Samu Varkama

MPLS Segment Routing Technology Study

Opinnäytetyö

Tietotekniikka

Toukokuu 2017

XAMK

Kaakkois-Suomen ammattikorkeakoulu

XAMK
Kaakkois-Suomen
ammattikorkeakoulu

Tekijä/Tekijät	**Tutkinto**	**Aika**
Samu Varkama	Insinööri (AMK)	Huhtikuu 2017

Opinnäytetyön nimi MPLS Segment Routing Technology Study	50 sivua 4 liitesivua

Toimeksiantaja

Kymenlaakson ammattikorkeakoulu

Ohjaaja

Lehtori Vesa Kankare

Tiivistelmä

Tämän opinnäytetyön tavoitteena oli tutkia segment routing -tekniikkaa Virtual Laboratory -ympäristöön asennetuilla reitittimillä. Virtuaalisuuden ansiosta fyysisiä laitteita ei tarvita harjoitusten tekemiseen, vaan kaikki voidaan suorittaa selaimen välityksellä. Lopuksi tutkimusten pohjalta luotiin segment routing -teknologiaan liittyviä harjoituksia, joita voidaan hyödyntää tulevilla oppitunneilla.

Työssä käytettiin IOS-XRv 6.0.1 -virtuaalireititintä virtuaaliympäristössä. Virtuaaliympäristöön kuului kahdeksan reititintä, neljä kytkintä ja viisi tietokonetta. Suurin osa ajasta kului eri ominaisuuksien kartoittamiseen. Perusteorian lisäksi mapping server -teknologia, kuormanjako, palveluiden eriyttäminen, anycast-SID ja segment routing traffic engineering olivat tärkeitä tutkimuksen kohteita.

Myös PCEP-tekniikka nousi tärkeäksi tekijäksi opinnäytetyön aikana. Tästä syystä jouduttiin ottamaan käyttöön yksi tietokone, johon PCE-kontrolleri saatiin asennettua. Täyttä potentiaalia PCE-kontrollerista ei onnistuttu saamaan irti, mutta opinnäytetyössä käydään sen perusominaisuudet läpi.

Segment routing -tekniikka on vielä suhteellisen uusi ja siihen tulee jatkuvasti uusia ominaisuuksia päivitysten kautta. Jo tätä opinnäytetyötä tehdessä IOS-XRv:stä on tullut uusia versioita, joissa konfiguraatiomahdollisuudet ovat paremmat. Myös PCEP-tekniikka etenee suurin harppauksin. XTC-ominaisuus siirtää PCE-kontrollerin tehtävät tietokoneelta suoraan reitittimille, mikä helpottaa työskentelyä huomattavasti.

Opinnäytetyö oli onnistunut. Kaikkia tehtävänannossa mainittuja teknologioita pystyttiin tutkimaan ja niiden pohjalta toteuttamaan toimivia harjoituksia. PCEP-teknologian osalta kaikkia asioita ei voitu toteuttaa ohjelmiston rajoituksista johtuen. Teknologian kehityksestä johtuen tässä aiheessa on vielä paljon tutkittavaa, varsinkin jos IOS-XRv-laitteesta saadaan uusin versio.

Asiasanat

segment routing, MPLS, operaattoriverkot, SDN

Author (authors)	Degree	Time
Samu Varkama	Bachelor of Engineering	April 2017

Thesis Title

MPLS Segment Routing Technology Study

50 pages

4 pages of appendices

Commissioned by

Kymenlaakso University of Applied Sciences

Supervisor

Vesa Kankare, Senior Lecturer

Abstract

The goal of this thesis was to study segment routing technology in the virtual laboratory environment using virtual routers. Virtualization makes it easier to study multiple routers and make topology changes into the network. The final goal of the study was to create exercises for future segment routing lessons. The exercises are can be found in the appendices section of the thesis.

IOS-XRv 6.0.1 was the router image used in the virtual laboratory. The virtual laboratory consisted of eight routers, four switches and one computer. The most time-consuming part of the thesis was to find all the technologies that are available for the router image that was used. Mapping server, load-balancing, disjointness, anycast-SID and segment routing traffic engineering were the primary subjects in the thesis.

PCEP technique also had an important role in the thesis. The PCE server was installed on a computer that was part of the virtual network. All features of the PCE server could not be tested during this thesis work so there is room for more research on the subject. However, the basic PCE functionality was successfully tested.

Segment routing is still a relatively new technology. This means that it is frequently being updated and information becomes outdated quickly. At the time of writing this thesis, a new version of IOS-XR that simplifies the configuration and adds more options is already available. PCEP technology has also received updates. XTC feature moves the PCE controller into the router itself, which should make the configuration easier.

The thesis successfully explored all the topics that were required and case studies were made based on the research. The PCEP protocol has still more room for research, since all the features were not available in the current versions of the software. The new IOS-XRv version will make it possible to study the XTC and other new features.

Keywords

Segment routing, MPLS, service provider networks, SDN

CONTENTS

ABBREVIATIONS

CSPF	Constrained Shortest Path First
ECMP	Equal-cost multi-path
FEC	Forwarding equivalence class
IETF	Internet Engineering Task Force
LSP	Label-switched path
LSR	Label switch router
MPLS	Multiprotocol Label Switching
OSPF	Open shortest path first
PCE	Path computation element
PCC	Path computation client
PHP	Penultimate hop popping
QOS	Quality of service
RSVP	Resource Reservation Protocol
SDN	Software-defined networking
SID	Segment identifier
SLA	Service-level agreement
SPF	Shortest path first
SR	Segment routing
TI-LFA	Topology independent Loop-free alternate
VPN	Virtual private network

1 INTRODUCTION

Segment routing is a new technology developed by Cisco systems and its partners to forward traffic more efficiently than in traditional MPLS networks. It is mainly targeted at service providers, data centres, metropolitan-area networks and large enterprises that use WAN. Segment routing can be implemented on top of OSPF, IS-IS, MPLS and IPv6 configurations with little to no changes. (Nokia Oyj 2016a.)

Segment routing has been well received because it simplifies the network and allows it to scale a lot more efficiently than MPLS networks by reducing the number of protocols required. Segment routing also makes IPv6 more relevant and it is the next step in making networks as scalable as possible. (Nokia Oyj 2016a.)

Scalability of the network is essential because the number of network devices and the amount of traffic are increasing rapidly over time. Cisco has forecasted that the amount of traffic will triple between 2013 and 2018. This means that the network must adapt extremely quickly. (Nokia Oyj 2016a.)

Segment routing is also the answer to the needs of software defined networking. Certain applications require the lowest latency path but a traditional routing protocol may forward the traffic through a higher latency higher bandwidth route. Source routing enables the application to choose the lower latency path if necessary. (Nokia Oyj 2016a.)

I got interested in the topic because it is a new technology that has not been taught to us before. It is also a great way to supplement my skills at service provider networks. I am also interested in how everything in service provider networks works on the smallest level and how it can be improved as the requirements of the network constantly grow.

There have not been any previous studies about this subject in Kymenlaakso University of Applied Sciences. However, Riku Koivula’s (2016) Core Hiding Migration briefly mentions segment routing in the study. Jaakko Nurmi’s (2016) Implementation of Nested Virtual Laboratory System is also related because the segment routing study uses the virtual laboratory created in Jaakko Nurmi’s thesis for case studies.

1.1 SCOPE OF THE STUDY

The scope of this thesis is to study segment routing and demonstrate its uses with case studies in the virtual laboratory using IOS-XRv 6.0.1 routers. The virtual laboratory will contain the configuration examples of segment routing anycast node labels, path selection, path protection and service termination. These configuration examples will be used in future MPLS lessons to demonstrate segment routing to the students. The thesis will also contain the basics of MPLS that will make it easier to understand how segment routing works.

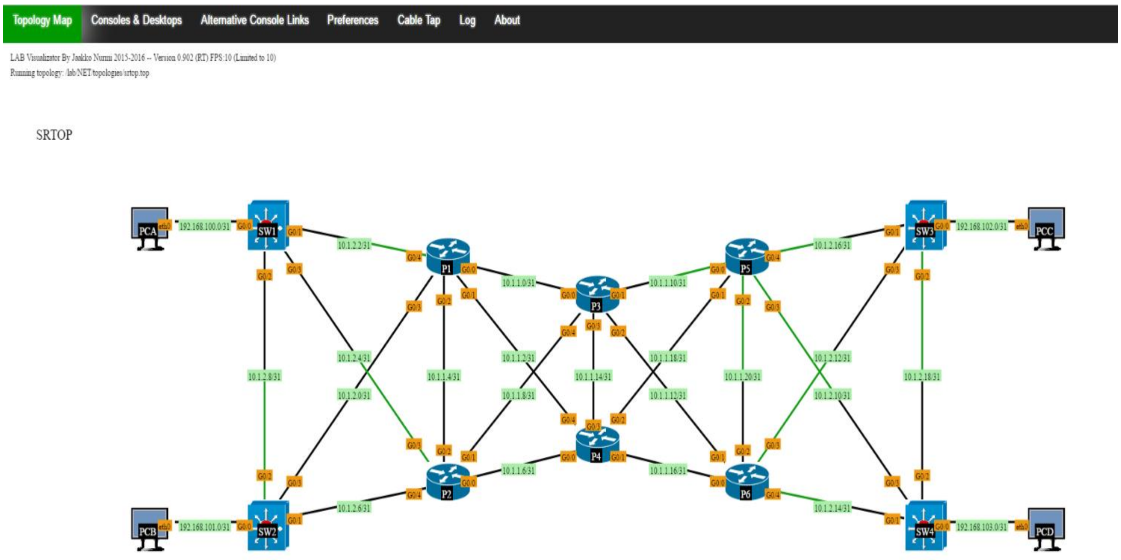
2 VIRTUAL LABORATORY

The case studies in this thesis are implemented in the virtual laboratory. A NestCore is used to run the other virtual machines in the virtual laboratory. In practice, NestCore is a computer that other virtual machines can be run on. The virtual laboratory can be used to implement virtual network devices, use cases and end hosts to simulate a real-life environment. The benefit of virtualization is that the hardware is not as strictly limited as in physical implementations and configuration exercises can be done from anywhere with a VPN. (Nurmi 2016.)

The virtual laboratory can be accessed with any web browser that supports HTML5. The authentication process is handled by a RADIUS server. Users with sufficient permissions can access the laboratory via a VPN. (Nurmi 2016.)

There are four important tabs in the virtual laboratory user interface: topology map, consoles & desktops, preferences and cable tap. The tabs can be navigated from the top of the screen. New topologies can be uploaded from the bottom of the screen. The bottom bar can also be used to navigate between different topologies.

The topology map tab shows the user the current topology used. The map includes details such as the virtual device type, cables, ports and an indication whether the cable has traffic or not. IP-addresses for each link are shown on top of the link. Above the map, there are details about the virtual laboratory version and the current running topology. An example of the topology map can be found in figure 1.



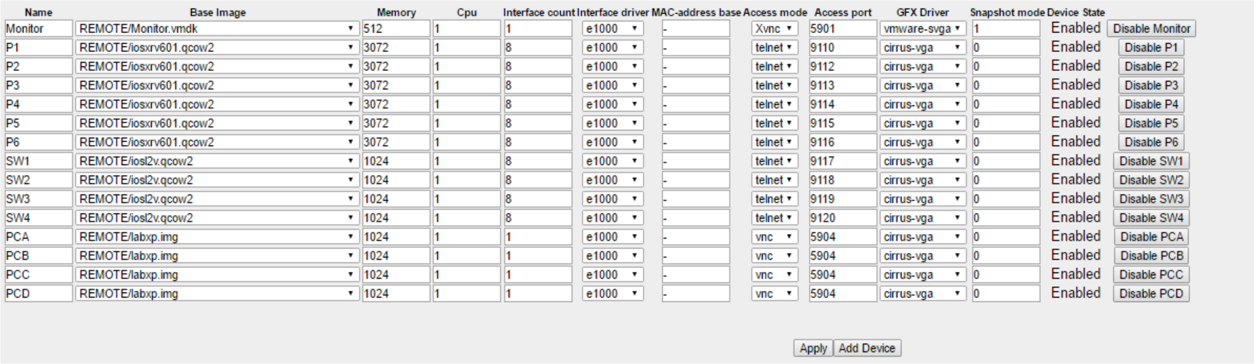
LAB Visualizator By Jaakko Nurmi 2015-2016 -- Version 0.902 (RT) FPS:10 (Limited to 10)
Running topology: /lab/NET/topologies/srtop.top

SRTOP

Figure 1. Virtual laboratory topology map

The consoles & desktops tab displays all the devices included in the topology map without the graphical element. This tab can be used to quickly connect to the desired device. The tab also displays all recently used devices for quick access.

The preferences tab is used to modify the virtual lab parameters. Cables can be modified to simulate delay, packet loss, jitter and bit error ratio. Bandwidth can also be modified from this interface. The different cable modification options offer helpful tools for testing load balancing and traffic engineering. The virtual device manager, which is shown in figure 2, allows the user to name devices, change the image they use and modify several parameters including memory, CPUs, interface count and access port. Devices can be added and removed via this interface.



Name	Base Image	Memory	Cpu	Interface count	Interface driver	MAC-address base	Access mode	Access port	GFX Driver	Snapshot mode	Device State	
Monitor	REMOTE/Monitor.vmdk	512	1	1	e1000	-	Xvnc	5901	vmware-svga	1	Enabled	Disable Monitor
P1	REMOTE/iosxrv601.qcow2	3072	1	8	e1000	-	telnet	9110	cirrus-vga	0	Enabled	Disable P1
P2	REMOTE/iosxrv601.qcow2	3072	1	8	e1000	-	telnet	9112	cirrus-vga	0	Enabled	Disable P2
P3	REMOTE/iosxrv601.qcow2	3072	1	8	e1000	-	telnet	9113	cirrus-vga	0	Enabled	Disable P3
P4	REMOTE/iosxrv601.qcow2	3072	1	8	e1000	-	telnet	9114	cirrus-vga	0	Enabled	Disable P4
P5	REMOTE/iosxrv601.qcow2	3072	1	8	e1000	-	telnet	9115	cirrus-vga	0	Enabled	Disable P5
P6	REMOTE/iosxrv601.qcow2	3072	1	8	e1000	-	telnet	9116	cirrus-vga	0	Enabled	Disable P6
SW1	REMOTE/iosl2v.qcow2	1024	1	8	e1000	-	telnet	9117	cirrus-vga	0	Enabled	Disable SW1
SW2	REMOTE/iosl2v.qcow2	1024	1	8	e1000	-	telnet	9118	cirrus-vga	0	Enabled	Disable SW2
SW3	REMOTE/iosl2v.qcow2	1024	1	8	e1000	-	telnet	9119	cirrus-vga	0	Enabled	Disable SW3
SW4	REMOTE/iosl2v.qcow2	1024	1	8	e1000	-	telnet	9120	cirrus-vga	0	Enabled	Disable SW4
PCA	REMOTE/labxp.img	1024	1	1	e1000	-	vnc	5904	cirrus-vga	0	Enabled	Disable PCA
PCB	REMOTE/labxp.img	1024	1	1	e1000	-	vnc	5904	cirrus-vga	0	Enabled	Disable PCB
PCC	REMOTE/labxp.img	1024	1	1	e1000	-	vnc	5904	cirrus-vga	0	Enabled	Disable PCC
PCD	REMOTE/labxp.img	1024	1	1	e1000	-	vnc	5904	cirrus-vga	0	Enabled	Disable PCD

Figure 2. Virtual device parameter configuration window

The cable tap tool can be used to analyse the traffic of any interface that exists in the topology. All interfaces are listed on the left of the screen. Wireshark running on a tinycore Linux is used for analysing the traffic. Traffic capturing is a useful tool when troubleshooting the network.

