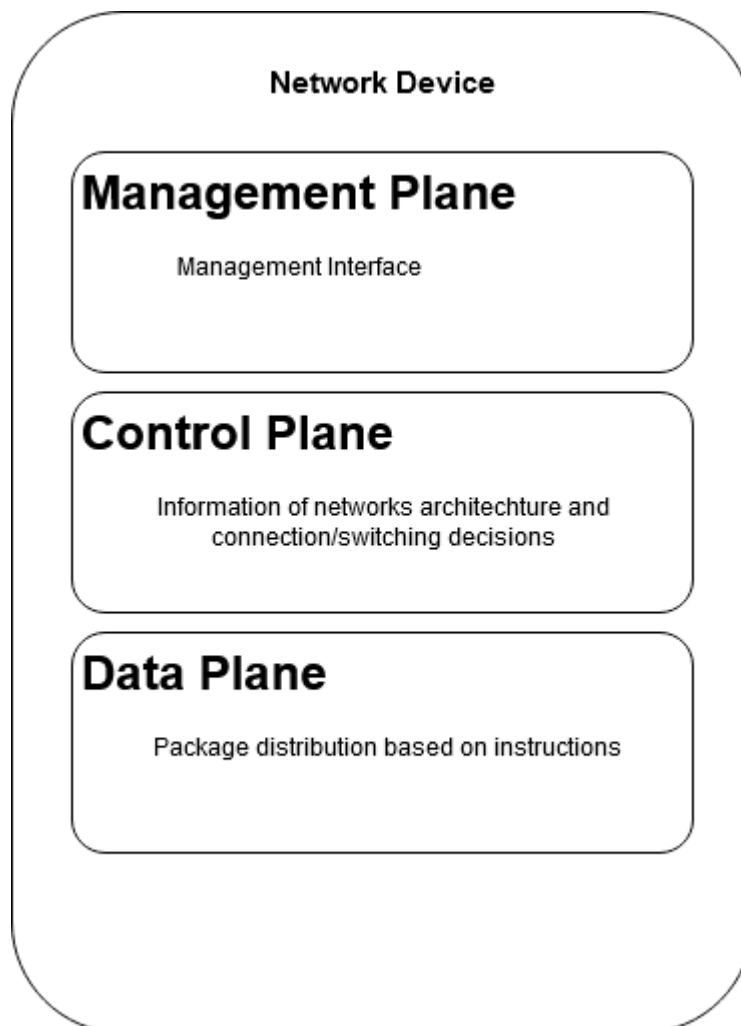


Figure 1. Traditional network device.

In a traditional network, each networking device has their own management interface that has been manually configured. This means that each network device does their own distribution choices based on what they have learned from their environment and how they have been configured.

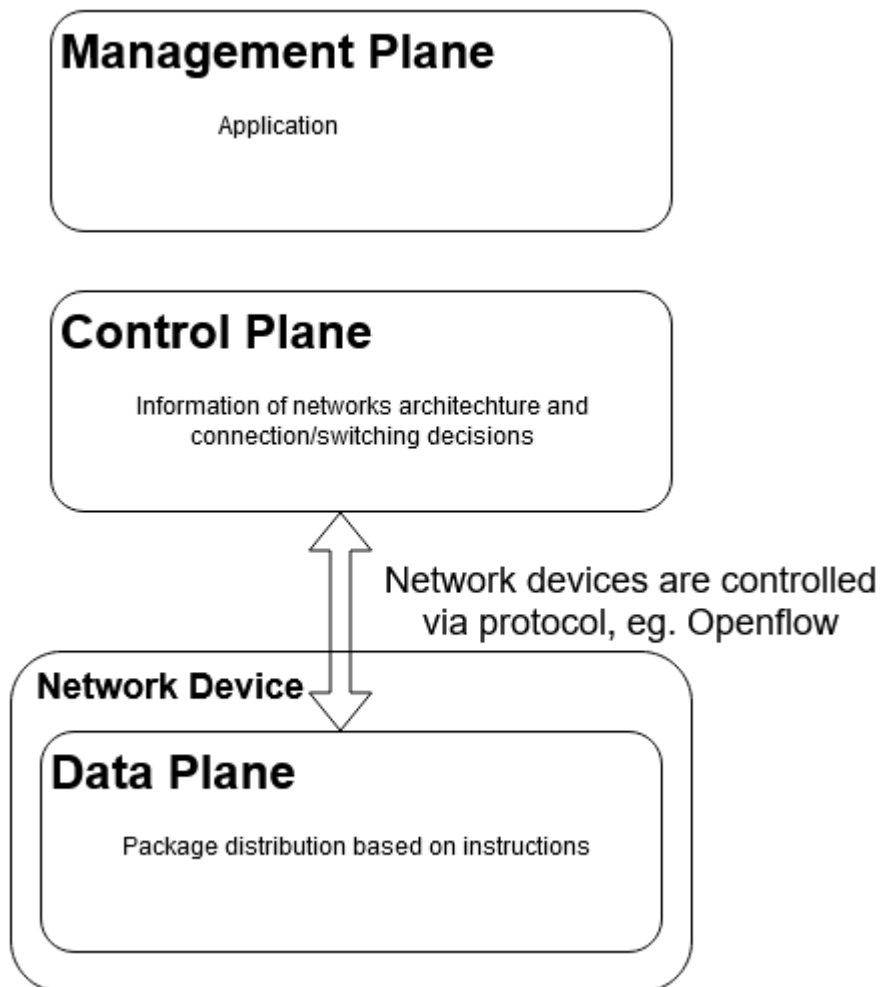


Figure 2. Software Defined Network.

One of the core ideas of SDN-networks is to separate packet distribution and configurations, in figure 2 these are depicted as data plane and control plane. In a nutshell, SDN network device is distributing packets based on instructions received from outside of the device. [4.]

2.2 Centralized Management with SDN

While it is possible to centralize the management without SDN, what SDN makes possible is automatic centralized configuration of the devices. When a network change is needed, it can be done via API-interface. When control layer receives information of the change, it can perform necessary configurations to the device automatically by using a protocol used for this, for example Segment Routing or OpenFlow. Figure 3 below describes a situation where network devices are being controlled by SDN.

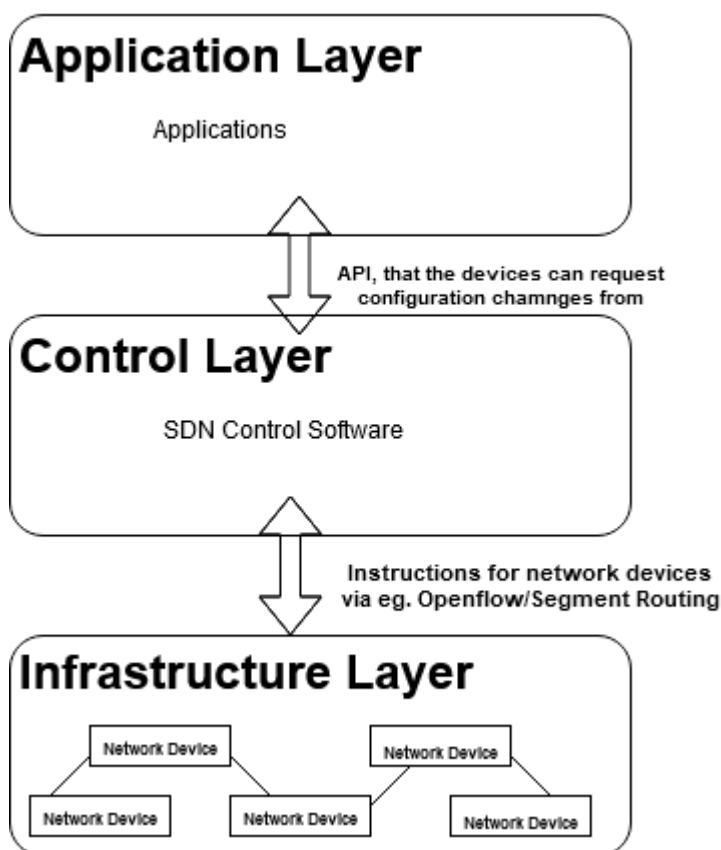


Figure 3. Network devices controlled by SDN.

Centralized management is beneficial in any network; however, the benefits will multiply in a larger environment. This is the way that large cloud service providers manage their networks. It would essentially be impossible to manage large environment without automated management. [5.]

We have been focusing on centralized, automated management, however it's not the only benefit that SDN brings. Another major benefit is that SDN makes network configuration changes faster. Without SDN you would require network specialists, who would manually sign into network devices and enter the configuration changes via terminal. With SDN there is no need for that, all the required changes can automatically be done to the necessary network devices. You can tell load balancers how to manage the traffic, make new rules on the firewalls, give routers information of new IP-networks, and give switches information of new devices. Thanks to this, there is no need to tell different experts what to do, each time a change is made to the environment.

This is essentially what cloud services do, they utilize SDN-architecture to automatically create all networks and network services that the customer requires. With SDN you can control both physical and virtual networks. These days virtual networks are not their own entity that lives their own life separated from the physical network devices, everything combines to a one complete environment now. [5.]

2.3 SDN with Segment Routing

Great combination of SDN and Segment routing may be what service providers will implement in the future. Processing business requirements and policies and translating them into SR paths is an easy task when your SDN controller has a global vision on the network. This opens the possibilities for service providers to optimize their networks and provide a number of new services.

As a platform for application engineered routing, SDN-SR allows specific path parameters to be required by an application as well as guiding the packets through the path and this required no communication with the network, giving mutual benefit to both network operation and application. [6.]

3 Segment Routing

In this segment I will dive into the couple of forwarding plane encapsulations that are being defined for segment routing: IPv6 with a Segment Routing Extension Header and Multiprotocol Label Switching (MPLS).

3.1 MPLS Routing Technique

Multiprotocol Label Switching (MPLS) is an almost 20-year-old technique used in routing that is generally classified as layer 2.5. It is most commonly used with IP but is compatible with any network protocol. When it comes to connectionless internet protocols, MPLS has increased capabilities in service creation and accomplished improvements in performance, including Quality of Service (QoS) in networks as well as Virtual Private Network (VPN) technologies. [7.]

MPLS networks attempt to eliminate delay that IP lookups in routing tables cause by routing the packets from network node to another appropriate to the MPLS labels which are 32-bit, this is important for voice, video and other high priority traffic. FEC is used to setup MPLS tunnels. One or two of the following signalling protocols are used to establish a tunnel, engineered by path calculation module: Label Distribution Protocol and Resource Reservation Protocol, which are known as RSVP and LDP, respectively. Every node on the route will fill up the MPLS routing table and reserve resources for a specific tunnel, based on the information being signalled. Packets will be processed by label edge router, when they come to the ingress of MPLS area. Edge router will attach a designated label to the packet based on policies set in place and pass it to the MPLS network. Whenever a packet is transit, received or when Label Switch Router checks the MPLS label, the label will be swapped to a new one based on the MPLS table and then the packet is forwarded to its next

destined router. When the packet is at the tunnels end, the final label will be removed, then the packet will be routed to its destination. [8, 14-17.]

3.2 Segment Routing comparison to MPLS

Let's start off our comparison by going through the technological features between SR and MPLS, based on the table below. [9.]

Technologies	MPLS	Segment Routing
50ms Fast Reroute (FRR)	IGP + RSVP-TE	IGP
Extra TE states for FRR	Extra states to manage	No extra state
ECMP-capability	No	Inbuilt
IGP/LDP synchronization	Required	Not Required
Link information	No	Through IGP domain
Operations & Troubleshooting	High	Low
Optimum backup path	No	Yes
Path Calculation	IGP + RSVP-TE	CSPF/PCE
Routing	Destination Based	Source Based
Scalability	Low	High
SDN Support	No	Yes
Interworks with classic MPLS	NA	Yes
TE state only at headend	No	Yes
MPLS Transport	IGP + LDP	IGP

Resource reservations and label signalling are accomplished via signalling protocols like LDP and RSVP-TE that exist in MPLS network. For SR network, IGP protocol is enough and after the configuration of SR, labels will be redistributed by IGP in the domain. It is not necessary to implement other signalling protocols, which will give major benefits in simplicity and bandwidth usage. Furthermore, when it comes to synchronization, there are some drawbacks with LDP when link failure happens, specifically LDP has to synchronize with IGP that will calculate a fresh array of shortest paths available in a network after the failure. It will take time for LSPs to stabilize, which can

end up with packet loss due to core routers not being able to forward the packets which are addressed to external networks. It's not necessary to synchronize with protocols besides IGP, since that is the only one used in Segment Routing. [10.]

For sensitive applications it is essential to have a great path protection. End-to-end path protection is a possibility in MPLS if both primary and secondary path are calculated, RSVP-TE protocol has to reserve both paths' resources.

Resource Reservation Protocol – Traffic Engineering is an extended version of RSVP used in MPLS to enact MPLS transport LSPs in case of traffic engineering demands. [11.]

The tunnel state must be maintained by routers that are in primary and secondary paths for guaranteed quality of service and preventing traffic loss, if a failure happens, rerouting will be swiftly done in under 50ms with FRR and the path is completely protected. Nonetheless, in busy core networks the efficiency of double resource reservation is not optimal. On the flip side, with Segment Routing IGP automatically calculates post-convergent path on link failure and new optimal path is guaranteed. Path protection requires no extra states to be maintained. Segment Routing uses TI-LFA as the FRR mechanism and with that a sub 50ms convergence is guaranteed. [12, 1-15.]

To balance the traffic among equal cost paths between destination and source, Equal Cost Multipath is used, which is built-in Segment Routing, different paths will be taken whenever two flows exist between invariable source and destination, for better network stability. ECMP is not supported by MPLS tunnels since hop-by-hop is used for their resolution. [13, 1-5.]

Source routing paradigm is used in Segment Routing, in comparison to MPLS, where destination IP address determines how labels are pushed to tunnel the packets. Constrained Shortest Path First (CSPF) algorithm determines the networks paths in Segment Routing, which is an extension of Shortest Path First (SPF) algorithm. SPF is run first and afterwards the constraints are being

applied e.g., available bandwidth and latency. [14, 2-7.] Typically external entities handle the path computation, for example Path Computation Element Protocol (PCEP) or SDN. RSVP-TE and IGP protocols are combined to setup paths in MPLS. Centralized data-plane network was the goal when building Segment Routing. Manual tunnel building is technically possible; however, SR is completely supported by SDN paradigm. On the other hand, in MPLS the control plane is distributed, and maintenance and setup of paths can be done by utilization of distributed protocols, which makes centralized control incredible hard if not difficult. In comparison to MPLS the scalability is massive in Segment Routing, necessity of signalling protocols is erased which will simplify the architecture and this means cost savings in hardware. On top of this Segment Routing decreases the amount of FIB entries, approximately each node will possess $n-1$ + the number of interfaces. On the contrary, any intermediate router will have n^2 number of entries in MPLS and this causes problems with scalability in large networks. [15, 48-63.] In conclusion, Segment Routing offers simplification for the network overall and reduces maintenance needs. No signalling protocols are needed, and data plane is simplified and constant labels over the network makes operating the network easy.

3.3 Segment Routing Architecture

As we have touched on, SR is a modern source routing archetype, it's a technology which aims to fix issues with simplicity, scalability and ease of operation in traditional IP/MPLS networks. SR is essentially the engine to application engineered routing, which is a business format which enables applications to control the behaviour of a network. Internet Engineering Task Force is standardizing Segment Routing under SPRING group. [16, 1-2.]

Packet-Forwarding behaviour is enhanced by Segment Routing, this allows packets to be carried by network by specific forwarding paths. The specific path may differ from the naturally shortest path that the packet traditionally takes in

the network. Some applications benefit from the possibility to control custom forwarding paths. While source-based routing is nothing new, there hasn't been a universal adaptation. A packet steering node that uses list of instructions in specific order is referred as a segment. For RSVP-TE and other modern traffic engineering solutions, signalling for every path is necessary and the presence of every paths state on every node that traffic passes through is required. This causes unnecessary complexity and issues with scalability that Segment Routing fixes by implementing these without needing signalling protocol, since SR uses MPLS data plane, neither LDP nor RSVP-TE are necessary. IGP is utilized for label distribution, either BGP and OSPF or Intermediate System-to-Intermediate System. Stability and scalability are improved by having less protocols inside the network. If a node/link failure happens the traffic will be rerouted in less than 50 milliseconds due to Segment Routing paths being protected by FRR. In traditional networks the traffic is mostly guided in accordance with destination IP by the routers. Distribution of network topology and the computation of shortest path from ingress node to egress node was done by utilizing IGP, yet these days available bandwidth, delay, jitter and packet loss have risen to be significant business differentiators in establishment of service level agreements, thus flexibility, combined with agility are looked for in networks by business requirements. Traffic steering functions and tunnelling mechanisms were introduced by MPLS, this is what brought success to MPLS as a technology. MPLS based Virtual Private Networks were introduced by MPLS, which was more popular than RSVP-TE, mostly due to poor load balancing, scalability issues and unpredictable traffic patterns and misuse of resources caused by distributed computation. [8, 44-47; 17, 2-23.]

Segment Routing as a technology has vast target audience that includes but is not limited to: Data centres, large enterprises, over-the-top providers, content providers and ISPs. Of course, SR still needs protocols (BGP, BGP-LS, IGP, PCEP, MPLS with SR) and tools (SDN controller) in the network. Global vision of a network is provided by a SDN controller as well as communication of messages and commands between network devices, while also serving as a middleman in between an application and network devices. For extraction of link

state information IGP and BGP-LS protocols are needed, the data is composed of links bandwidth, delay, metric and others. The data in question can always be accessed by the SDN controller which also means by high-level application. PCEP is necessary for path calculation and directing the path to network nodes by utilizing SDN controller.