3 MPLS

Multiprotocol label switching (MPLS) is a technique that uses labels to forward packets in the network. The label switching technology was originally proposed by Cisco, but was developed and standardized by IETF. The benefit of the labels is that the routers in the MPLS network do not have to make any lookups on the routing table. The packet header will only be analysed once when it enters the MPLS network at the ingress router. (Mplsinfo 2009.)

MPLS supports quality of service (QOS) to meet service level agreements (SLAs). For example, SLAs can be used to give high priority traffic more bandwidth or lower latency compared with the rest of the traffic. This can be used to guarantee the functionality of services such as voice over internet protocol (VOIP) or video traffic. (Rouse 2014.)

MPLS can be used to create virtual leased lines (VLLs) and separate different services from each other. It also supports the creation of virtual private networks (VPNs). Virtual private LAN service (VPLS) makes it possible connect multiple remote sites through the public network. VPLS makes it look like all the connected sites are on the same LAN. (Rouse 2014.)

Downside of MPLS is that it is complicated to manage because it relies on other routing protocols for transferring control plane data. Troubleshooting can also be difficult because MPLS labels are locally significant, which makes it harder to track any problems. The traffic can be easily disrupted by failures because the switch to a failover link is relatively slow. (Mplsinfo 2009.)

3.1 OPERATIONS

MPLS operates in both layer 2 and layer 3. This ensures that it works for any protocol and removes protocol dependency from datalink layer technologies. (Mplsinfo 2009.)

MPLS traffic is assigned a forward equivalence class (FEC) when it first enters the network. FEC determines the service requirement of the data traffic. FEC is always the same for the same type of traffic. (Mplsinfo 2009.)

An MPLS router can perform three different operations to the packet when it arrives. The three operations are called SWAP, PUSH and POP. In a SWAP operation, the MPLS router swaps a new label for the existing label and then forwards the packet per the new label. In PUSH operation, a new label is added to the packet. PUSH can only be done if the packet has no MPLS labels. POP operation removes the top label of the stack, which is usually done before the label reaches the egress router. (Juniper Networks 2012.)

Penultimate Hop Popping (PHP) is an operation that is used to remove the label from the packet before the packet is passed to the Label Edge Router. PHP reduces the number of label lookups the LER needs to do, which reduces the processing power required on the LER. The usage of PHP moves some of the load from LER to the LSRs of the network. Figure 3 demonstrates the PHP operation: P3 removes the label from the packet before passing it to P1. (Cisco Systems 2005.)

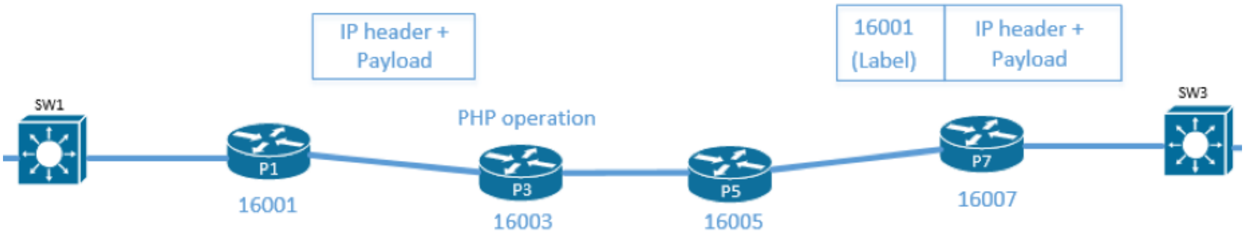


Figure 3. Penultimate Hop Popping in the network

An ingress label edge router (LER) is the entry point to the MPLS network. This is where the packet will be prepared for the MPLS network and a PUSH operation is performed. An egress label edge router (LER) is the router where the data traffic exits the MPLS network and a POP operation is performed. (Mplsinfo 2009.)

A label switch router (LSR) is a router in the MPLS network that forwards the packets per their labels by performing the SWAP operation. Depending on the position of the router, it may also have to POP or PUSH labels. (Mplsinfo 2009.)

A label switch path (LSP) is the path that an MPLS packet takes in the MPLS network. LSPs can be used for VPNs or routing via a specified path in the network. (Mplsinfo 2009.)

3.2 MPLS LDP

MPLS Label Distribution Protocol (LDP) is a protocol that enables label switch routers to request, distribute and release label binding information with other neighbouring routers. A label switch router advertises its label bindings to its neighbours when LDP is enabled. LDP is also known as hop-by-hop forwarding and it is used in MPLS VPN. MPLS network can be difficult to troubleshoot in large networks because the labels are only locally significant for each router. (Cisco Systems 2005.)

When two LSRs discover each other via hello messages, they start negotiating the LDP parameters. The router with a higher transport IP address will establish the session and function as an active router. The other router will be passive. Once two routers have negotiated the parameters, they form a Label-Switched Path (LSP). LSPs are formed based on the paths calculated by the Interior Gateway Protocols (IGPs). LDP uses the information it receives from the IGP and assigns labels to each route. (Cisco Systems 2011.)

3.3 MPLS RSVP TE

Resource reservation protocol (RSVP) is a signalling protocol that reserves a certain path for certain type of traffic. RSVP-TE can be used to implement QoS and load-balancing into the MPLS network. It calculates the routes with CSPF and Explicit Route Objects (EROs) unlike LDP, that only uses the shortest path. (Juniper Networks 2016.)

The head-end router, which is the source of the traffic, signals a PATH message toward the tail-end router. The PATH message specifies the sender of the message and the desired resources for the path. The message is also examined by all the routers on the way to the tail-end. This updates the session-state, reserves the desired resources and allocates an MPLS label on all the routers for the path. The path is then maintained by periodical PATH and RESV messages from all the routers along the path. (Cisco Systems 2015.)

The requirement of a state on every router of the path limits the scalability of RSVP-TE. Session state must be signalled every 30 seconds and the session will expire if no state message is received in 3 minutes. After 3 minutes, the RSVP session will be moved to another LSP on another router. (Juniper Networks 2016.)

4 SEGMENT ROUTING

Segment routing is a new packet forwarding method created by Cisco. Segment routing utilizes source routing, which means that the path to the destination is determined before the packet leaves the ingress router. This is achieved by assigning a segment identifier (SID) or a segment list to the packet. A segment list is a set of instructions that each router uses to forward the packet through the network. Once the first instruction has been executed, it will be removed from the list and the packet continues forward with the remaining segment identifiers. This will continue until the packet reaches its destination. (Nokia Oyj 2016a.)

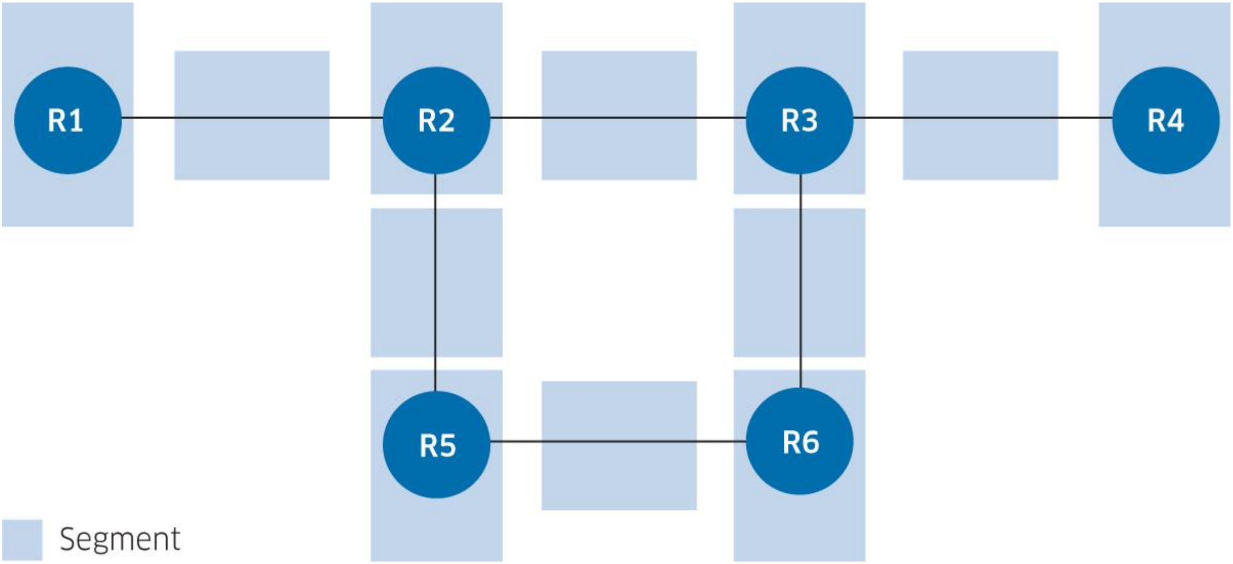


Figure 4. Each part of the network forms its own segment (Nokia Oyj 2016)

Figure 4 shows which parts of the network receive their own segment identifier. Segment identifiers are encoded as either MPLS labels or IPv6 addresses. Each node and link get their own segment identifier in a segment routing network. These segment identifiers are globally significant in the segment routing domain, which makes troubleshooting the network easier than in MPLS LDP networks where labels have only local significance. (Nokia Oyj 2016a.)

Segment routing does not require LDP or RSVP-TE to work. The label distribution is handled by IGP. RSVP-TEs downside is that it must maintain a state on each router along the path, which makes it scale poorly. The only state that segment routing traffic engineering (SR-TE) must maintain is at the ingress router. This allows the network to scale significantly better than with the older technologies. (Nokia Oyj 2016a.)

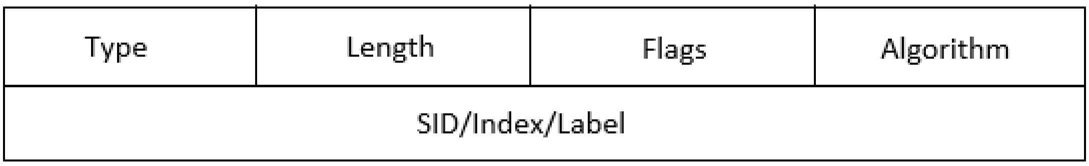
The introduction of segment routing has also renewed the interest in stateful active path computation element (PCE). PCE is a traffic engineering controller that manages the network by allocating the correct paths and resources to the services that need them. The use of PCE also makes software defined networking (SDN) more appealing in WAN networks. The simplification of the network allows features like bandwidth calendaring and bandwidth on demand which are needed by the SDN applications. (Nokia Oyj 2016a.)

4.1 SEGMENT IDENTIFIERS

Segment identifiers (SIDs) are used in the SR network to identify different parts of the network. Segment routing reduces the number of labels required in the network because a label can indicate a whole path instead of just one hop to the next node like in RSVP-TE. Commonly used segment identifiers include Prefix, Adjacency and Anycast-SID. (Cisco Systems 2016.)

4.1.1 PREFIX-SID

A prefix-SID is the SID of an IGP-prefix segment. A prefix-SID is a unique identifier within the SR/IGP domain. An IGP-prefix segment consists of the following fields: type, length, flags, algorithm and SID/Index/Label as shown in figure 5. (Previdi 2015.)



Type	Length	Flags	Algorithm
SID/Index/Label			

Figure 5. Prefix-SID structure. (Nokia Oyj 2016)

The type field signifies the type of the packet. In this case, the type value is set to 3. Length is a variable that can be different for every label. (Previdi 2015.)

The flag field specifies how the prefix-SID should be handled. There are six different flags in total. Each flag can have a value of either 0 or 1. The behaviour of the packet changes depending on the flag values. (Previdi 2015.)

R-flag means that the prefix-SID is redistributed or from another level, for example a level-1 prefix-SID propagated to level-2. Redistribution means the flag has been redistributed by another protocol. N-flag specifies that the prefix-SID is a Node-SID which means that it refers to a router’s loopback address. N-flag can be unset to prevent it from referring to a router. This is needed when configuring an anycast-SID. (Previdi 2015.)

P-flag prevents the PHP-operation from being applied to the prefix-SID when set. P-flag can be disabled to enable the PHP-operation. PHP reduces the load on edge routers. E-flag specifies that the upstream routers must have an explicit-null value for that prefix-SID. A label with explicit-null value will be popped when received by an LSR. (Previdi 2015.)

V-flag is unset by default, which means that the packet carries an index. If V-flag is set, then the packet carries a local label value instead. An index value is used to determine the SID or label value. L-flag is also unset by default. If L-flag is set, the prefix-SID is only locally significant. Locally significant labels are not distributed to other routers. (Previdi 2015.)

The algorithm field defines how the packet must be forwarded to its destination. A value 0 in the algorithm field signifies that the packet must take the Shortest Path First (SPF) computed by IS-IS. However, this SPF calculation can be overwritten by any router on the path that has a local policy. A value of 1 means that the Shortest Path First calculation is strict and cannot be overwritten by local policies. (Previdi 2015.)

The SID/Index/Label field contains an index that defines the offset in the label space. This can be used to calculate the SID of the segment. (Previdi 2015.)

Segment routing Node-SID or IGP Node Segment identifies a specific router in the network by using the nodes loopback address as the prefix. Node-SID can be used to navigate from any point of the segment routing network to a segment routing node (LSR) with a corresponding Node-SID. (Previdi 2015.)

4.1.2 ADJACENCY-SID

An adjacency segment identifier (Adj-SID) is used by a segment routing node to advertise its links to adjacent routers. Adj-SIDs are not unique within the SR domain by default. Adj-SIDs are allocated from the dynamic label range which starts from 24000 in IOS-XRv. Multiple links can have the same Adj-SID. (Singh 2015.)

A global Adj-SID is advertised beyond the adjacent routers to the whole SR domain. This can be used to reduce the size of the segment list required. However, the usage of global Adj-SID introduces an additional state to the SR network. (Filsfils 2015.)

Adjacency-SID consists of Type, Length, Flags, Weight and SID/Index/Label fields. Adj-SIDs suggested type value is 31. Length is variable on label-to-label basis. (Previdi 2015.)

The flag field has the following flag settings: F, B, V, L and S. F-flag is the address-family flag. If the F-flag is set, the adjacency is IPv6. If left unset, the adjacency is IPv4. B-flag is the Backup-flag. If backup-flag is set, the adjacency-SID can be protected by TI-LFA. More information about TI-LFA can be found in Chapter 4.6. (Previdi 2015.)

V-flag signifies that the Adj-SID has a value. It is set by default. L-flag is the Local-flag. It is set by default and means that the Adj-SID is locally significant. S-flag indicates that the Adj-SID is used for multiple adjacencies when set. (Previdi 2015.)

The weight field determines the Adj-SIDs weight in load-balancing scenarios where there are multiple paths with the same Adj-SID. For example, link A and link B both have Adj-SID of 24005. Link A has a weight of 1 and link B has a weight of 2. The traffic will be load-balanced between link A and B in 1:2 ratio. (Filsfils 2015.)

The SID/Index/Label field contains either a local label, an index defining the offset in Label space or an IPv6 address. The local label requires that V and L flags are set. However, if they are unset, the field will be an index field. An IPv6 address requires that V-flag is set. L-flag can be either set or unset depending whether the IPv6 address is globally or locally significant. (Previdi 2015.)

4.1.3 ANYCAST SID

Anycast segment-identifier (Anycast-SID) is a traffic-engineering tool for steering traffic through certain paths. For example, anycast-SIDs can be used to differentiate two different planes or regions. This can be useful if there are two or more paths that have different attributes. (Cisco Systems 2015.)

Plane A can have high bandwidth and high latency while Plane B has low bandwidth and low latency. Services with a low latency requirement are steered to Plane B using the corresponding anycast-SID, while the rest of the traffic with no latency requirements can use the high bandwidth path. Figure 6 shows a typical anycast-SID topology. (Cisco Systems 2015.)

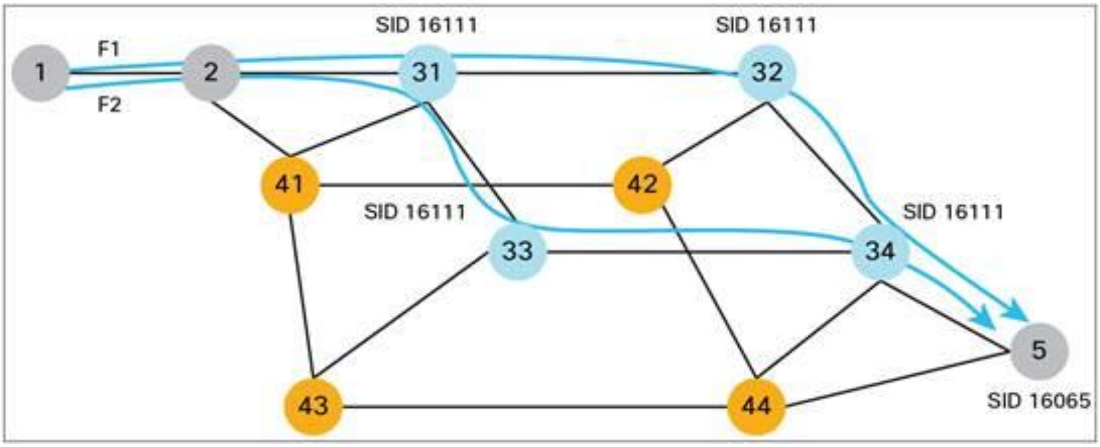


Figure 6. Dual-plane network using anycast-SID (Cisco Systems 2015)

Anycast-SID paths are also ECMP aware. This means that once the packet reaches Plane A, the traffic can be split between multiple equal cost paths inside of that plane. This provides a good way to load-balance the traffic. (Cisco Systems 2015.)

In addition, anycast-SIDs provide resiliency for the path. When a packet with anycast-SID 16001 enters the corresponding plane, it is not tied to any specific router. In the case of a node failure, the other routers with anycast-SID

16001 will fulfil the traffic-engineering policy and the packet will not be dropped. (Cisco Systems 2015.)

4.1.4 BINDING-SID

Binding-SID is a label that can be used to nest and stitch domains together. This feature is useful if there are non-SR domains on the path of the traffic. For example, if two SR-domains are separated by an RSVP-TE domain, the binding-SID can be used to reach the start of the RSVP-TE tunnel. Using a binding-SID between two domains on a gateway allows the usage of a much shorter label stack. (Singh 2016.)

However, in IOS-XR 6.2.0 binding-SID is used as a policy identifier for different SR-TE policies. SR-TE binding-SID steers the traffic to an SR-TE tunnel. It is recommended that an SR policy should have a stable binding-SID. (Segment-routing 2017.)

4.2 SR OPERATIONS

Segment routing operations are very similar to MPLS. The underlying operations stay the same as before. A packet entering the segment routing network needs to be modified to be able to traverse through the network. These modifications keep happening until the packet reaches the destination in the segment routing network or the egress router. (Nokia Oyj 2016b.)

When a packet reaches the ingress router of a segment routing network, it receives a segment or a list of segments. This is called the PUSH operation. The topmost label in the list of segments determines where the packet goes next. (Nokia Oyj 2016b.)

When the packet reaches the next label switch router, the router will apply a CONTINUE operation, which is the same as SWAP operation in MPLS. In the CONTINUE operation, the outgoing label value is equal to the incoming label value. (Nokia Oyj 2016b.)

Before the final hop to the destination, the router applies a NEXT operation to the packet, which removes the labels from the packet just like POP operation does in MPLS. The NEXT operation can also work in the same way as Penultimate Hop Popping in MPLS. (Nokia Oyj 2016b.)

4.3 MAPPING SERVER

Segment routing-only-nodes are not able to work with LDP-only-nodes by default. A mapping server must be configured on a segment routing node to enable the interworking between segment routing and LDP nodes. A network can have multiple mapping servers active at one time. Figure 7 shows a typical use case for a mapping server. (Segment-routing 2016.)

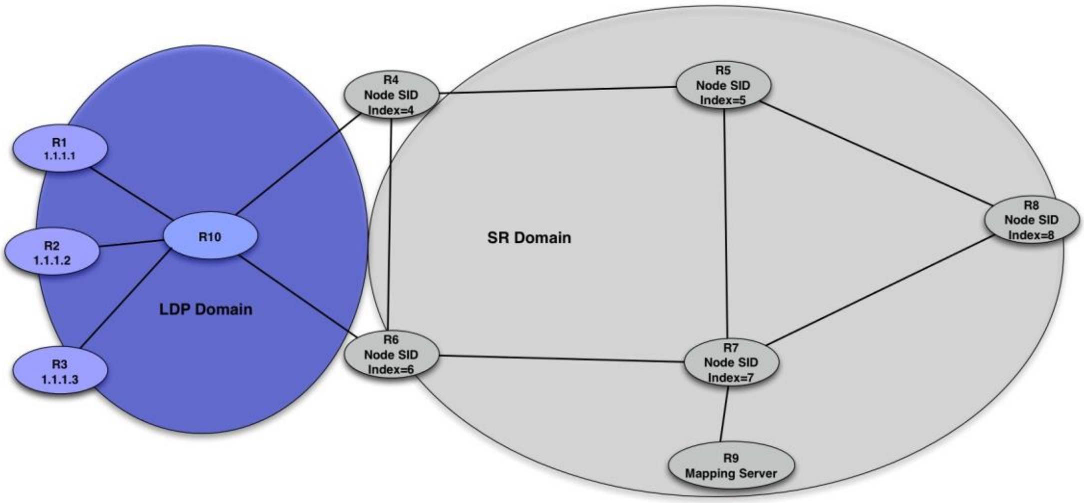


Figure 7. A mapping server in the network (Packetpushers 2016)

A mapping server assigns a SID to every LDP prefix and advertises it to the network. This ensures that the segment routing only nodes can reach the LDP nodes with a SID. Every node except the mapping server must be configured as a mapping server client to enable the prefix-to-SID mappings. (Segment-routing 2016.)

LDP-to-SR interworking happens automatically. When a packet from the LDP network reaches the SR border, the border router looks up the Prefix Segment bound to the destination IP-address. Once the correct forwarding entry has been found, the border router installs the Prefix Segment and the packet will be forwarded to the correct outgoing interface. (Segment-routing 2016.)

Mapping server functionality can be configured to either OSPF or IS-IS instance of the router. If multiple mapping servers are configured, the overlapping entries will be put to the backup policy. Active policy must be identical on every participating router. Every IS-IS area requires its own mapping server, because the prefixes do not get advertised between levels. However, multiarea OSPF can be covered with one mapping server because the advertisements go between different areas. (Segment-routing 2016.)

4.4 PATH SELECTION

Segment routing has multiple ways to determine the path (SR tunnel) that the packet will take. These include the Constrained Shortest Path First (CSPF) which is a traffic-engineering extension to IGP, standard shortest path routing and centralized TE. Path selection becomes important when the traffic has other requirements than the shortest path to the destination. (Nokia Oyj 2016b.)

CSPF is similar with Open Shortest Path First with one important addition. It filters out any paths that do not fulfil the requirements of the traffic such as bandwidth, latency or number of hops to the destination. CSPF gets the information it requires from TE extensions made to IGP. Segment routing network does not have a control plane for LSPs which means that CSPF cannot use bandwidth as a constraint to filter out paths. (Nokia Oyj 2016b.)

Standard shortest path routing means that the packet will traverse the IGP shortest path from ingress to the destination. This is the easiest way to set up a segment routing network. IS-IS and OSPF have an extension to advertise Prefix-SIDs to the network. Equal Cost Multiple Path (ECMP) will also be used if there are multiple paths with equal cost and the setting is enabled. ECMP balances the traffic evenly between multiple paths. (Nokia Oyj 2016b.)

4.4.1 LOAD-BALANCING

Segment routing uses ECMP to load-balance the traffic between two equal cost routes. ECMP is enabled by default in IOS-XRv devices. The ECMP-enabled ingress router can use IP, TCP/UDP header information and the router ID as a hash key to determine which traffic flow is mapped to which ECMP. The ECMP calculation is used by the ingress network processing unit (NPU) of the router to determine, which way the traffic is forwarded. (Cisco Systems 2017a.)

The problem with this method is that LSRs can also have ECMP routes and they require the same hash keys as the ingress router. Importing the hash keys to another non-ingress LSR requires the usage of Deep Packet Inspection (DPI), which requires a lot of router’s processing capacity and is not necessarily supported by all routers. (Juniper Networks 2015.)

The Entropy Label (ER) is a solution to the processing requirements and lack of support that the DPI suffers from. An Entropy Label is an additional label that is added to the label stack. It is used as a hash key by the LSRs to determine which ECMP the packet will take to its destination. An entropy label gets a standard label value from the 20-bit range, which makes it overlap with the existing label value range. To combat this, an entropy label indicator (ELI) is added in front of the label to distinguish it from normal labels. ELI has a value of 7, which is a special value outside of the 16 to 1048575 standard range. (Juniper Networks 2015.)

The usage of Entropy Labels in a network requires that every participating router supports the EL technique. It is also important to consider the maximum depth of the label stack the routers can process. If the routers do not support a certain stack depth, the labels cannot be processed. This can be prevented by informing the ingress router of the maximum stack depth that can be imposed on a packet without causing issues. Each router sends the ingress router a Readable Label Depth (RLD) message which allows the ingress router to keep the label stacks from going over the maximum value. In segment routing the RLD will be advertised in ELC TLV or ELC Sub-TLV for IS-IS. (Nokia Oyj 2016b.)

4.4.2 DISJOINTNESS

Disjointness is a technique to separate two traffic flows and make them completely independent of each other. Disjointness ensures that even if one traffic flow gets interrupted by a failure, the services can be maintained using the working path instead. These duplicated paths can be used by applications that require low latency and zero packet loss such as robot-assisted remote surgery applications. (Aubury 2015.)

A certain level of disjointness can be achieved by using the anycast-SIDs in a segment routing network. Anycast-SIDs cause the traffic to distribute evenly between the links that share the anycast-SID. Anycast-SIDs do not guarantee a 100% disjointness because IGP shortest path calculation does not take disjoint paths into account. (Nokia Oyj 2016b.)

A guaranteed disjointness can be achieved via a PCEP protocol extension called path-profiles. PCEP peers can use path-profiles to affect the routing decisions by imposing a list of parameters and policies in the path-profile. Each path profile is assigned a profile identifier that can be used to reference to a certain path-profile. Path-profile objects are used for sending path-profiles to other routers. (Nokia Oyj 2016b.)

The path computation using path-profiles consists of five steps. First the Operational Support System (OSS) messages the PE router which also works as the PCC with service provisioning details like tunnel type and path constraints. (Nokia Oyj 2016b.)

Next the PCC will send a Path Computation Request to the PCE server if a tunnel with the specific constraints does not exist. The PCReq message will also contain the path-profile details. (Nokia Oyj 2016b.)

Since the path-profile contains the constraint of disjoint paths, the PCE server can take it into account when calculating paths for services. PCE will then signal an adequate path-profile to all PCC routers that are part of the disjoint paths. (Nokia Oyj 2016b.)

Finally, the PCC will delegate the control of its LSPs to the PCE. PCC will keep monitoring all LSPs and inform the PCE. This ensures that the PCE can keep track of LSPs and update or optimize when necessary. (Nokia Oyj 2016b.)

4.4.3 TRAFFIC ENGINEERING

Segment routing traffic engineering is a tool that can be used to customize the path that the traffic flow takes. For example, SR-TE makes it possible to configure disjointness and dynamic paths. Dynamic paths can be configured to avoid links and nodes. SR-TE utilizes ECMP routers whenever they are available. (Singh 2016.)

SR-TE does not require signalling between the head-end and tail-end router. The only state required is in the head-end router, which makes SR-TE more scalable than RSVP-TE. SR-TE does not require a state because the path is carried in the packet as labels. This means that the routers after the head-end are not aware of the SR-TE tunnel. (Singh 2016.)

SR-TE tunnels can be configured manually or a PCE server can configure them. SR-TE configuration is very similar with RSVP-TE configuration. Only difference is that instead of enabling RSVP-TE the user must enable segment routing. The configuration consists of enabling the traffic-engineering in the IS-IS instance, configuring the SR-TE tunnel and making an explicit path for the tunnel to use. (Singh 2016.)

4.5 PCEP

Path computation element protocol (PCEP) is a TCP based protocol that enables the use of path computation element (PCE). PCE is a network device, application or component that calculates optimal paths for path computation clients (PCC) based on the network data it receives from PCC. (Juniper Networks 2015.)

PCE has been used with RSVP-TE to some extent in the past. However, it never became popular due to scalability issues. Segment routing does not require midpoint states in the network, which solves the scalability issues of RSVP-TE. The lack of midpoint states also means that the SR network ingress router cannot calculate paths with bandwidth constraints. PCE solves the path calculation problem that segment routing has. (Nokia Oyj 2016b.)