Traffic engineering can be enabled with Segment Routing in three different ways:

1. Segment Routing tunnel creation dynamically. Path Computation Element (PCE) calculates the path with SLA or existing network information.
2. Segment Routing tunnel creation manually. PCE calculates the path, which is later pushed to the network by SDN controller.
3. Segment Routing Label Switched Path creation manually, with routes being defined inside the network. This is comparable with MPLS-TE, but there is no need for added protocols.

Segment Routing can hardly be called a new technology in 2022, however the usage by major network hardware vendors has only gradually been increasing. Concisely SR is a packet forwarding technology in which the source node defines traffic path, which is then sent through certain nodes and segments that are forwarding paths. Hop-by-hop signalling dependency is not prevalent in SR paths, RSVP or LDP, since segments are used for forwarding. [18, 2-22.]

3.3.1 Segments

Segment is something you would call an instruction for node to follow on an inflowing packet. For example, the instruction could be to forward a packet via set interface or to convey a packet to specified application or service. Segment

identifier exists to identify the segment and it's encoded in 32-bits MPLS label in MPLS environment. [19.]

3.3.2 Advertisement of Segments

BGP and IGP routing protocols are utilized to advertise the segments, in case of both protocols the SR extensions are to encompass Segment Routing information. This means that in the network segments signalling is enabled by these routing protocols. There is an OSPF or IS-IS running in each IGP area, in an autonomous system that consists of IGP areas, they advertise the segments inside an IGP domain. Although, segment exchanging has to be enabled between BGP peers to implement traffic engineering between autonomous systems. Segments related to the BGP-prefix will be advertised by BGP. SDN was a key technology when Segment Routing was designed, end-to-end paths are determined by SDN controller in software designed networks. BGP-LS informs SDN of the underlying network topology and is responsible of segment identifier advertisement to the SDN controller in a SDN with SR. SDN controller calculates a topological path that gets pushed to the source node as a list of segments and the path is transported with Path Computation Element Protocol. BGPLS and PCEP are both mandatory for Segment Routing in a Software Defined Network. [20.]

3.3.3 Local and Global Segments

Every segment in a network can be divided to local and global based on their significance in the network.

Global segments are always unique in the domain and nodes in IGP domain require the global segments in the FIB. Segment Routing Global Block (SRGB) will provide value for global segment identifiers, SRGB is a subspace of 32-bit security identifier space with values between 16000-23999.

Instruction supported by the node it came from, is what you would call local segment, their value will be outside of the previously mentioned SRGB range,

and its value will only be related to local routers FIB, since it solely has local significance. Routers in a domain have no awareness of each other's local segments, in addition since there is only local meaning for local SID values, they can be recycled within the IGP domain. [18, 2-22.]

3.3.4 IGP-SIDs – IGP Segment Identifiers

Segment Routing has major role for link state protocols, IGP is used to distribute the local and global segments in the domain. As a result of protocol extensions both IS-IS and OSPF, support Segment Routing.

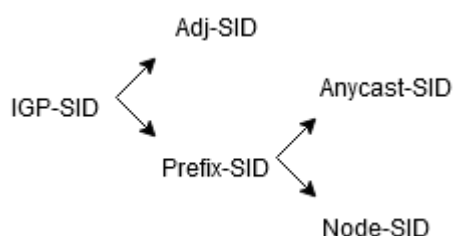


Figure 4. Segment Identifier Distribution.

Figure above visualizes how IGP distributed segment identifiers can be classified. [19.]

3.3.5 Segment Identifiers

Let's delve into the specifics of both Prefix-SID and Adjacency-SID. Starting off with Prefix-SID which refers to segment that points to a specific network prefix, it points to the shortest path which IGP computes to associated prefix. Within an IGP domain the Prefix-SID is global with no exceptions and will be transported to the ECMP best path to the related prefix. In an IGP domain the prefix can refer to a group of nodes or a singular node, thus as illustrated in figure 4, the Prefix-SIDs are divided in to Anycast-SIDs and Node-SIDs of which the next bullet points will go into:

- Anycast-SID is advertised by multiple nodes, typically two and a group known as anycastset is formed by nodes that advertise the same anycast-SID. Shortest path is provided to the nearest node in an anycast set by anycast-SID IGP. Figure 5 illustrates how the flows are load-balanced among the nodes when two or more nodes have the same distance. [21.]

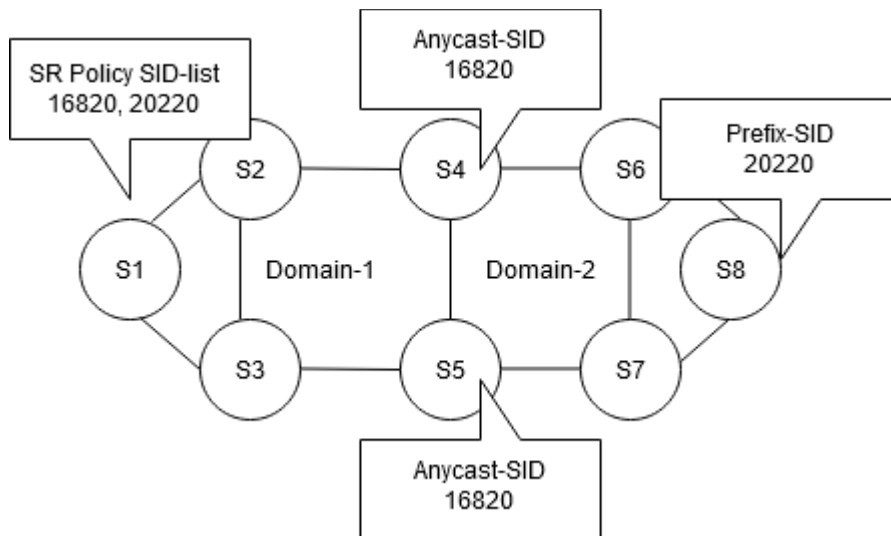


Figure 5. Anycast-SID topology.

- Node-SID identifies a specific node and is a prefix-SID. Loopback address of a node as the prefix will be used to configure the node-SID under the loopback interface. In segment routing domain, prefix-SID will be universally unique since the prefix segment is a global segment. [22, 1.]

Adjacency-SID which is commonly shortened to Adj-SID is an IGP-SID. In the IGP domain, Adj-SID will point to a specific link. Router will maintain Adj-SIDs only for its neighbours due to Adj-SIDs having local significance. The value of adjacency segments will always be outside of the SRGB range and traditionally the value will be allocated dynamically by a router. Adj-SIDs do not have to be unique in the segment routing domain, since they only have local significance, and they can conveniently direct traffic flow through particular interface. [23.]

Figure 6 demonstrates how the traffic flow direction works: the Adj-SID are assigned to R4s interfaces, and the values are outside of SRGB; 24001, 24002

and 24003. Should there be a need for link between R4 and R5, only the local label 24002 is needed to be pushed and the packet will be forwarded to next hop.

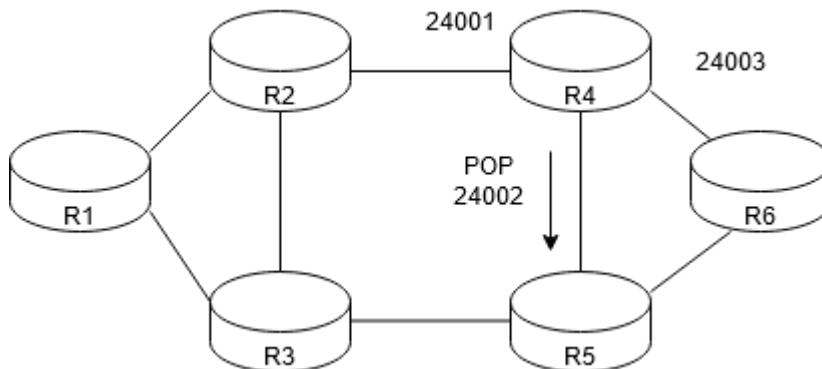


Figure 6. Adjacency-SID traffic flow directing.

3.3.6 Routing

Incoming traffic flow will be directed by the source node, which attaches an ordered SID list to a packet header and the first segment to be put into effect is the top segment. Once segment put into effect so when a packet arrives to the intermediate destination the following segment gets carried through. The flow reaches its destination or exits the segment routing domain and remains to be routed based on destination IP address when the last segment gets executed. SR capable nodes can perform three actions on segments, which are similar to operations executed on MPLS labels in MPLS networks, in SR they are:

1. PUSH: this operation consists of the insertion of the top segment. Top segment is the topmost label of the label stack in SR-MPLS.
2. NEXT: as the name implies, this operation embodies inspection of the next segment, when the active segment is completed, and the next segment becomes active. Next is executed as a Point of Presence (POP) on the topmost label in SR-MPLS.
3. CONTINUE: in SR-MPLS this operation is executed as a SWAP of the top label. Here the active segment won't be completed, thus it will stay active.