The PCE server's ability to control LSPs in the network is an important part of Software Defined Networking (SDN). PCE enables the applications to affect the paths and constraints their traffic gets. This will be increasingly important in the future when more devices have internet access and everything needs to be optimized. (Linuxfoundation 2015.)

There are two types of PCE: stateful and stateless. A stateful PCE server constantly adjusts and maintains the paths and resources in the network. Because of this, it is important that the PCE is synchronized well with the network and Traffic Engineering Database (TED). A stateless PCE server cannot actively adjust the paths because it cannot see which Label-Switched Paths are active at any given time. A stateful PCE server gives the network a better performance than a stateless server if it stays well synchronized consistently. (Nokia Oyj 2016b.)

4.5.1 PASSIVE

A passive stateful PCE server gets its LSP information from the PCC using the PCEP. This information is used to compute paths for the PCC. When PCC needs to compute a path, it sends a Path Computation Request (PCReq) to the PCE. The PCReq message contains the requirements and constraints for the path. The PCE server then replies with Path Computation Reply (PCRep) that is either positive or negative. A positive PCRep causes the PCC to establish a path per the instructions that were received in the reply. If the reply is negative, the PCE will try to ease the requirements and send a new PCReq. (Nokia Oyj 2016b.)

4.5.2 ACTIVE

An active stateful PCE server does everything that a passive stateful PCE does. However, the PCC can delegate the control of its LSPs to the active stateful PCE server. This is done by setting the U-flag to 1 during the negotiation between the PCE and PCC. The active status allows the PCE to modify the LSP parameters of PCC such as bandwidth or Explicit Route Objects (ERO). To modify these settings, the PCE sends a Path Computation Update (PCUpd) message to the PCC that specifies the changes that must be made to the LSP. PCC will then reply with a Path Computation Report (PCRpt) which can indicate that the changes were either successful or unsuccessful. (Nokia Oyj 2016b.)

4.5.3 SOFTWARE-DEFINED NETWORKING

Software-Defined Networking (SDN) is an architecture that makes the network more flexible for applications that require dynamic changes to the paths they use. Changes to bandwidth or latency constraints normally require the reconfiguration of the network. SDN is mainly an architecture used internally in data centres and enterprises. (Opennetworking 2016.)

OpenDaylight (ODL) is an open source SDN controller that can be used to implement a wide variety of functions to the network. ODL is a modular software that allows the user to choose and install desired functions manually. (Opendaylight 2016.)

Software-Defined WAN (SD-WAN) is a technology similar to SDN but for Wide Area Networks. SD-WAN uses a centralized controller for adjusting the policies and bandwidth of the WAN. SD-WAN controller can be a physical or virtual device. Currently less than 1% of the enterprises use SD-WAN technology, however the percentage of SD-WANs is predicted to go up to 30% by 2019. (Lerner 2015.)

SD-WAN simplifies the management of the network by centralizing the control over the network to a single location. All non-physical changes can be implemented from a graphical user interface without the requirement of being on-site (Networkworld 2016). This greatly increases the cost-effectiveness of the WAN. SD-WAN is estimated to be up to 2.5 times cheaper to maintain than a traditional WAN. (Lerner 2015.)

The PCEP protocol can be utilized in the deployment of SDN and SD-WAN. A PCE works as a centralized controller that can modify the network automatically. Segment Routing Traffic Engineering (SR-TE) limits the modification of the network to the ingress LSR. This allows the bandwidth and policies to be more dynamically allocated between the tunnels because the tunnels are controlled by a single router. (Nokia Oyj 2016b.)

4.6 TI-LFA

Topology independent loop-free alternate (TI-LFA) is a feature that protects links, nodes and SRLGs. It is simple to configure; only two lines of configuration are required to implement a simple TI-LFA configuration into the router. It does not require any changes to the existing protocols used in the router. (Nokia Oyj 2016a.)

TI-LFA improves fast-rerouting by not needing targeted LDP sessions. Targeted LDP sessions must be dynamically created and torn down, cleaned up and re-established once the topology changes. Figure 8 shows a comparison between TI-LFA, RSVP-TE and LFA. TI-LFA covers 100% of the network and the backup paths are more optimal than in LFA or RSVP-TE. (Sánchez-Monge 2015.)

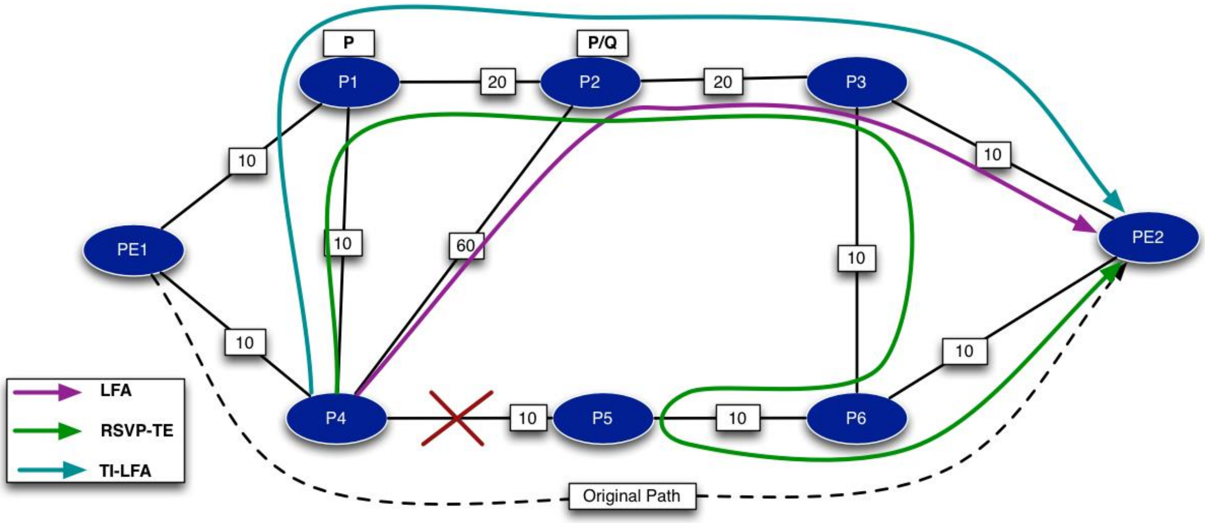


Figure 8. A comparison between TI-LFA and its precursors (Singh 2015)

Every protected node and path has a pre-calculated backup path that can be enabled quickly. The convergence time for a protected path is 50 milliseconds or less. This means that even the most latency or packet loss sensitive applications can work with no disruptions in case a node or a link fails. (Nokia Oyj 2016a.)

TI-LFA calculates the backup path by temporarily removing the protected link or node from the database. After this, it calculates the backup path with shortest path first. This ensures that the backup path has the lowest possible metric cost while avoiding the protected path. (Sánchez-Monge 2015.)

A traffic-engineered tunnel that follows the backup path will be used for traffic if a failure occurs. A repair label list determines the path for the packets that need a new route to their destination. A repair label list is a normal label stack but it is only used when a failure occurs in the protected route. (Sánchez-Monge 2015.)

4.7 SRv6

IPv6 segment routing utilizes the IPv6 technology for implementing SR. SRv6 works in native IPv6 environment, which means that no MPLS is required. The flexibility of the IPv6 header enables it to support segment routing. (Trate 2016.)

An IPv6 packet has a group of headers called Extension Headers. One of these headers is called the Routing Header. The routing header specifies intermediate hops that the packet must visit before going to its destination. (Trate 2016.)

The Routing Header has been modified for the purposes of Segment Routing and is called the Segment Routing Header (SRH). The SRH defines the packets path with a list of segments just like in normal segment routing. However, in this case the individual segments are IPv6 addresses instead of the usual SIDs. This eliminates the need for signalling SIDs in the network. (Trate 2016.)

SRv6 supports TI-LFA, disjointness and any other SRv4 feature. SRv6 is still in development but it will be a vital part of the future service provider networks. (Trate 2016.)

5 CASE STUDY IMPLEMENTATION

The case studies were implemented in the virtual laboratory environment. The core of the network consists of eight IOS-XRv routers. Four switches and one PC were also needed in the implementation.

One of the goals of the case study was to implement a Path Control Element server (PCE). This PCE was implemented on a Linux server that was attached to one of the ingress routers of the topology. The PCE uses PCEP protocol to communicate with the PCC.

The case study also contains various examples of path protection, service termination, load balancing and anycast-SID. Some technologies were not available on the IOS-XRv 6.0.1. These technologies are mentioned at the end of the study.

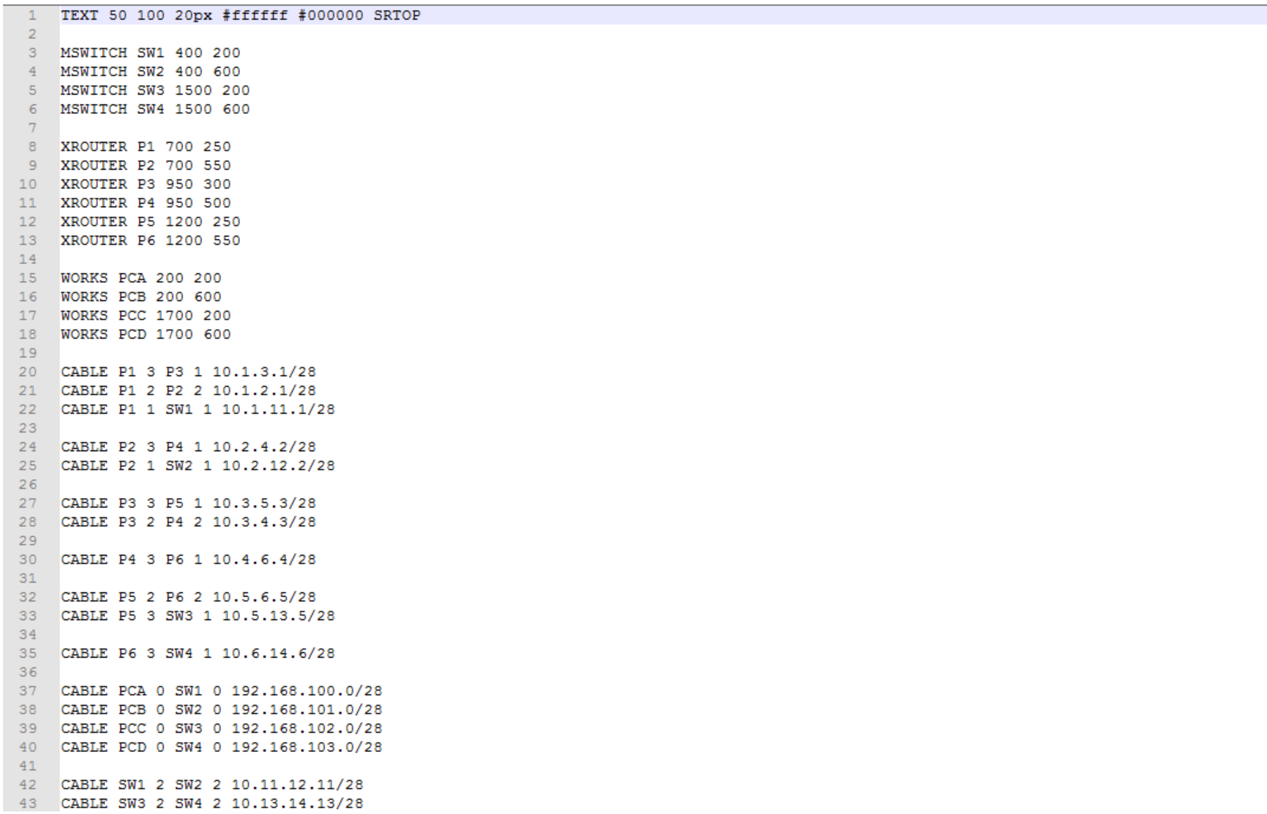
Path protection consists of TI-LFA link protection configuration and testing of the convergence speed. As of December 2016, the TI-LFA node protection feature does not work on the virtual routers but configuration examples will be included in the end of the TI-LFA part.

Anycast-SIDs can be used to demonstrate a form of load-balancing, however the main objective of anycast-SIDs is to demonstrate the dual plane design that can be achieved. Dual plane design allows disjoint services to be implemented. Load-balancing explores the usage of Equal Cost Multiple Path (ECMP) and Entropy Labels (ER).

Segment routing with IPv6 (SRv6) will be part of the study if it is supported by the current version of the IOS-XRv. If the current version is incompatible, configuration examples will be shown anyway.

5.1 TOPOLOGY

The implementation of the case studies began with the configuration of the virtual laboratory topology. The topology can be configured by modifying a text file that was then uploaded to the virtual laboratory from the virtual laboratory user interface. The text file can be used to determine the type of device, its coordinates, cables and the ports that it is attached to.



```
TEXT 50 100 20px #ffffff #000000 SRTOP

MSWITCH SW1 400 200
MSWITCH SW2 400 600
MSWITCH SW3 1500 200
MSWITCH SW4 1500 600

XROUTER P1 700 250
XROUTER P2 700 550
XROUTER P3 950 300
XROUTER P4 950 500
XROUTER P5 1200 250
XROUTER P6 1200 550

WORKS PCA 200 200
WORKS PCB 200 600
WORKS PCC 1700 200
WORKS PCD 1700 600

CABLE P1 3 P3 1 10.1.3.1/28
CABLE P1 2 P2 2 10.1.2.1/28
CABLE P1 1 SW1 1 10.1.11.1/28

CABLE P2 3 P4 1 10.2.4.2/28
CABLE P2 1 SW2 1 10.2.12.2/28

CABLE P3 3 P5 1 10.3.5.3/28
CABLE P3 2 P4 2 10.3.4.3/28

CABLE P4 3 P6 1 10.4.6.4/28

CABLE P5 2 P6 2 10.5.6.5/28
CABLE P5 3 SW3 1 10.5.13.5/28

CABLE P6 3 SW4 1 10.6.14.6/28

CABLE PCA 0 SW1 0 192.168.100.0/28
CABLE PCB 0 SW2 0 192.168.101.0/28
CABLE PCC 0 SW3 0 192.168.102.0/28
CABLE PCD 0 SW4 0 192.168.103.0/28

CABLE SW1 2 SW2 2 10.11.12.11/28
CABLE SW3 2 SW4 2 10.13.14.13/28
```

Figure 9. An example topology created for the virtual lab

Figure 9 shows how the virtual laboratory topology is configured. MSWITCH is a layer 3 switch, XROUTER is an IOS-XR router and WORKS is a workstation. These lines also determine the name of the device and their coordinates.

Cables are configured by determining the first device and one of its ports and then second device and a port. After this one can optionally type an IP-address range that will be shown on top of the link in the virtual laboratory topology map. A new topology file must be uploaded each time the user wishes to alter the topology.

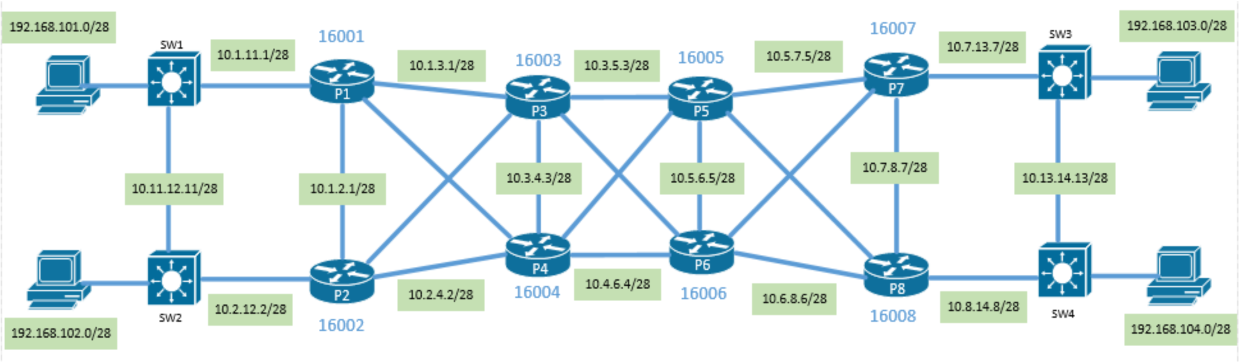
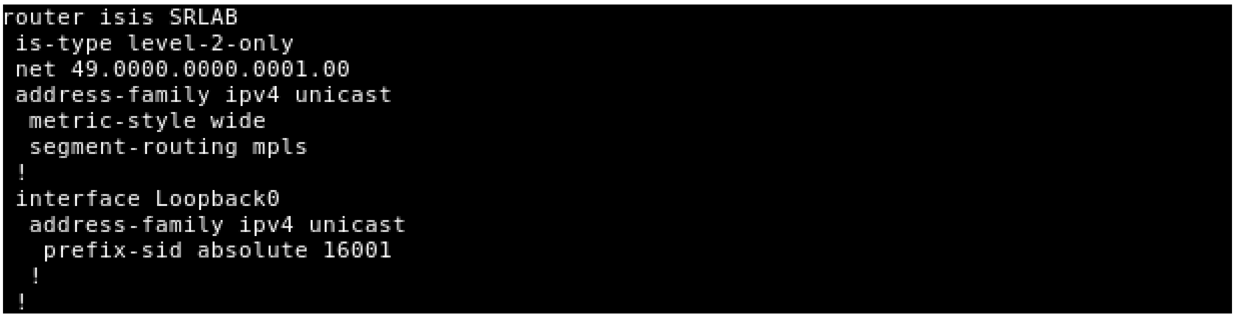


Figure 10. The virtual laboratory topology

The topology can be seen in figure 10. It is designed in a way that offers a good environment to test multiple paths and see how quickly the network recovers from a link or node failure. IOS-XRv cannot see a link failure directly, which can affect the recovery time. PCs exist in the topology to test the PCEP protocol. In this case, the PC will act as a PCE. The topology may be slightly altered between the case studies if it is required to demonstrate certain techniques.

All IP addresses in the segment routing topology are allocated in the same way. The address format is 10.x.y.z, where X is the device with the lower number, Y is the router with the higher number and Z is the device the address belongs to. For example, a link from P1 to P3 is 10.1.3.1 on P1 and 10.1.3.3 on P3.

The layer 3 switches have numbers from 11 to 14. This is done to avoid overlapping the addresses with the core. The PCs are connected to the switches and logically numbered per the number of the switch.



```
router isis SRLAB
 is-type level-2-only
 net 49.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid absolute 16001
  !
 !
```

Figure 11. The basic configuration of the routers

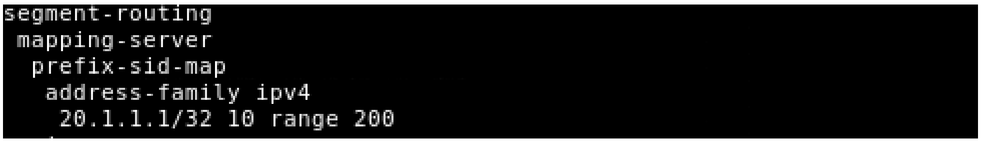
Figure 11 shows the basic IS-IS and segment routing configuration on the routers. Prefix-SIDs are allocated based on the router's name. For example, P1 is allocated the prefix-SID 16001. Loopback addresses follow the same logic. In this case, P1 has the loopback address 10.0.0.1.

5.2 MAPPING SERVER

A mapping server can be configured on any IOS-XR router in the network. The router does not have to be in the path of the traffic to work. In this case, P1 was used as the mapping server and P3 as the mapping client. A hypothetical address space of 20.1.1.1/32 – 20.1.1.200/32 is used to simulate the LDP domain routers. Mapping server functionality can be enabled with the following configuration:

```
router isis SRLAB
address-family ipv4 unicast
segment-routing prefix-sid-map advertise-local
```

Advertise-local makes the router advertise its own prefix-sid-mappings to other routers that have been configured as clients.

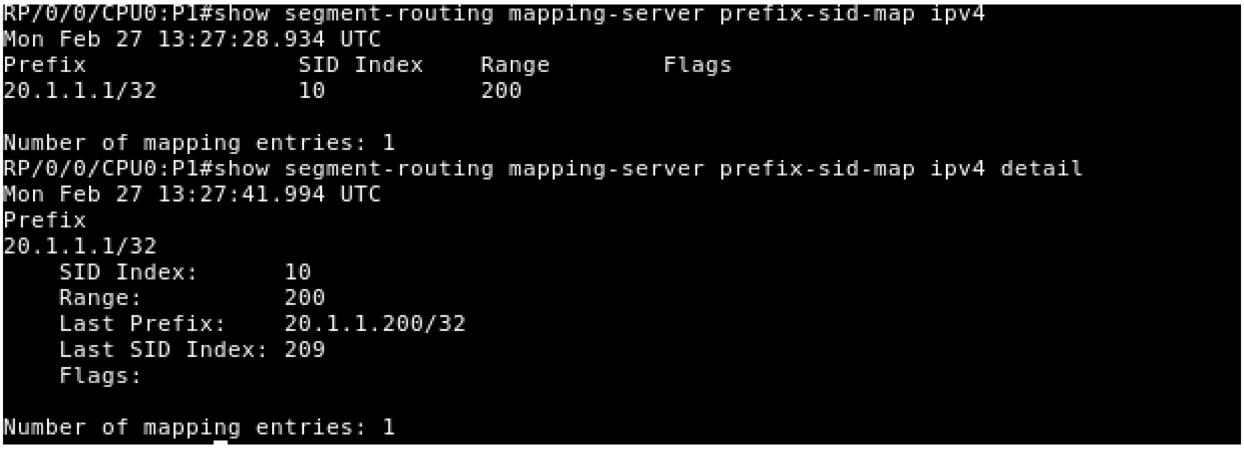


```
segment-routing
 mapping-server
  prefix-sid-map
   address-family ipv4
    20.1.1.1/32 10 range 200
```

Figure 12. Prefix-sid-map configuration

Prefix-sid-mappings can be configured in the segment-routing mapping-server configuration shown in figure 12. Prefix-sid-map entry consists of the IP address and mask, SID index value and range. A SID index of 10 means that the mapping server allocates SIDs starting from 16010. Range defines the final SID that can be allocated, which is 16209 in this case.

Prefix-sid-map configuration can be verified with the following commands, seen in figure 13:



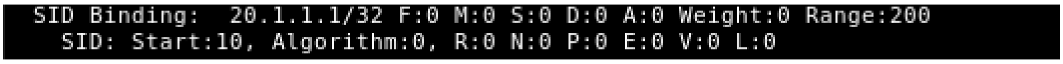
```
RP/0/0/CPU0:P1#show segment-routing mapping-server prefix-sid-map ipv4
Mon Feb 27 13:27:28.934 UTC
Prefix               SID Index    Range        Flags
20.1.1.1/32          10           200

Number of mapping entries: 1
RP/0/0/CPU0:P1#show segment-routing mapping-server prefix-sid-map ipv4 detail
Mon Feb 27 13:27:41.994 UTC
Prefix
20.1.1.1/32
    SID Index:      10
    Range:          200
    Last Prefix:    20.1.1.200/32
    Last SID Index: 209
    Flags:

Number of mapping entries: 1
```

Figure 13. Prefix-sid-map verification

A mapping client can be configured with the following commands:

```
router isis SRLAB
address-family ipv4 unicast
segment-routing prefix-sid-map receive
```

Mapping client functionality can be verified from the routers IS-IS database. *Show isis database verbose P1.00-00* shows IS-IS database entries related to P1, which is the mapping server. The output in figure 14 shows that the prefix-sid-mapping has been received from P1:

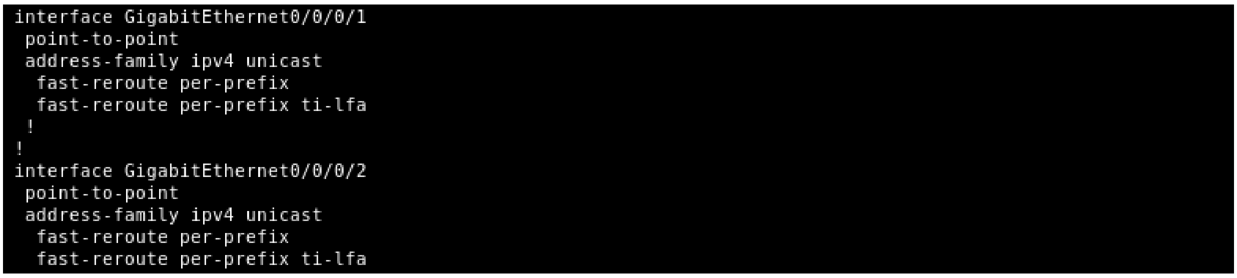


```
  SID Binding:  20.1.1.1/32 F:0 M:0 S:0 D:0 A:0 Weight:0 Range:200
    SID: Start:10, Algorithm:0, R:0 N:0 P:0 E:0 V:0 L:0
```

Figure 14. Prefix-sid-map propagated from P1 to P3

5.3 TI-LFA

TI-LFA can be configured on the routers interfaces to protect its links with the path protection feature. The configuration of path protection in advance allows the routers to calculate backup paths and switch to them almost immediately. In the event of a link failure, the router can then switch to the backup path in 50 milliseconds or less.



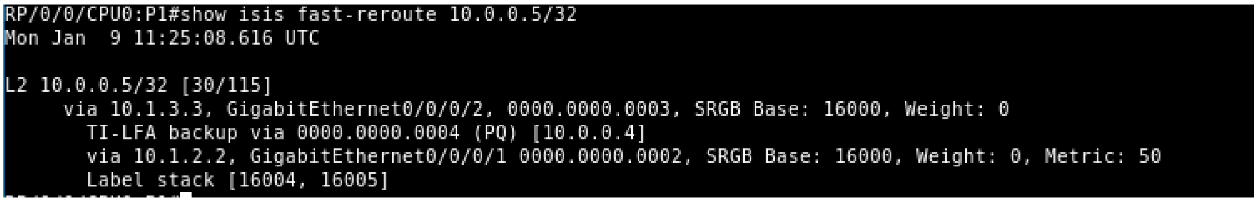
```
interface GigabitEthernet0/0/0/1
 point-to-point
 address-family ipv4 unicast
  fast-reroute per-prefix
  fast-reroute per-prefix ti-lfa
 !
!
interface GigabitEthernet0/0/0/2
 point-to-point
 address-family ipv4 unicast
  fast-reroute per-prefix
  fast-reroute per-prefix ti-lfa
```

Figure 15. TI-LFA path protection can be configured on a link-to-link basis

In this example, TI-LFA is used to protect the link between P1 and P5. Figure 15 shows the additional configuration lines that are needed to enable this technique under router IS-IS mode. The following command will show the backup path to the address defined in the command:

```
show isis fast-reroute 10.0.0.5/32
```

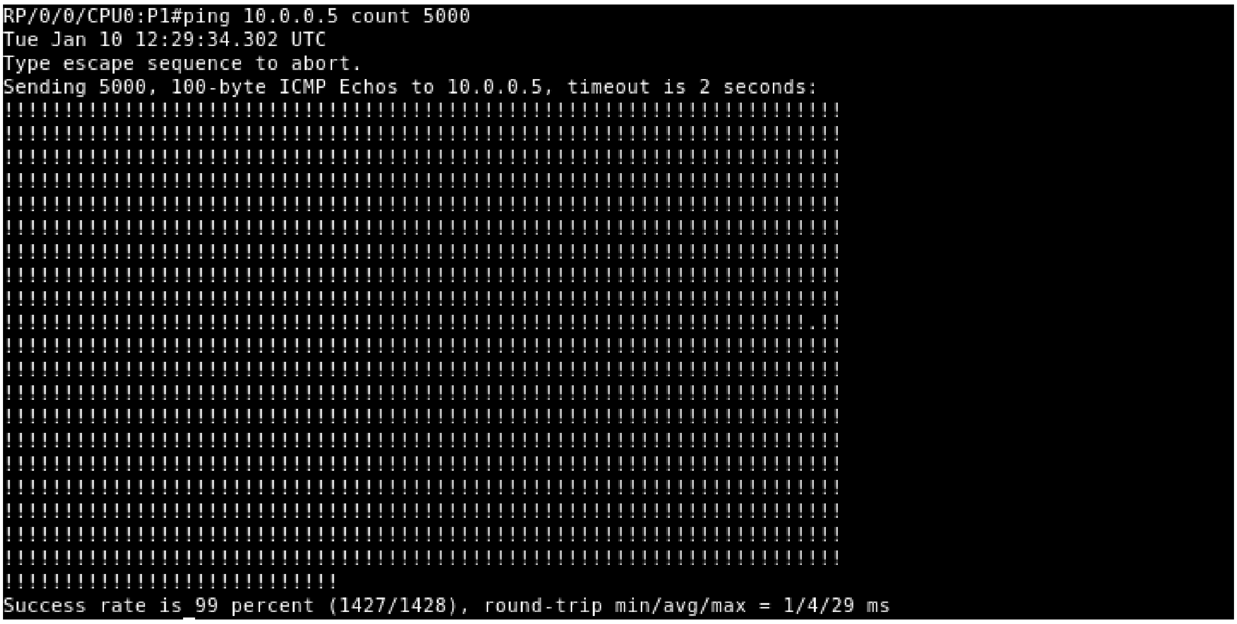
This command shows the normal path first, followed by the backup path that will be taken in the event of a failure.



```
RP/0/0/CPU0:P1#show isis fast-reroute 10.0.0.5/32
Mon Jan  9 11:25:08.616 UTC

L2 10.0.0.5/32 [30/115]
     via 10.1.3.3, GigabitEthernet0/0/0/2, 0000.0000.0003, SRGB Base: 16000, Weight: 0
       TI-LFA backup via 0000.0000.0004 (PQ) [10.0.0.4]
       via 10.1.2.2, GigabitEthernet0/0/0/1 0000.0000.0002, SRGB Base: 16000, Weight: 0, Metric: 50
       Label stack [16004, 16005]
```

Figure 16. Backup path via P2 and P4

In figure 16, TI-LFA calculation has chosen P4 as the backup router that will forward the traffic to P5. P4 can forward the traffic to P5 via either P3 or P6 because no metrics have been configured. Metrics can be used to direct the traffic flow through the route that is the most desirable.



```
RP/0/0/CPU0:P1#ping 10.0.0.5 count 5000
Tue Jan 10 12:29:34.302 UTC
Type escape sequence to abort.
Sending 5000, 100-byte ICMP Echos to 10.0.0.5, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 99 percent (1427/1428), round-trip min/avg/max = 1/4/29 ms
```

Figure 17. Testing TI-LFA by shutting down a link

Shutting down the link between P1 and P3 demonstrates the effectiveness of TI-LFA, which can be seen in figure 17. The traffic is forwarded to the backup path shown earlier in less than 50 milliseconds. However, this configuration is not enough to protect the traffic from a node failure. If the node P3 fails completely, it will take the traffic a much longer time to recover. This delay can be critical in networks that require consistent low packet loss. The following configuration enables node protection in a single segment routing node:

```
interface GigabitEthernet0/0/0/2
point-to-point
address-family ipv4 unicast
fast-reroute per-prefix
fast-reroute per-prefix tiebreaker node-protecting index 100
fast-reroute per-prefix ti-lfa
```

The current version of IOS-XRv used in the virtual laboratory does not support node protection. This configuration can be used in the future IOS-XRv versions and has been tested on a device that supports it. Node-protecting command can also be configured as default in the IS-IS instance.