Figure 7 illustrates packet forwarding in segment routing domain. [18, 5-6.]

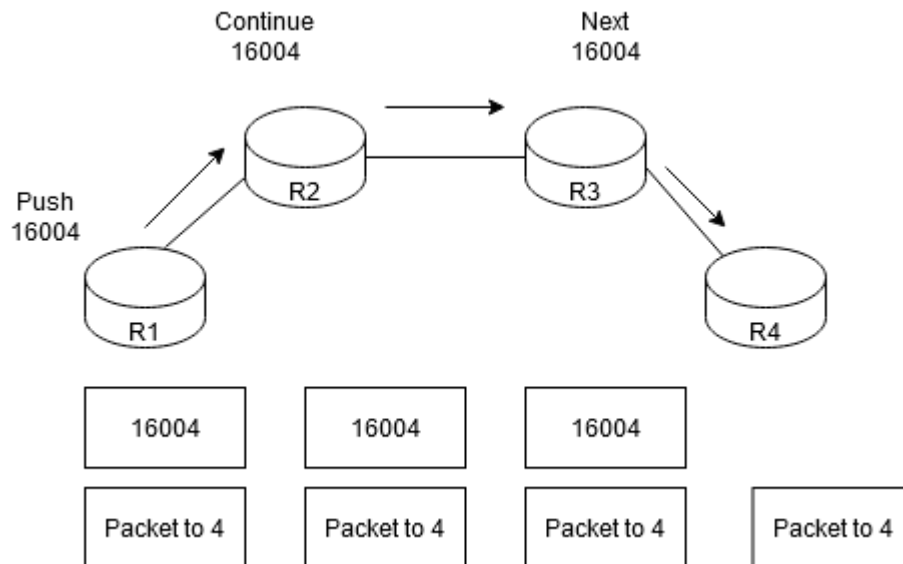


Figure 7. Packet forwarding in SR domain.

4 Virtual Demonstration

In this segment I will talk about different parts of the environment as well as going through how the environment was built.

4.1 Test Environment

I've used a virtualization software, network software emulator software and virtual Cisco devices, lets dive into them a bit more in depth.

4.1.1 GNS3

Graphical Network Simulator-3 more commonly referred as GNS3 is over a decade old open source, free network software emulator software that was released in 2008. In this project it's used as a base since it grants the possibility to simulate complex networks. With GNS3 you can also combine real and virtual devices. Cisco IOS is simulated with Dynamips emulation software by GNS3.

GNS3 is commonly used to prepare for professional network certification exams like CCNA, it's also used by multiple large companies. [25.]

4.1.2 Oracle VirtualBox

As the virtualization software I've decided to use Oracle VirtualBox which is a very popular open source and free software that sees usage in enterprises and regular homes. VirtualBox was first released in 2007 and is still receiving frequent updates and new features. [26.]

4.1.3 Cisco IOS-XRV

The Cisco IOS XRv Router is the router used in this virtual demonstration, it's a Virtual Machine based platform that runs a 32-bit IOS XR software with QNX microkernel. This means that it shares large amount of the same routing properties as physical IOS-XR devices. Including routing and forwarding functionality, control plane features and manageability the IOS XRv is a representation of IOS XR operating system and software. Figure 8 illustrates the architecture of IOS XRv router. It can be used on top of VMWare or Kernel-based VM hypervisors. [24.]

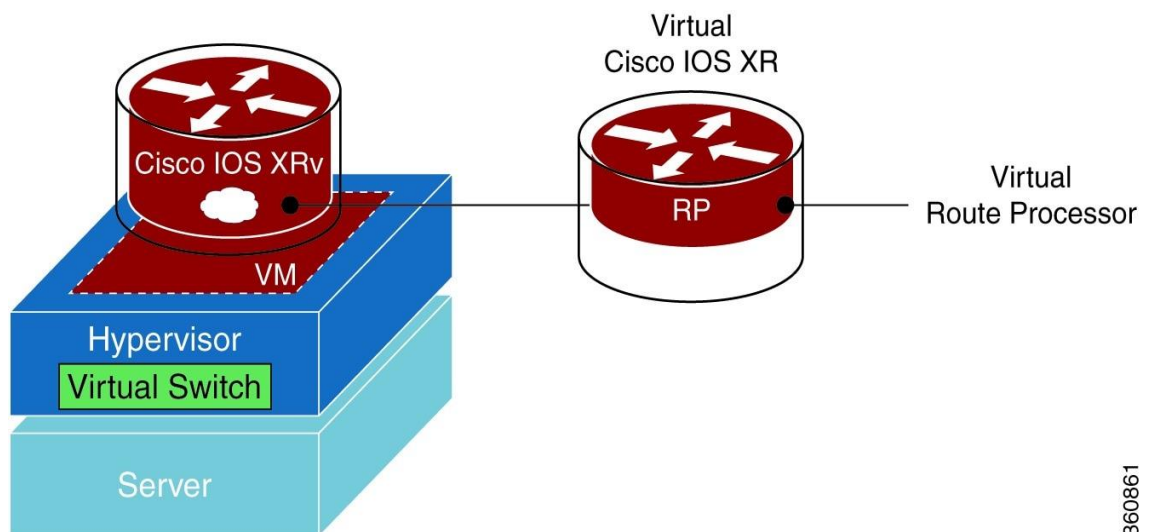
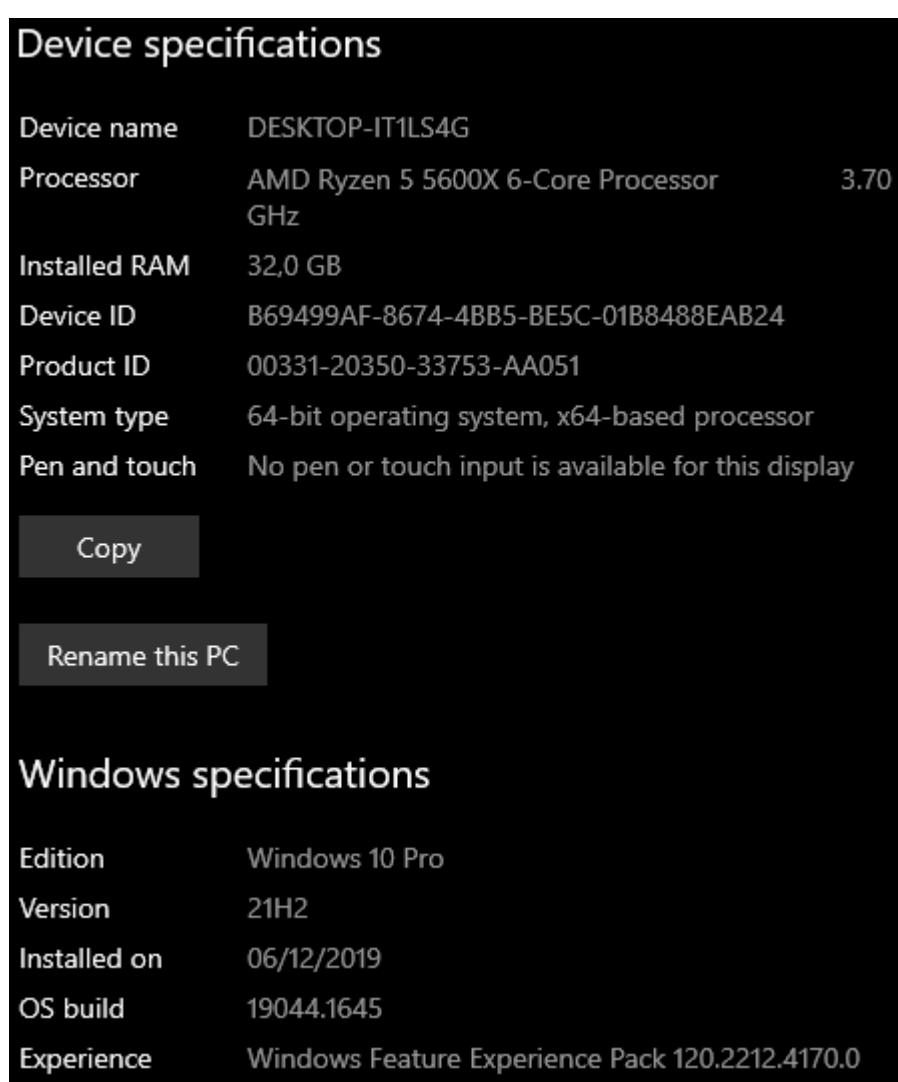


Figure 8. IOS-XRv Architecture. [24]

4.2 Network Creation in GNS3

This segment describes how the virtual environment was created and which software versions were used. The base system for everything was my own home PC with the system specifications shown in Figure 9. For my system it was necessary to enable Secure Virtual Machine from bios, for the virtualization to work.