5.4 LOAD BALANCING

Load-balancing is done in a segment routing domain by ECMP automatically. ECMP is enabled by default and will affect all paths that have the same metric value and destination. For example, a path from router P1 to router P6 has two equal cost path options when sending data.

Segment routing uses IS-IS to distribute labels instead of MPLS LDP. MPLS LDP uses a mechanism called Entropy label to load balance traffic between different paths. IOS-XRv 6.0.1 is not able to signal EL capability via IS-IS and therefore it cannot be demonstrated in this thesis. However, this feature will most likely be implemented in the future IOS-XRv releases.

5.5 ANYCAST-SID

Anycast-SID enables the usage of disjoint paths in the network. The goal is to separate two services so that they never go through the same router. In this case, a dual-plane network design is used to achieve the desired result.

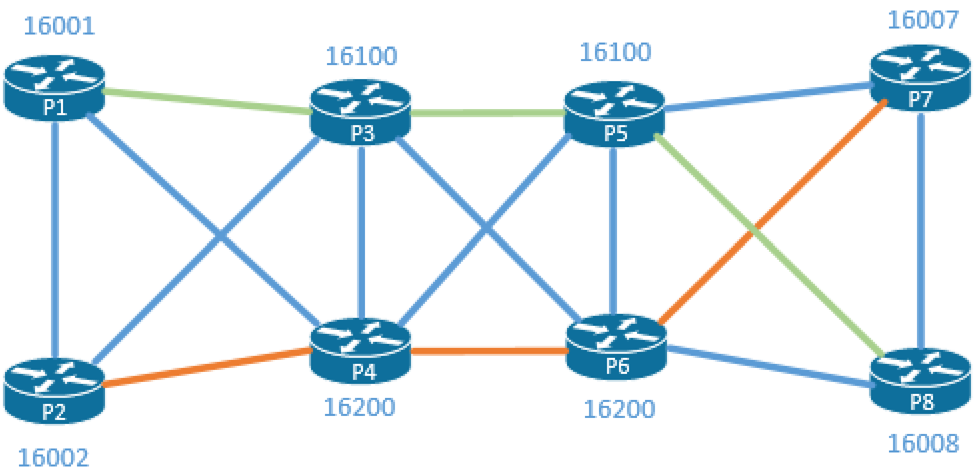


Figure 18. Two disjoint services

In figure 18, the routers P3 and P5 have been configured with the prefix-SID of 16100. P4 and P6 have the prefix-SID of 16200. The cross links in the core have been configured with metric 100 to keep the traffic from using the same path. These links will only be used if one of the paths fail.

The P1 to P8 service uses an SR-TE tunnel with segments 16100, 16008 to reach P8. Similarly, P2 to P7 is configured with the segments 16200, 16007.

Anycast-SID can be configured by assigning a secondary SID for the router. Firstly, a loopback1 with address 10.0.0.100/32 is configured to P3 and P5. Loopback1 is then added to the SRLAB IS-IS configuration. SID is assigned by issuing the command *prefix-sid absolute 16100 n-flag-clear*. N-flag is enabled by default and it makes the prefix-SID to be considered as a node SID. Anycast-SID is not linked to a single router so n-flag must be cleared. The same loopback1 configuration is repeated on P4 and P6 but with the IP address 10.0.0.200 and SID 16200.

With a properly configured PCE, it is possible to use the two disjoint paths to separate traffic with different requirements. For example, 16200 might have higher latency but more bandwidth while 16100 has low latency and low bandwidth as shown in the figure below.

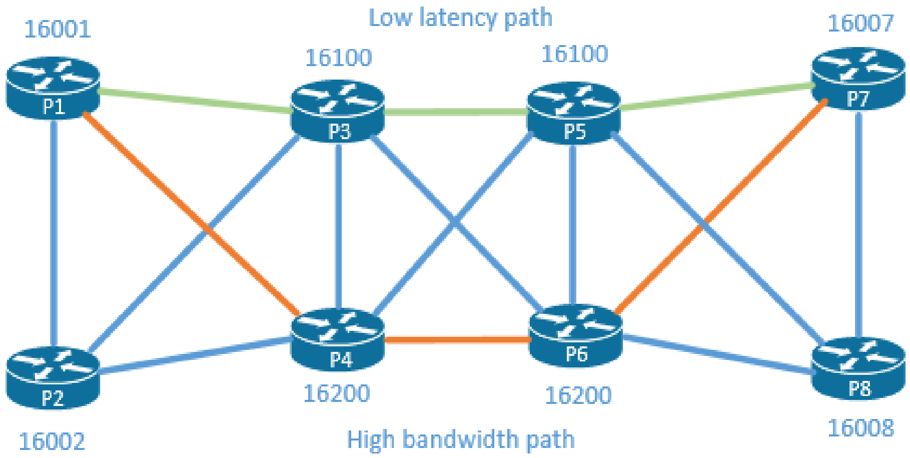


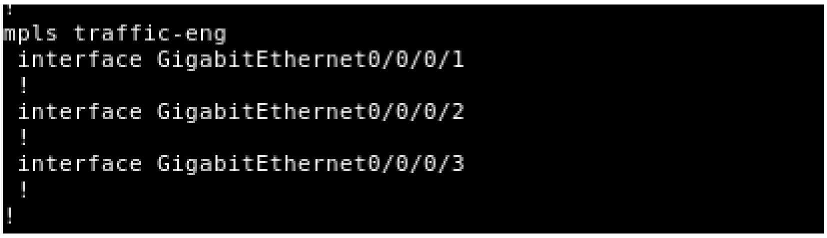
Figure 19. Two anycast-SID paths.

Low latency path could be used for services that are more sensitive to latency. The PCE recognizes the difference between the paths and directs the traffic to 16100 or 16200 depending on the requirements of the traffic. The anycast-SIDs can also be used with SR-TE tunnels.

5.6 SR-TE

Segment routing traffic engineering configuration enables the usage of explicit paths to any given destination within the segment routing domain. The configuration of SR-TE consists of tunnel configuration, altering the IS-IS configuration, explicit-path configuration and enabling MPLS-TE. The goal is to make a tunnel that takes the following path: P1 – P3 – P5 – P7 – P8 – P6 – P4 – P2.

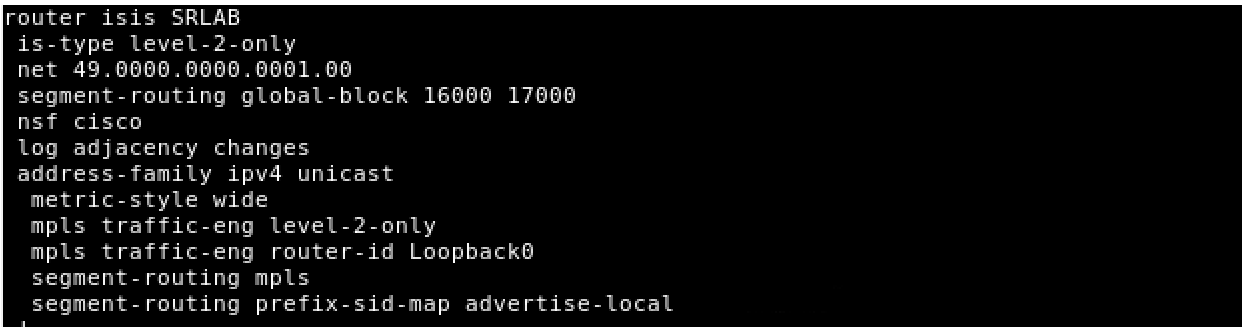
The first part of the configuration is to enable MPLS-TE and configure it to the IS-IS instance. This can be done by configuring mpls traffic-engineering in the global configuration mode. MPLS-TE must be enabled on every router in the IS-IS instance. Figure 20 shows the required MPLS-TE configuration.



```
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
 interface GigabitEthernet0/0/0/3
 !
!
```

Figure 20. MPLS TE configuration

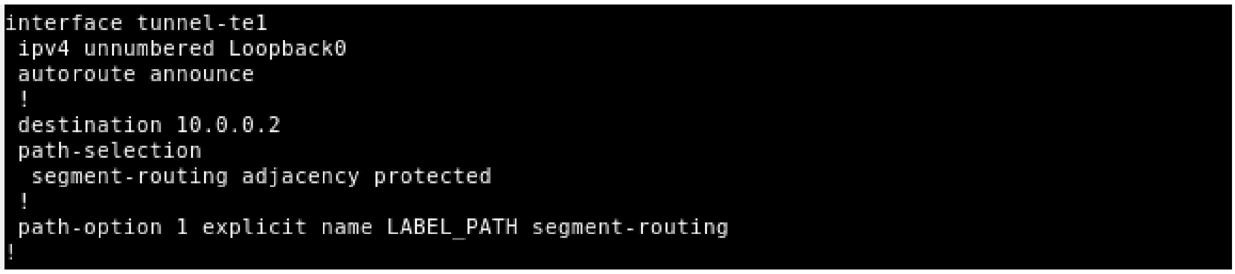
The IS-IS configuration in this case is configured under router isis SRLAB instance. It is important to note that the MPLS traffic-eng level configuration must be the same across all routers or the configuration will not work. Router-id Loopback0 causes the SR-TE to use Loopback0 address as the router-id. Figure 21 shows the complete configuration of router isis SRLAB on P1.



```
router isis SRLAB
 is-type level-2-only
 net 49.0000.0000.0001.00
 segment-routing global-block 16000 17000
 nsf cisco
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
  mpls traffic-eng level-2-only
  mpls traffic-eng router-id Loopback0
  segment-routing mpls
  segment-routing prefix-sid-map advertise-local
```

Figure 21. SR-TE configuration in the IS-IS instance

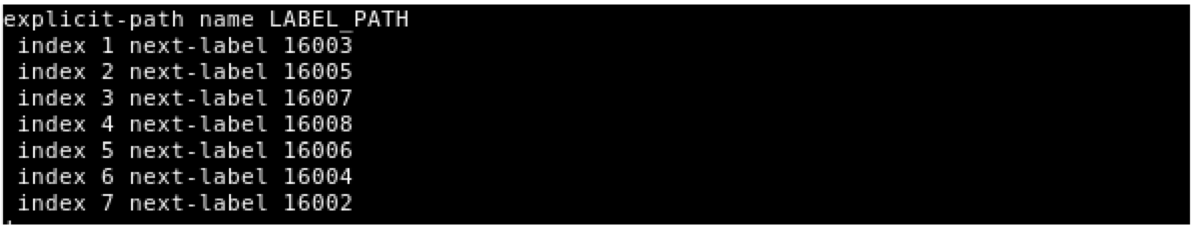
Once the basic SR-TE configuration is done, one must configure an SR-TE tunnel with the correct parameters. SR-TE tunnel also requires the configuration of explicit-path list that will then be attached to the SR-TE tunnel.



```
interface tunnel-te1
 ipv4 unnumbered Loopback0
 autoroute announce
 !
 destination 10.0.0.2
 path-selection
  segment-routing adjacency protected
 !
 path-option 1 explicit name LABEL_PATH segment-routing
!
```

Figure 22. SR-TE tunnel configuration

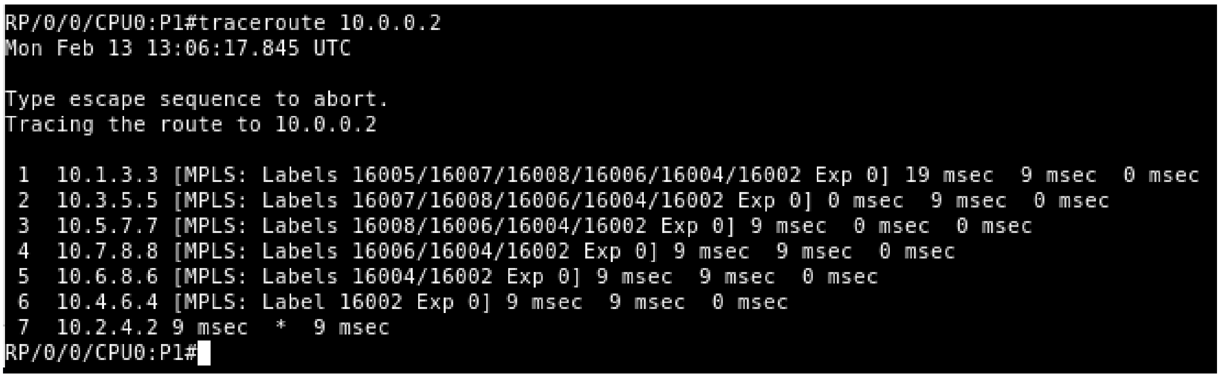
SR-TE tunnel configuration in figure 22 consists of the configuration of the tunnel address, advertising the tunnel to the IS-IS instance, destination, SR-TE configuration and choosing the path option. It is important not to forget *auto-route announce.* Skipping this step will prevent the tunnel from being seen by the IS-IS instance and thus no traffic will use the tunnel.



```
explicit-path name LABEL_PATH
 index 1 next-label 16003
 index 2 next-label 16005
 index 3 next-label 16007
 index 4 next-label 16008
 index 5 next-label 16006
 index 6 next-label 16004
 index 7 next-label 16002
```

Figure 23. Explicit-path configuration

Explicit-path configuration can be done using labels or addresses. Figure 23 uses labels in the explicit-path configuration. Index followed by a number determines the order in which the steps are executed. It is important to consider that all routers on the explicit-path must support the label stack depth used in the tunnel. In this case, the label stack is seven labels deep.



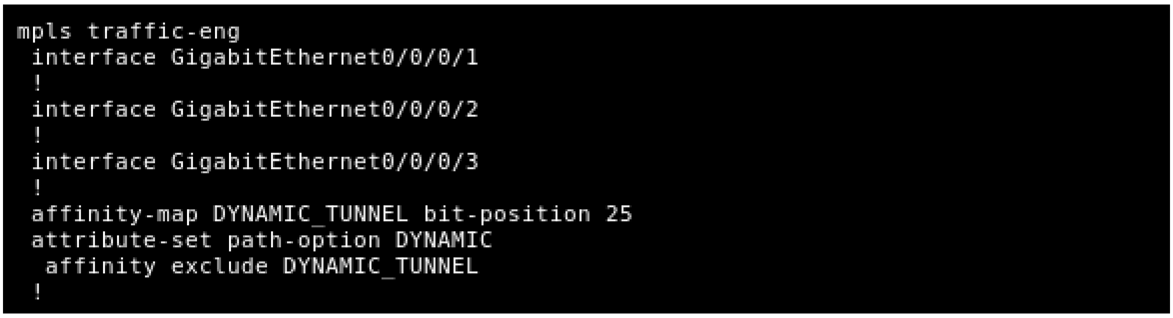
```
RP/0/0/CPU0:P1#traceroute 10.0.0.2
Mon Feb 13 13:06:17.845 UTC

Type escape sequence to abort.
Tracing the route to 10.0.0.2

 1  10.1.3.3 [MPLS: Labels 16005/16007/16008/16006/16004/16002 Exp 0] 19 msec  9 msec  0 msec
 2  10.3.5.5 [MPLS: Labels 16007/16008/16006/16004/16002 Exp 0] 0 msec  9 msec  0 msec
 3  10.5.7.7 [MPLS: Labels 16008/16006/16004/16002 Exp 0] 9 msec  0 msec  0 msec
 4  10.7.8.8 [MPLS: Labels 16006/16004/16002 Exp 0] 9 msec  9 msec  0 msec
 5  10.6.8.6 [MPLS: Labels 16004/16002 Exp 0] 9 msec  9 msec  0 msec
 6  10.4.6.4 [MPLS: Label 16002 Exp 0] 9 msec  9 msec  0 msec
 7  10.2.4.2 9 msec  *  9 msec
RP/0/0/CPU0:P1#
```

Figure 24. Traceroute using the SR-TE tunnel

The functionality of the SR-TE tunnel can be tested by tracerouting the tunnel destination as demonstrated in figure 24. A traceroute from router P1 to the adjacent router P2 takes a long path around the network before finally reaching its destination seven hops later. Each hop of the traceroute demonstrates how the segment routing pops a label from the packet on each hop on the way to its destination.

5.6.1 SR-TE DYNAMIC TUNNELS

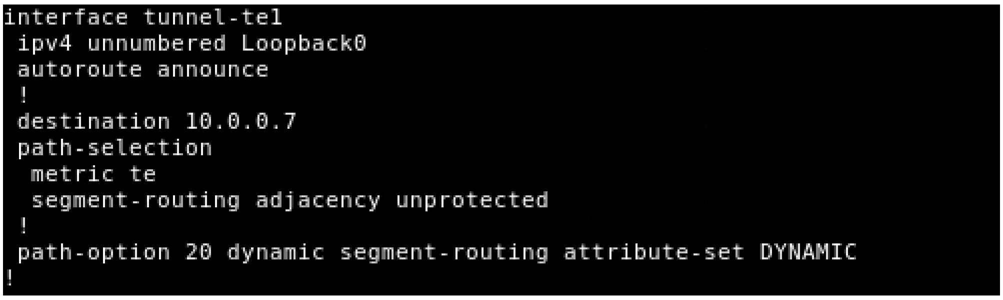
SR-TE can be configured to use a dynamic path in its tunnels. This method is useful if a link or node fails in the SR-TE tunnel’s path. A normal explicit-path will not work if a failure occurs and IGP will be used as a last resort. However, with dynamic paths, the tunnel can adjust its route and keep the SR-TE tunnel functional.



```
mpls traffic-eng
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
 interface GigabitEthernet0/0/0/3
 !
 affinity-map DYNAMIC_TUNNEL bit-position 25
 attribute-set path-option DYNAMIC
  affinity exclude DYNAMIC_TUNNEL
 !
```

Figure 25. Dynamic SR-TE configuration

The above configuration will be attached to the SR-TE tunnel to convert it into a dynamic one. The names typed in all upper-case letters can be any names the user desires.

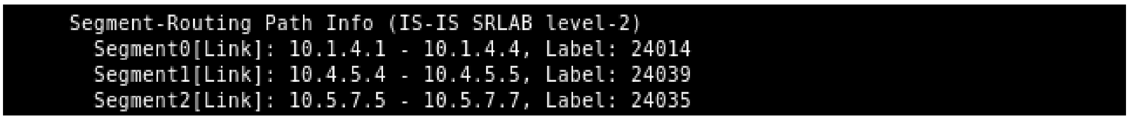


```
interface tunnel-te1
 ipv4 unnumbered Loopback0
 autoroute announce
 !
 destination 10.0.0.7
 path-selection
  metric te
  segment-routing adjacency unprotected
 !
 path-option 20 dynamic segment-routing attribute-set DYNAMIC
!
```

Figure 26. Dynamic SR-TE tunnel configuration

The tunnel configuration in figure 26 has been slightly altered from the explicit-path configuration. *Metric te* has been added under path-selection and *segment-routing adjacency protected* has been changed to *unprotected*. The path option must be changed to dynamic and attribute-set from the earlier configuration will be linked to the tunnel here.

The configuration of this tunnel makes it take P1 – P3 – P5 – P7 route by default. The dynamic behaviour can be demonstrated by shutting down all interfaces on P3 and then issuing the command “show mpls traffic-eng tunnels”.



```
     Segment-Routing Path Info (IS-IS SRLAB level-2)
       Segment0[Link]: 10.1.4.1 - 10.1.4.4, Label: 24014
       Segment1[Link]: 10.4.5.4 - 10.4.5.5, Label: 24039
       Segment2[Link]: 10.5.7.5 - 10.5.7.7, Label: 24035
```

Figure 27. Dynamic SR path

As shown in the image above, the dynamic tunnel has changed its path to P1 – P4 – P5 – P7. Labels used in this path are adjacency-SIDs.

5.7 PCEP

PCEP protocol requires some configuration changes on the PCCs end to work. Firstly, a configuration line must be added in the global configuration mode that allows the PCE to impose tunnels with loopback0 on PCC.

```
Ipv4 unnumbered mpls traffic-eng loopback0
```

Without this configuration, the imposed tunnel is not able to find the P1 loopback0 address. The following configuration must be added under mpls traffic-eng to allow the PCC to peer with PCE:

```
pce
peer source ipv4 192.168.105.1
peer ipv4 192.168.105.2
segment-routing
stateful-client
instantiation
reoptimize 60
auto-tunnel pcc
tunnel-id min 1 max 100.
```

This configuration determines the peer source which is PCC and the peer which is the PCE. Auto-tunnel PCC allows the PCE to create tunnels with id range 1-100. After this configuration, the PCC will peer with the PCE server once it is up and running.

5.7.1 PYTHON SCRIPT

PCE server can be set up on a Linux server with python-gevent and protobuf. Python-gevent and protobuf are required to run the PCE server python script used in this thesis. In this case, Ubuntu 16.04.1 is used as a PCE server.

The PCE must be connected to internet before any downloads can be made. To connect a host from the virtual laboratory to internet, one must modify the topology file of the virtual laboratory to include the following:

```
CABLE PCE 0 eth1 -1
```

With this configuration in place, the PCE can be connected to the internet after assigning a proper static IP address to it.

Before beginning the PCE configuration, python-gevent, python3 and protobuf must be installed. Python3 and python-gevent can be simply installed by using the following lines:

```
sudo apt-get install python3
sudo apt-get install python-gevent
```

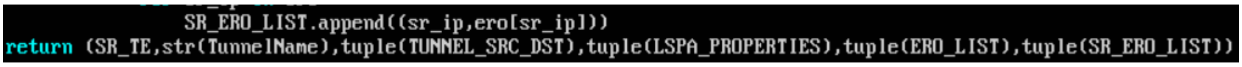
Protobuf installation requires the installation of pip first. Once pip has been successfully installed, it can be used to install protobuf as follows:

```
apt-get install python-pip
sudo pip install protobuf
```

Once the basic setup is done, PCE script files can be exported from GitHub. This thesis uses github.com/Dipsingh/mpls-pce. To import from GitHub, the following command must be used:

```
Git clone https://github.com/Dipsingh/mpls-pce
```

After the GitHub repository has been exported, an internet connection should no longer be required. To restore the previous functionality, the topology file must be changed back to the previous configuration and IP address must be changed back to the one used in virtual laboratory.

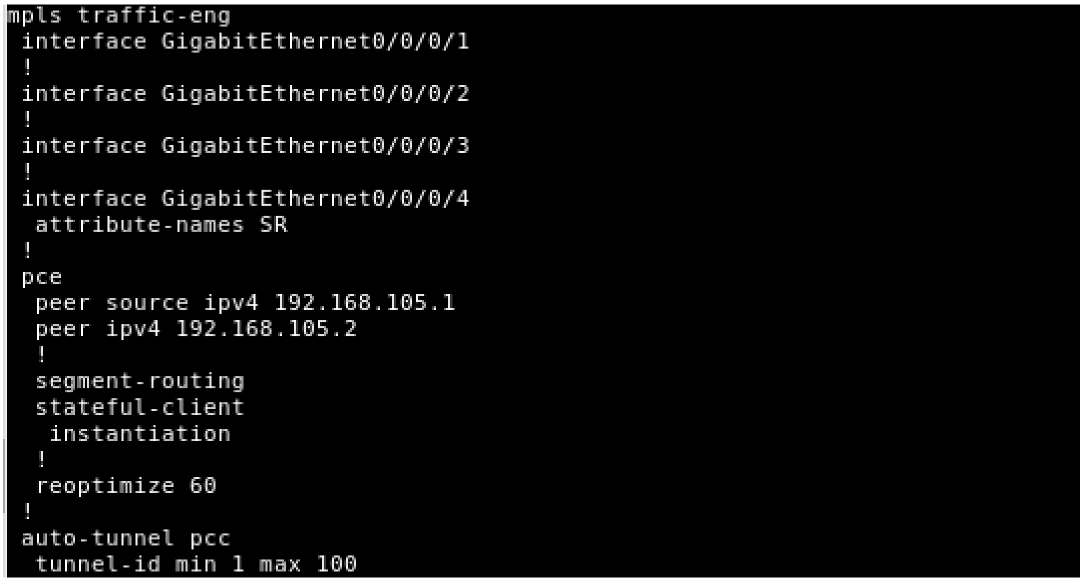


```
            SR_ERO_LIST.append((sr_ip,ero[sr_ip]))
return (SR_TE,str(TunnelName),tuple(TUNNEL_SRC_DST),tuple(LSPA_PROPERTIES),tuple(ERO_LIST),tuple(SR_ERO_LIST))
```

Figure 28. A functional version of line 52 in pce_controller.py

The code did not work on the first try and a change had to be made to make it functional. The line 52 in pce_controller.py file that originally reads “return (SR_TE, str.encode(TunnelName)” had to be changed to “str(TunnelName)”. After the change that can be seen in figure 28, the script started working properly.

However, before the PCE server can be started, the P1 must be configured to act as the PCC. Firstly, MPLS traffic-engineering configuration must be altered with the PCEP configuration. Figure 29 contains the necessary configuration changes.



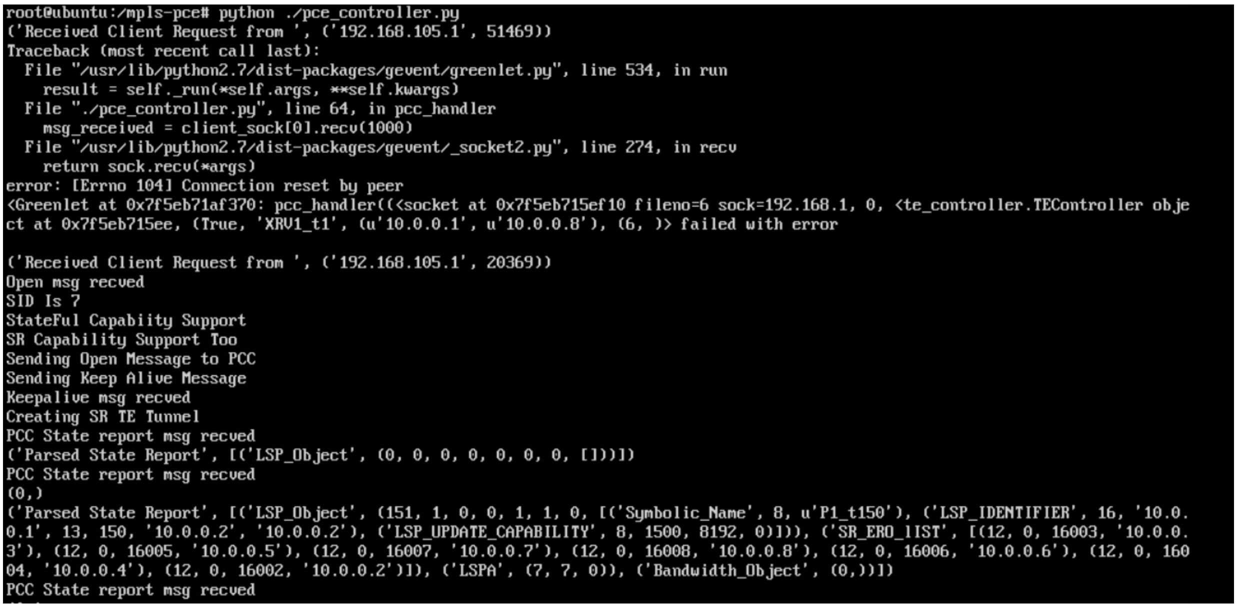
```
mpls traffic-eng
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
 interface GigabitEthernet0/0/0/3
 !
 interface GigabitEthernet0/0/0/4
  attribute-names SR
 !
 pce
  peer source ipv4 192.168.105.1
  peer ipv4 192.168.105.2
  !
  segment-routing
  stateful-client
   instantiation
  !
  reoptimize 60
 !
 auto-tunnel pcc
  tunnel-id min 1 max 100
```

Figure 29. PCE configuration under MPLS TE

IP-address 192.168.105.1 is used in GigabitEthernet0/0/0/4 and .2 on the PCE server. Tunnel-id reserves tunnel ids from 1 to 100 for the tunnels created by the PCE. All manually created tunnels must be outside of this range on the P1.

```
Interface tunnel-te150
pce
delegation
```

With the configuration above, the manually configured tunnel-te150 on the P1 has been configured to delegate the control of the tunnel to the PCE.