The image shows a Windows system information window with a dark background. It is divided into two main sections: 'Device specifications' and 'Windows specifications'. The 'Device specifications' section lists hardware and system details, while the 'Windows specifications' section lists OS details. There are two buttons: 'Copy' and 'Rename this PC'.

Device specifications	
Device name	DESKTOP-IT1LS4G
Processor	AMD Ryzen 5 5600X 6-Core Processor 3.70 GHz
Installed RAM	32,0 GB
Device ID	B69499AF-8674-4BB5-BE5C-01B8488EAB24
Product ID	00331-20350-33753-AA051
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Copy

Rename this PC

Windows specifications	
Edition	Windows 10 Pro
Version	21H2
Installed on	06/12/2019
OS build	19044.1645
Experience	Windows Feature Experience Pack 120.2212.4170.0

Figure 9. Base system specifications.

4.2.1 Setup of GNS3 GUI

GNS3 was installed from the official GNS3 website [27] and the version was 2.2.32 at the time of the installation. Installation required a registration to the website, other than that it was a free download. As for the installation process I agreed to the license agreement and kept all default installation settings and components to install. GNS3 has a local server process running on my PC as illustrated in Figure 10.

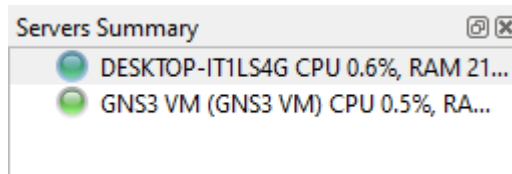


Figure 10. GNS3 Local server process.

4.2.2 VirtualBox and GNS3 VM

After the installation of GNS3 GUI the next step is to make the GNS3 run with VirtualBox. It is recommended to use GNS3 VM on Windows machines so the system will run GNS3 GUI and GNS3 VM. Hypervisor is needed to run the GNS3 VM which is a Linux based machine and that is what VirtualBox is used for. GNS3 VM was installed from the GNS3 official website [28] and the version at the time of the installation was 2.2.32. Oracle VirtualBox was installed from the official VirtualBox website [29] and the version was 6.1.32 at the time of the installation. For the VirtualBox installation I kept all the default installation settings. GNS3 VirtualBox VM came as a zip file that I extracted and imported the .ova file in VirtualBox by using VirtualBox's import appliance feature. For the Import I did not change any of the default settings which are shown in Figure 11.










 General	
Name:	GNS3 VM
Operating System:	Ubuntu (64-bit)
 System	
Base Memory:	2048 MB
Processors:	2
Boot Order:	Hard Disk, Optical
Acceleration:	VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization
 Display	
Video Memory:	8 MB
Graphics Controller:	VBoxVGA
Remote Desktop Server Port:	5966
Recording:	Disabled
 Storage	
Controller:	IDE Controller
IDE Primary Device 0:	GNS3 VM-disk001.vdi (Normal, 19,53 GB)
Controller:	SATA Controller
SATA Port 2:	GNS3 VM-disk002.vdi (Normal, 488,28 GB)
 Audio	
Host Driver:	Windows DirectSound
Controller:	ICH AC97
 Network	
Adapter 1:	Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')
Adapter 2:	Intel PRO/1000 MT Desktop (NAT)
 USB	
Disabled	
 Shared folders	
None	
 Description	
None	

Figure 11. GNS3 VirtualBox VM specifications.

After this I had to enable GNS3 VM in the GNS3 GUI preferences with the specifications shown in Figure 12. Essentially all changes and configurations will be made in the GNS3 GUI, however we still need to make sure that the virtual machine boots up properly and make sure that KVM is enabled in VirtualBox, which can be seen in Figure 13. KVM must be enabled for Cisco IOS devices to work.

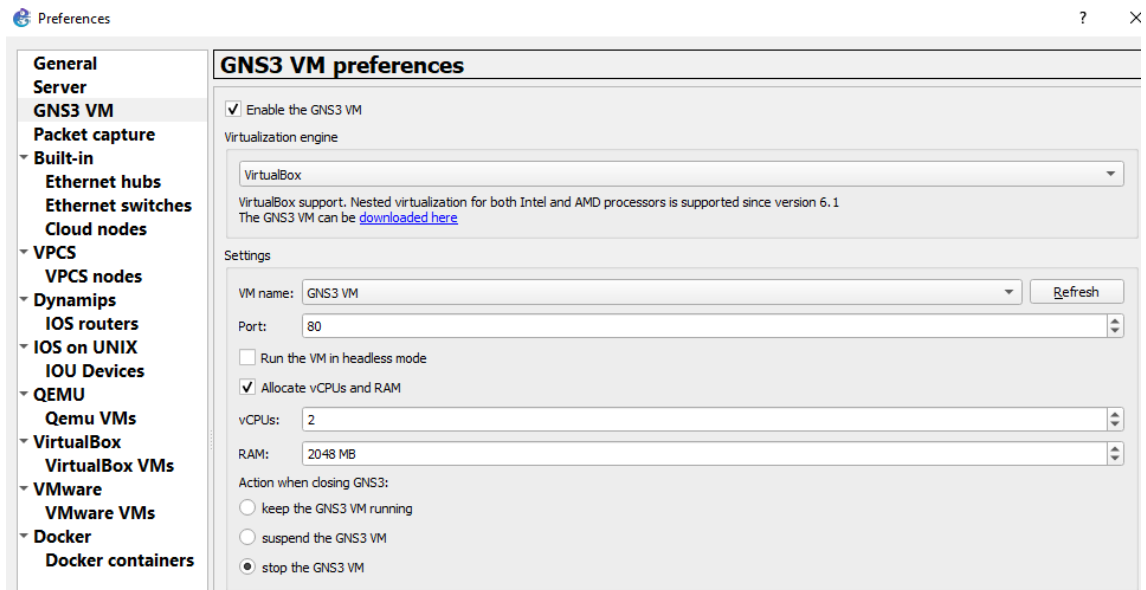


Figure 12. GNS3 VM specifications in GNS3 GUI.

```
GNS3 server version: 2.2.31
Release channel: 2.2
VM version: 0.13.0
Ubuntu version: focal
Qemu version: 4.2.1
Virtualization: virtualbox
kvm
KVM support available: True
Uptime: up 1 hour, 59 minutes

IP: 192.168.56.101 PORT: 80

To log in using SSH: ssh gns3@192.168.56.101
Password: gns3

To launch the Web-Ui: http://192.168.56.101

Images and projects are stored in '/opt/gns3'
```

< OK >

Figure 13. GNS3 VM running in VirtualBox with KVM enabled.

After the set up of the virtual machines I created a new blank project in the GNS3 GUI called Segment Routing Thesis. At this point there were no appliances in the GNS3 GUI, and I had to import them myself, which the next segment will cover.

4.2.3 Importing Cisco IOS devices

Cisco IOS devices can be easily acquired from Cisco VIRT; however, the service is expensive for a student, I was fortunate enough to receive the images from Metropolia. To import the devices as applications to the GNS3 I needed .gns3a files for them, which I downloaded for free from the GNS3 Marketplace in the appliances section [30]. After having the necessary files, it's possible to import them as appliances in GNS3 by using the "Import Appliance" feature. Following process uses the IOS XRv as an example. To start the import, I chose the previously downloaded gns3a file and chose "Install the appliance on the GNS3 VM" as the server type, as for the Qemu binary I used the default one which was /bin/qemu-system-i386 (v4.2.1). Unfortunately, GNS3 did not officially support the versions that I had in possession, so I had to create a new version. After the import process is completed, the imported appliance will be visible in the routers category as is shown in Figure 14 and in Figure 15 illustrates the appliance settings of Cisco IOS XRv 6.3.1 template.

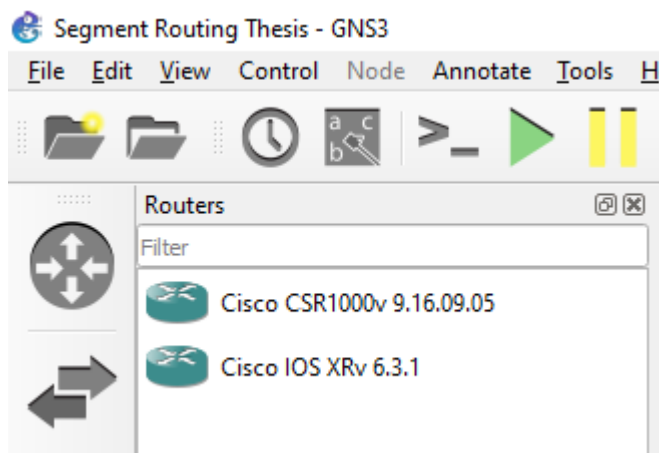


Figure 14. GNS3 Imported appliances.

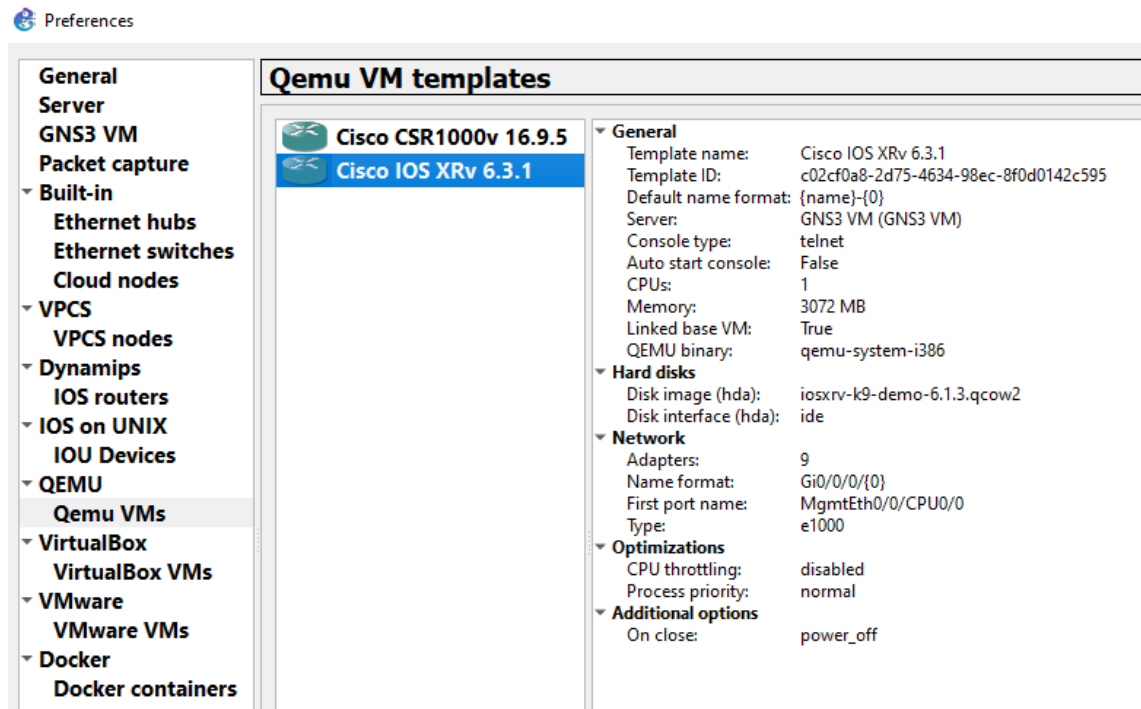


Figure 15. Cisco IOS XRv template settings.

With CSR1000v I experienced multiple issues, to begin with I had an iso image of the device which I had to convert to qcow2 file, so it could be used as Qemu VM. For the conversion I downloaded the qemu-img.exe software and did the conversion with following command: “.\qemu-img.exe convert csr1000v-universalk9.16.09.05.iso csr1000v-universalk9.16.09.05.qcow2”. Following this I imported the device to GNS3 the same way I previously did with IOS XRv and once again I stuck with the default settings. However, when I tried to start up the device and open the console, the console was blank. After some troubleshooting, I managed to fix the blank console by adding “-nographic” command to the additional options in the appliances template settings. While this fixed the blank console I was plagued by a new issue, the device could not find a bootable image. During my hours of troubleshooting, I tried multiple different fixes varying from trying all possible Qemu binaries to different additional options and even changing the network adapter types with no avail. Based on some research I found that a few other people had similar issues with CRS1000v when used in GNS3 with VirtualBox, however I was very reluctant on switching to another hypervisor at this point. I decided to attempt a different approach and created the

CSR1000v as a virtual machine directly in VirtualBox with the settings that are shown in figure 16.

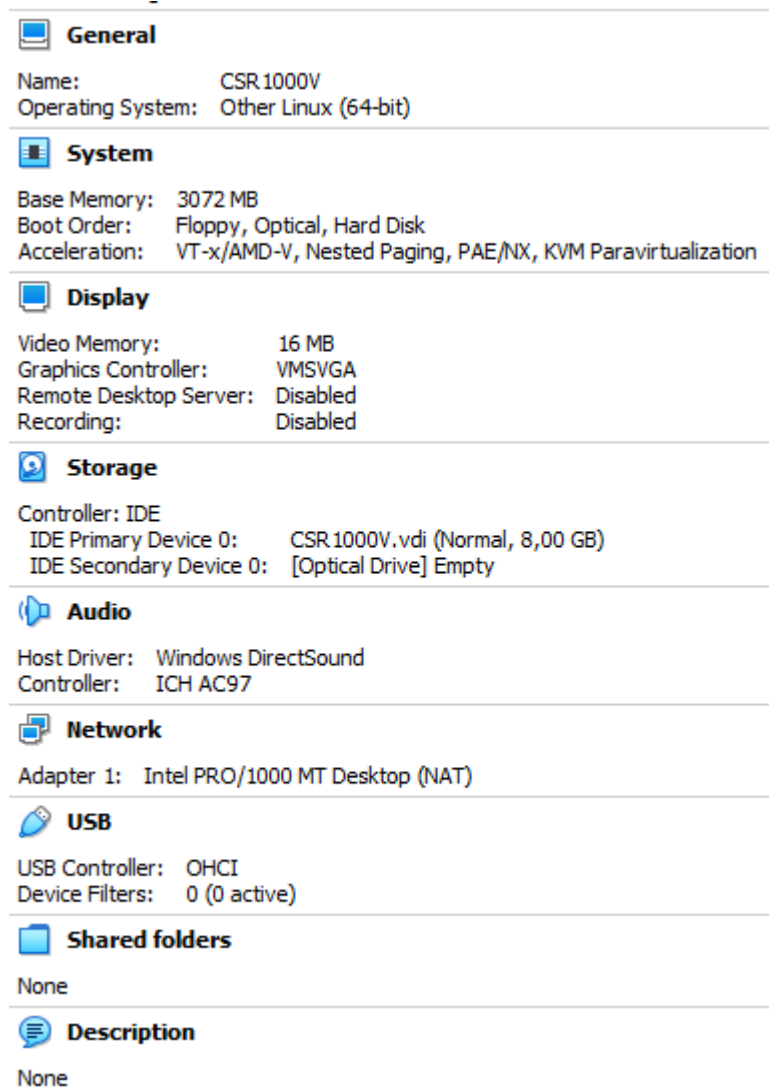


Figure 16. VirtualBox CSR1000v settings.

After creating the virtual machine, I imported it directly to GNS3 by doing the following steps, after these steps the VM template is as is shown in figure 17.

:

1. Edit > Preferences > VirtualBox VMs > New.
2. Run this VirtualBox VM on my local computer.
3. From VM list I chose the recently created CRS1000V and enabled “Use as a link base VM (experimental) option.”

4. Finish > Apply > Ok.
5. Edit > Preferences > VirtualBox VMs > CSR1000V > Edit.
6. Console type > telnet.
7. Ok > Apply > Ok.

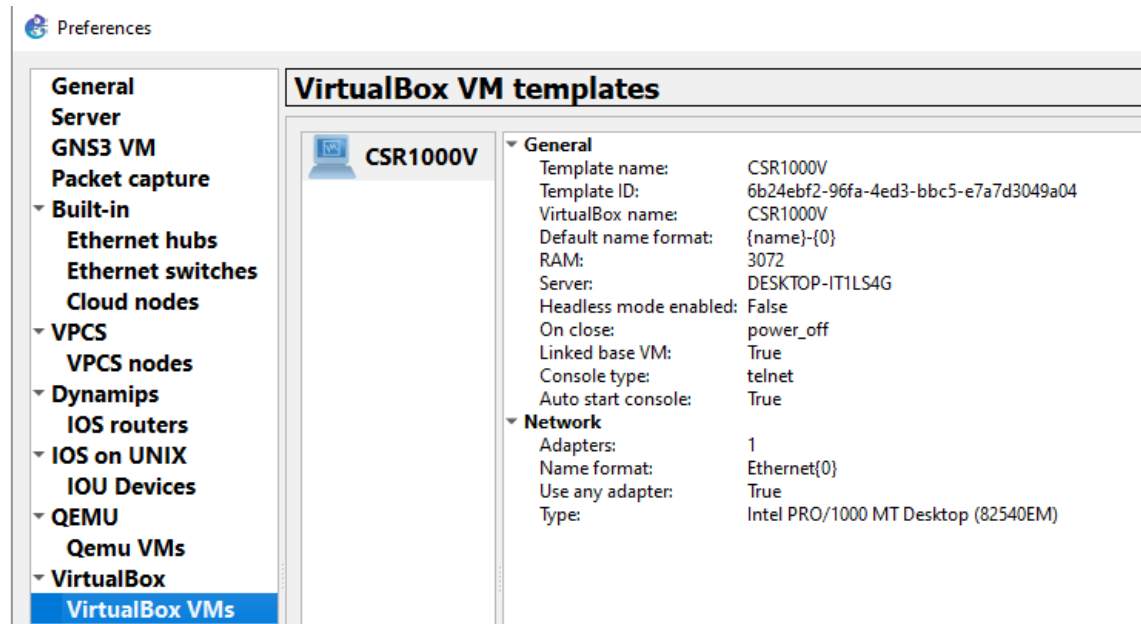


Figure 17. VirtualBox VM GNS3 template.

After this I had to activate the license to enable advanced features, this was done with the command “license boot level ax” followed by a save and reboot. Unfortunately, even with this method the console inside GNS3 did not work, however this way I could enter the configurations straight into the VM from VirtualBox. After this we finally have working appliances for both IOS-XRv and CSR1000v, which allowed me to build the topology shown in figure 18, which will be used for the rest of the demonstration. To add on to this when I began the configurations, there was a new problem with the CSR1000V. The OS did not recognize the network interfaces, this was fixed by changing the network adapter type to “Paravirtualized network”.

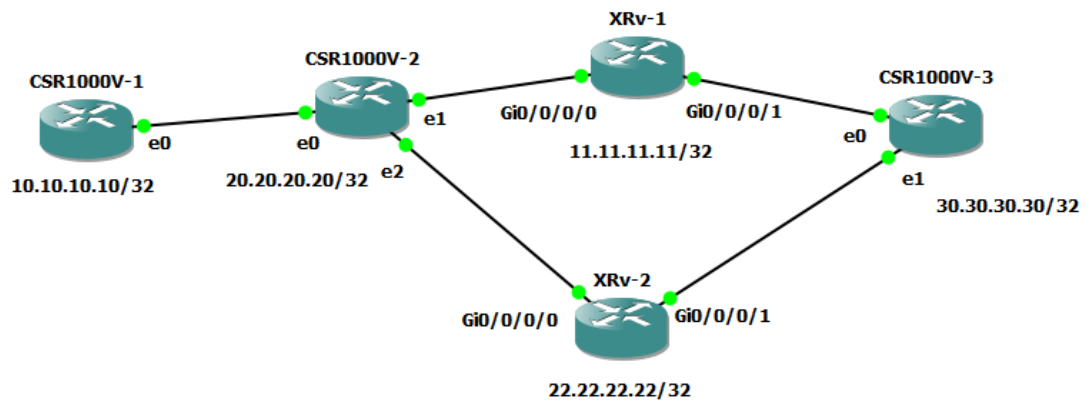


Figure 18. Topology of the virtual demonstration.

4.3 Network Configuration

As previously mentioned, the topology shown above will be used for this virtual demonstration. The actual configuration will be split in four parts which are: Basic segment routing configuration, traffic engineering configuration, explicit and dynamic path configuration and lastly weighted ECMP configuration.

4.3.1 Basic Segment Routing configuration

To start things off I configured only basic segment routing configurations on the devices. As an example, configuration of XRv-1 will be illustrated in figure 19 and the configuration of CSR1000V-1 will be shown in text form due to the terminal being cluttered visually in VirtualBox VM. After this part all configurations will be shown in text form for consistency.


```
hostname XRv-1
interface Loopback1
  ipv4 address 11.11.11.11 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
  shutdown
!
interface GigabitEthernet0/0/0/0
  ipv4 address 100.100.100.6 255.255.255.252
  shutdown
!
interface GigabitEthernet0/0/0/1
  ipv4 address 100.100.100.13 255.255.255.252
  shutdown
!
interface GigabitEthernet0/0/0/2
  shutdown
!
interface GigabitEthernet0/0/0/3
  shutdown
!
interface GigabitEthernet0/0/0/4
  shutdown
!
interface GigabitEthernet0/0/0/5
  shutdown
!
interface GigabitEthernet0/0/0/6
  shutdown
!
interface GigabitEthernet0/0/0/7
  shutdown
!
router isis lab
  is-type level-1
  net 49.0000.0000.0000.0003.00
  address-family ipv4 unicast
    metric-style wide
    segment-routing mpls
  !
  interface Loopback1
    address-family ipv4 unicast
      prefix-sid index 3
  !
  !
  interface GigabitEthernet0/0/0/0
    circuit-type level-1
    point-to-point
    address-family ipv4 unicast
  !
  !
  interface GigabitEthernet0/0/0/1
    circuit-type level-1
    point-to-point
    address-family ipv4 unicast
  !
!
end
```

Figure 19. XRv-1 Basic SR configuration.

CSR1000V-1 Configuration:

```
interface Loopback1
 ip address 10.10.10.10 255.255.255.255
 ip router isis LAB
!
interface GigabitEthernet0
 ip address 100.100.100.1 255.255.255.252
 ip router isis LAB
 negotiation auto
 no mop enabled
 no mop sysid
 isis circuit-type level-1
 isis network point-to-point
!
segment-routing mpls
!
connected-prefix-sid-map
 address-family ipv4
  10.10.10.10/32 index 1 range 1
 exit-address-family
!
!
router isis LAB
 net 49.0000.0000.0000.0001.00
 is-type level-1
 metric-style wide
 segment-routing mpls
!
```

4.3.2 Traffic Engineering configuration

When configuring traffic engineering in this topology, it's necessary to make the core facing interfaces point-to-point, since traffic engineering is not supported over broadcast interface on any Cisco IOS-XE routers when ISIS is used as IGP. Here we configured TE tunnels between the routers CSR1000V-1 and CSR1000V-3. First, I will show the configuration of the CSR1000V-1, which is the tunnel headend in this case:

```

!
mpls traffic-eng tunnels
!
interface GigabitEthernet1
 ip address 100.100.100.1 255.255.255.252
 ip router isis LAB
 mpls traffic-eng tunnels
 isis circuit-type level-1
 isis network point-to-point
!
router isis LAB
 net 49.0000.0000.0000.0001.00
 is-type level-1
 metric-style wide
 segment-routing mpls
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng level-1
!
segment-routing mpls
!
connected-prefix-sid-map
 address-family ipv4
  10.10.10.10/32 index 1 range 1
 exit-address-family

```

```
!  
interface Tunnel1  
  ip unnumbered Loopback1  
  tunnel mode mpls traffic-eng  
  tunnel destination 30.30.30.30  
  tunnel mpls traffic-eng autoroute announce  
  tunnel mpls traffic-eng priority 6 6  
  tunnel mpls traffic-eng path-option 10 explicit name SR-PATH segment-routing  
!  
ip explicit-path name SR-PATH enable  
index 1 next-label 16002  
index 2 next-label 16004  
index 3 next-label 16005  
!
```

And following this here is the configurations of CSR1000V-3, which is the tunnel tail end:

```
!  
mpls traffic-eng tunnels  
!  
interface GigabitEthernet1  
  ip address 100.100.100.14 255.255.255.252  
  ip router isis LAB  
  mpls traffic-eng tunnels  
  isis circuit-type level-1  
  isis network point-to-point  
!  
!  
interface GigabitEthernet2  
  ip address 100.100.100.18 255.255.255.252  
  ip router isis LAB  
  mpls traffic-eng tunnels  
  isis circuit-type level-1
```

```
isis network point-to-point
!
router isis LAB
 net 49.0000.0000.0000.0005.00
 is-type level-1
 metric-style wide
 segment-routing mpls
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng level-1
!
segment-routing mpls
!
connected-prefix-sid-map
 address-family ipv4
  30.30.30.30/32 index 5 range 1
 exit-address-family
!
!
interface Tunnel1
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 10.10.10.10
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 10 explicit name SR-PATH segment-routing
!
ip explicit-path name SR-PATH enable
 index 1 next-label 16004
 index 2 next-label 16002
 index 3 next-label 16001
!
```

4.3.3 Explicit and dynamic path configuration

In this part traffic engineering will be configured between CSR1000V-1 and XRv-2 routers. To start off we need to bring up the tunnels, here are the configurations for CSR1000V-1:

```

!
interface Tunnel2
 ip unnumbered Loopback1
 tunnel mode mpls traffic-eng
 tunnel destination 22.22.22.22
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng path-option 20 explicit name SR-PATH-XRv2 segment-
routing
!
ip explicit-path name SR-PATH-R4 enable
 index 1 next-label 16002
 index 2 next-label 16004
!

```

And the configurations for XRv-2:

```

segment-routing
traffic-eng
 segment-list SR-PATH-CSR1000V1
  index 1 mpls label 16002
  index 2 mpls label 16001
!
policy POL1
 color 20 end-point ipv4 10.10.10.10
 candidate-paths
!
 preference 100

```

```

explicit segment-list SR-PATH-CSR1000V1
!
!

```

After these configurations there will be functioning bidirectional TE path between the routers. After the tunnel is up, traffic can be put on the tunnel. Now I shall show what the configuration is like on the XR router in case of dynamic TE tunnel:

```
RP/0/0/CPU0:XRv-2#sh run router isis
```

```
router isis LAB
```

```
is-type level-1
```

```
net 49.0000.0000.0000.0004.00
```

```
distribute link-state
```

```
!
```

```
RP/0/0/CPU0:XRv-2#sh run segment-routing
```

```
!
```

```
segment-routing
```

```
traffic-eng
```

```
!
```

```
policy POL1
```

```
color 20 end-point ipv4 10.10.10.10
```

```
candidate-paths
```

```
!
```

```
preference 101
```

```
dynamic
```

```
metric
```

```
type te
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

4.3.4 Weighted ECMP configuration

At this point we will configure WECMP on the routers. At this point we have two paths between the routers CSR1000V-1 and XRv-2, these paths will be used in WECMP in 1/3 to 2/3 distribution. This means out of 3 packets one will go to one path and two to the other. Here is the configuration in XRv-2:

```
RP/0/0/CPU0:XRv-2#sh run router isis
router isis LAB
  is-type level-1
  net 49.0000.0000.0000.0004.00
  distribute link-state
  address-family ipv4 unicast
  metric-style wide
  mpls traffic-eng level-1
  mpls traffic-eng router-id Loopback1
  segment-routing mpls
!
interface Loopback1
  address-family ipv4 unicast
  prefix-sid index 4
!
!
interface GigabitEthernet0/0/0/0
  circuit-type level-1
  point-to-point
!
!
interface GigabitEthernet0/0/0/1
  circuit-type level-1
  point-to-point
!
!
```



```

!

RP/0/0/CPU0:XRv-2#sh run segment-routing
segment-routing
traffic-eng
  segment-list SR-PATH-1-R1
  index 1 mpls label 16002
  index 2 mpls label 16001
!
  segment-list SR-PATH-2-R1
  index 1 mpls label 16005
  index 2 mpls label 16003
  index 3 mpls label 16002
  index 4 mpls label 16001
!
  policy POL1
  color 20 end-point ipv4 10.10.10.10
  candidate-paths
  preference 100
  explicit segment-list SR-PATH-1-R1
  weight 4
!
  explicit segment-list SR-PATH-2-R1
  weight 2
!
!
!

```

On CSR1000V-1 the configurations are the same as in previous part, however a single command is needed for specification of weight and SRLIST:

```

“CSR1000V-1(config-srte-policy-path-pref)#explicit segment-list SR-LIST weight
2”

```

5 Conclusion

This project's goal was for me to familiarize myself with a virtualized testing environment and segment routing as a technology, which both were achieved.

There were challenges with the virtualized demonstration environment that I chose, however I managed to achieve desirable end results. But if I were to do another virtual demonstration of segment routing, I would consider other options, at the very least use VMWare over VirtualBox in GNS3, since based on my research it tends to support Cisco appliances better.

When it comes to segment routing itself the configurations were simple and easy implement on the devices. However, that is the extent of this demonstration, for further testing of segment routing I would build a separate MPLS environment that's performance could directly be compared with the segment routing environment. But that would be on a far larger scale.

As I earlier touched upon, I managed to familiarize myself with segment routing as well as GNS3 as a platform and during this process I did learn many things. Most important lesson would be to do proper research on compatibilities of different technologies and tools that I will use in a project, before committing to the project.

References

- 1 Adrian Farrel, Ron Bonica. 2017. Segment Routing: Cutting Through the Hype and Finding the IETF's Innovative Nugget of Gold. <https://www.ietfjournal.org/segment-routing-cutting-through-the-hype-and-finding-the-ietfs-innovative-nugget-of-gold/>. Accessed 20 May 2021.
- 2 Ozgur Gule. 2018. Why should I do Segment Routing? <https://343544.medium.com/why-should-i-do-segment-routing-51db5c3fe121>. Accessed 25 May 2021.
- 3 Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky and Steve Uhlig, Software-Defined Networking: A Comprehensive Survey, IEEE. Accessed 04 April 2022.
- 4 Connor Craven, What is Software Defined Networking (SDN)? Definition. <https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/>. Accessed 04 April 2022.
- 5 What is Software Defined Networking (SDN)? <https://www.vmware.com/topics/glossary/content/software-defined-networking.html>. Accessed 05 April 2022.
- 6 Stephen Liu. 2013. Segment Routing: Impact on Software Defined Networks. <https://blogs.cisco.com/sp/segment-routing-impact-on-software-defined-networks>. Accessed 05 April 2022.
- 7 Josh Fruhlinger. 2022. What is MPLS, and why isn't it dead yet? <https://www.networkworld.com/article/2297171/network-security-mpls-explained.html>. Accessed 05 April 2022.
- 8 Arun Viswanathan, Eric Rosen, Ross Callon. 2001. Multiprotocol Label Switching Architecture. Accessed 06 April 2022.
- 9 Shivlu Jain. 2015. Segment Routing Based MPLS Vs Classic MPLS. <http://www.mplsvpn.info/2015/07/segment-routing-based-mpls-vs-classic.html>. Accessed 06 April 2022.
- 10 2011. LDP-IGP Synchronization in MPLS Networks. <https://blog.ip-space.net/2011/11/ldp-igp-synchronization-in-mpls.html>. Accessed 07 April 2022.
- 11 2020. Cisco. MPLS Traffic Engineering—Fast Reroute Link and Node Protection. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mp_te_path_protect/configuration/xr-16-11/mp-te-path-protect-xr-

[16-11-book/mpls-traffic-engineering-fast-reroute-link-and-node-protection.html](#). Accessed 08 April 2022.

- 12 C. Filsfils, P. Francois, M. Shand, B. Decraene, J. Uttarom, N. Leymann, M. Horneffer. 2012. Loop-Free Alternate (LFA) Applicability in Service Provider (SP) Networks. Accessed 09 April 2022.
- 13 C. Hopps. 2000. Analysis of an Equal-Cost Multi-Path Algorithm. Accessed 15 April 2022.
- 14 Manayya K.B. 2009. Constrained Shortest Path First. Accessed 15 April 2022.
- 15 Daniel Amyot, Pau Fonseca i Casas, Gunter Mussbacher. 2014. System Analysis and Modeling: Models and Reusability. Accessed 16 April 2022.
- 16 S. Previdi, C. Filsfils, B Decraene, S. Litkowski, M. Horneffer, R. Shakir. 2016. SPRING Problem Statement and Requirements draft-ietf-spring-problem-statement-08. Accessed 16 April 2022.
- 17 Zheng Ma, Hao Wang, Yang Richard Yang. 2006. Traffic engineering in MPLS and VPN networks. Accessed 16 April 2022.
- 18 Clarence Filsfils, Stefano Previdi, Les Ginsberg, Bruno Decraene, Stephane Litkowski, Rob Shakir. 2018. Segment Routing Architecture. Accessed 17 April 2022.
- 19 Ahmed Bashandy, Clarence Filsfils, Stefano Previdi, Bruno Decraene, Stephane Litkowski, Rob Shakir. 2019. Segment Routing with the MPLS Data Plane. Accessed 17 April 2022.
- 20 S. Previdi, C. Filsfils, D. Jain, P. Mattes, E. Rosen, S. Lin. 2018. Advertising Segment Routing Policies in BGP. Accessed 17 April 2022.
- 21 Dinesh Bathla. 2020. Anycast-SID in Segment Routing. <https://segmentroutingexpert.com/2020/05/29/anycast-sid-in-sr/>. Accessed 23 April 2022.
- 22 Cisco. About Segment Routing. https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/segment-routing/61x/b-segment-routing-cg-ncs5500/b-segment-routing-cg-ncs5500_chapter_01.pdf. Accessed 23 April 2022.
- 23 Juniper. 2021. Understanding Adjacency Segments, Anycast Segments, and Configurable SRGB in SPRING. <https://www.juniper.net/documentation/us/en/software/junos/is-is/topics/concept/understanding-support-for-spring-anycast-and-prefix-segments.html>. Accessed 23 April 2022.

- 24 Cisco. Cisco IOS XRv Router Overview. https://www.cisco.com/en/US/docs/ios_xr_sw/ios_xrv/install_config/b_xrvr_432_chapter_01.html. Accessed 02 May 2022
- 25 GNS3. Getting Started with GNS3. <https://docs.gns3.com/docs/>. Accessed 02 May 2022.
- 26 VirtualBox. Welcome to VirtualBox.org! <https://www.virtualbox.org/>. Accessed 03 May 2022.
- 27 GNS3. <https://gns3.com/software>. Accessed 04 May 2022.
- 28 GNS3. <https://gns3.com/software/download-vm>. Accessed 04 May 2022.
- 29 VirtualBox. <https://www.virtualbox.org/wiki/Downloads>. Accessed 04 May 2022.
- 30 GNS3. <https://www.gns3.com/marketplace/appliances>. Accessed 05 May 2020.