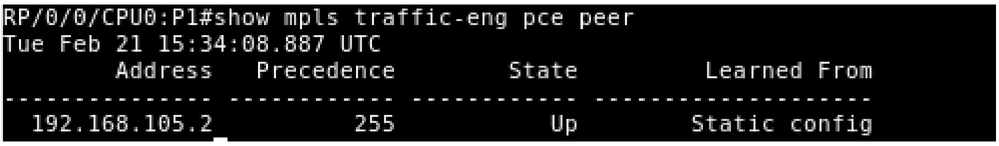
```
root@ubuntu:/mpls-pce# python ./pce_controller.py
('Received Client Request from ', ('192.168.105.1', 51469))
Traceback (most recent call last):
  File "/usr/lib/python2.7/dist-packages/gevent/greenlet.py", line 534, in run
    result = self._run(*self.args, **self.kwargs)
  File "./pce_controller.py", line 64, in pcc_handler
    msg_received = client_sock[0].recv(1000)
  File "/usr/lib/python2.7/dist-packages/gevent/_socket2.py", line 274, in recv
    return sock.recv(*args)
error: [Errno 104] Connection reset by peer
<Greenlet at 0x7f5eb71af370: pcc_handler((<socket at 0x7f5eb715ef10 fileno=6 sock=192.168.1, 0, <te_controller.TEController obje
ct at 0x7f5eb715ee, (True, 'XRV1_t1', (u'10.0.0.1', u'10.0.0.8'), (6, )> failed with error

('Received Client Request from ', ('192.168.105.1', 20369))
Open msg recved
SID Is 7
StateFul Capabiity Support
SR Capability Support Too
Sending Open Message to PCC
Sending Keep Alive Message
Keepalive msg recved
Creating SR TE Tunnel
PCC State report msg recved
('Parsed State Report', [('LSP_Object', (0, 0, 0, 0, 0, 0, 0, []))])
PCC State report msg recved
(0,)
('Parsed State Report', [('LSP_Object', (151, 1, 0, 0, 1, 1, 0, [('Symbolic_Name', 8, u'P1_t150'), ('LSP_IDENTIFIER', 16, '10.0.
0.1', 13, 150, '10.0.0.2', '10.0.0.2'), ('LSP_UPDATE_CAPABILITY', 8, 1500, 8192, 0)])), ('SR_ERO_lIST', [(12, 0, 16003, '10.0.0.
3'), (12, 0, 16005, '10.0.0.5'), (12, 0, 16007, '10.0.0.7'), (12, 0, 16008, '10.0.0.8'), (12, 0, 16006, '10.0.0.6'), (12, 0, 160
04, '10.0.0.4'), (12, 0, 16002, '10.0.0.2')]), ('LSPA', (7, 7, 0)), ('Bandwidth_Object', (0,))])
PCC State report msg recved
```

Figure 30. PCE client request from P1

PCE can be started once the configuration is done and there is connectivity between the PCE and P1. PCE can be started by issuing the following command:

```
python ./pce_controller.py
```

The script will display information about the PCE tunnel states and clients. If there are errors, a traceback message will appear. Traceback messages can be used to find the exact line where the error has occurred. Errors shown in figure 30 are a result of interrupting the script.



```
RP/0/0/CPU0:P1#show mpls traffic-eng pce peer
Tue Feb 21 15:34:08.887 UTC
        Address   Precedence        State         Learned From
--------------- ------------ ------------ --------------------
  192.168.105.2          255           Up        Static config
```

Figure 31. PCE peer information

Figure 31 shows what a successful PCE connection looks like. On the PCC, the connection to PCE can be verified by issuing the command show mpls traffic-eng pce peer. If the state reads TCP pending, the PCE script is not running or there is a connectivity issue.

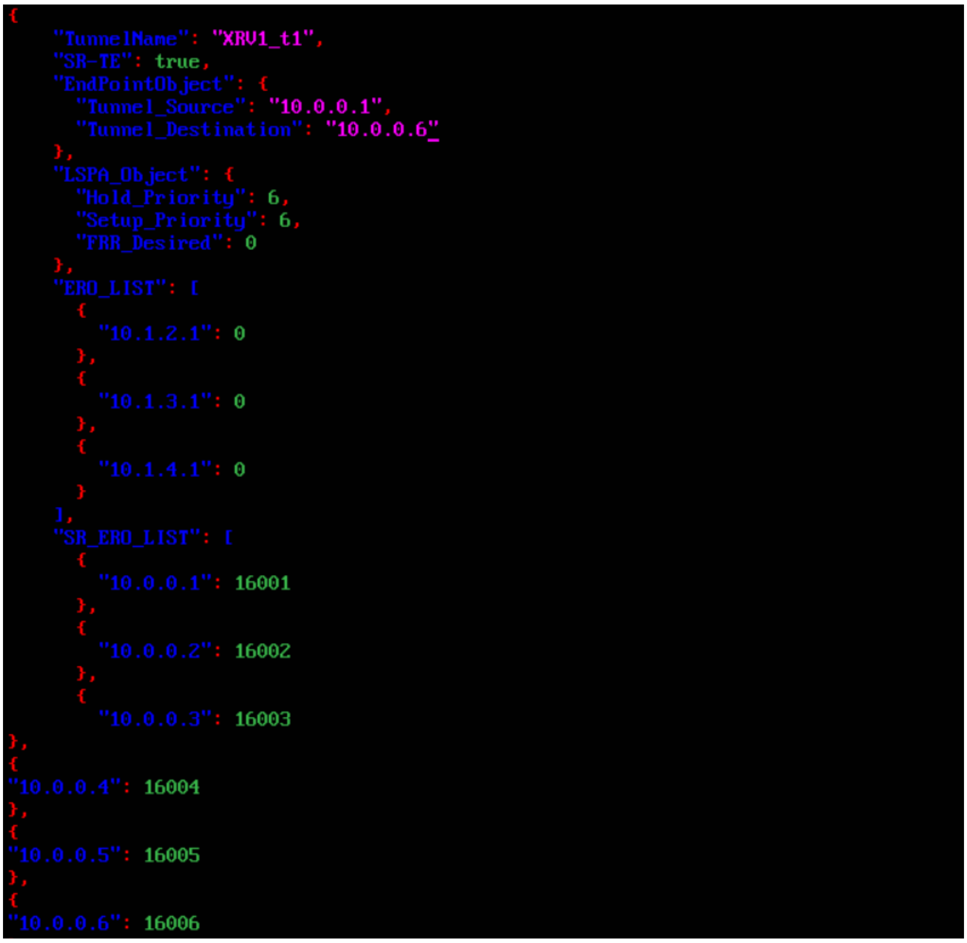
5.7.1.1 TUNNEL SETUP

PCE can be configured to remotely impose a tunnel on the PCC. One line of configuration must be added to the PCC to make the tunnel function properly:

Ipv4 unnumbered mpls traffic-eng loopback0

This will allow the loopback0 address to be used as a source address in the tunnel configuration imposed by the PCE.

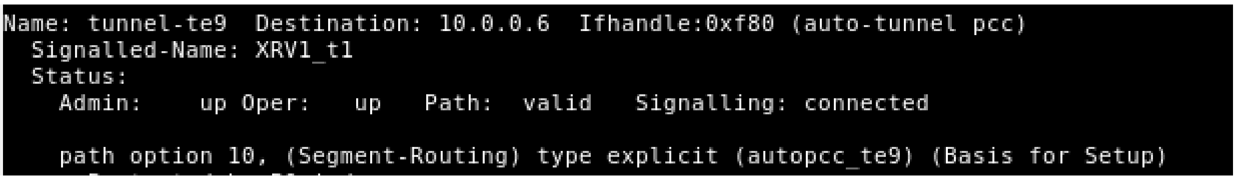
Tunnel configuration seen in figure 32 can be modified in PCE_Config.json that is part of the mpls-pce folder.



```
{
    "TunnelName": "XRV1_t1",
    "SR-TE": true,
    "EndPointObject": {
      "Tunnel_Source": "10.0.0.1",
      "Tunnel_Destination": "10.0.0.6"
    },
    "LSPA_Object": {
      "Hold_Priority": 6,
      "Setup_Priority": 6,
      "FRR_Desired": 0
    },
    "ERO_LIST": [
      {
        "10.1.2.1": 0
      },
      {
        "10.1.3.1": 0
      },
      {
        "10.1.4.1": 0
      }
    ],
    "SR_ERO_LIST": [
      {
        "10.0.0.1": 16001
      },
      {
        "10.0.0.2": 16002
      },
      {
        "10.0.0.3": 16003
},
{
"10.0.0.4": 16004
},
{
"10.0.0.5": 16005
},
{
"10.0.0.6": 16006
```

Figure 32. SR-TE tunnel configuration on the PCE

The tunnel source address is Loopback0 and the destination is the loopback address in the other end of the tunnel. SR_ERO_LIST must be modified to include the explicit-route used in the SR-TE tunnel. Once the configuration is complete, the file can be saved and the PCE process started again.



```
Name: tunnel-te9  Destination: 10.0.0.6  Ifhandle:0xf80 (auto-tunnel pcc)
  Signalled-Name: XRV1_t1
  Status:
    Admin:     up Oper:    up    Path:  valid    Signalling: connected

    path option 10, (Segment-Routing) type explicit (autopcc_te9) (Basis for Setup)
```

Figure 33. Tunnel information on the PCC

If the configuration is successful, there should be no traceback errors about the tunnel configured in PCE_Config.json. Turning on the PCE again will impose the tunnel on the P1 router. The command show mpls traffic-engineering tunnels will now show the tunnel imposed by the PCE as shown in figure 33.

5.7.2 OPENDAYLIGHT

A more advanced version of PCEP controller can be implemented with OpenDaylight Beryllium software. The software can be downloaded from the OpenDaylight website with wget. OpenDaylight also required Java to work. *Apt-get install default-jdk* was used in this demonstration.

After both pieces of software have been installed, the OpenDaylight software can be started by going to the *distribution-karaf-0.4.4-Beryllium-SR4* folder and issuing the following command:

```
./bin/karaf
```

OpenDaylight features can be installed with feature:install followed by the desired feature. To view currently installed features, feature:install -i can be used. The following features were installed:

```
Feature:install odl-bgpcep-pcep
Feature:install odl-dlux-all
Feature:install odl-restconf-all
```

Dlux is a web interface that can be used to configure PCEP parameters. Restconf is another feature that can be used to view OpenDaylight configuration in a web browser.

The basic PCE controller functionality is preconfigured in the odl-bgpcep-pcep package. Connectivity to the PCC can be tested by issuing the command *show mpls traffic-eng pce peer* on the PCC. If configuration is done correctly, the peer should be up and the tunnel should be delegated to the PCE. Tunnel delegation can be checked with the command *show mpls traffic-eng tunnels*.

Restconf can be used to view the PCEP configuration at *http://localhost:8181/restconf/operational/network-topology:network-topology/topol-*

ogy/pcep-topology. LSP can be added by configuring it at *http://localhost:8181/restconf/operations/network-topology-pcep:add-lsp* per the opendaylight wiki. However, this feature returns an HTTP error 500.

DLUX can be accessed at *http://localhost:8181/index.html*. LSP configuration can be found in the Yang UI tab of the web interface. LSP parameters can be configured using the URL *http://localhost:8181/restconf/operations/network-topology-pcep:add-lsp*. Additional configuration might be required, because none of the requests made with the UI went to the router.

5.8 IPV6

IOS-XRv 6.0.1 segment routing does not have a fully implemented IPv6 support. The IS-IS instance does not allow the configuration of IPv6 based segment routing, which means that most features cannot be configured with IPv6. The lack of IPv6 based segment routing means that MPLS labels must be used anyway, which defeats the purpose of IPv6 segment routing.

SR-TE does not support IPv6 when configuring the explicit paths and destinations. This makes SR-TE unusable in an IPv6 only environment. The support for these techniques is in development and will be implemented later.

Address-family ipv6 unicast

Fast-reroute per-prefix

Fast-reroute per-prefix ti-lfa

TI-LFA is currently the only technique that supports IPv6. IPv6 TI-LFA can be configured to interfaces in the same way as it is done in IPv4.

6 CONCLUSIONS

Segment routing is an important addition to the service provider networks. There are many improvements compared to RSVP-TE and MPLS LDP. For example, segment routing allows the network to become scalable and programmable. Segment routing also makes the configuration of the network drastically easier.

Features like the mapping server, anycast-SID, disjointness, SR-TE, TI-LFA and the PCEP protocol were successfully implemented in the thesis. These techniques are important for any large-scale network that plans to implement segment routing.

There were some difficulties with the implementation of Opendaylight based PCEP. The creation and teardown of tunnels did not work properly. However, the python based PCE offers a simple demonstration of the PCE server's capabilities. The way PCEP protocol is implemented into the network is currently undergoing major changes. The future IOS-XRv versions will have a router-based PCE server called the the XR transport controller.

Some features were not implemented in IOS-XRv 6.0.1. Anycast-SID policies, disjointness policies and the improved SR-TE capabilities were not implemented in this version of the router. However, the newer versions introduce many additions and noticeable changes to the configuration.

From a personal perspective, the segment routing thesis offered a great learning opportunity. Most of the techniques presented in this thesis were completely foreign to me when I began studying the subject. However, the relatively simple design of the techniques allowed me to learn them quickly.

All in all, the thesis successfully explored the segment routing options available in IOS-XRv 6.0.1. There is still plenty of room for more research on the subject, especially with the release of the new version. Segment routing is still being actively developed by Cisco and its partners. It will be an integral part of the future of the service provider networks.

7 FUTURE DEVELOPMENT

This thesis covers the configuration of segment routing in IOS XRv 6.0.1. However, the release of IOS XRv 6.2.0 will introduce a wide array of changes to the configuration. The 6.2.0 version will also add support to techniques that could not be used before. (Segment-routing 2017.)

The SR-TE configuration has been moved under a segment-routing traffic-eng submode. SR-TE tunnels can be configured with policies that make the tunnels more customizable. For example, links can be colour coded, which allows configuring constraints such as “avoid red” or “use only red”. Other constraints include but are not limited to IP-addresses, metric, SRLG and maximum SID count. (Segment-routing 2017.)

6.2.0 introduces the XTC (XR transport controller). XTC is a stateful PCE that runs on the router itself, which eliminates the need for an external PCE server. XTC has the same functionality as a normal PCE, but it does not necessarily require MPLS on the router. Multiple XTCs can be used in a single network to achieve high availability. (Cisco Systems 2017b.)

Weighted ECMP allows the traffic to be load-balanced between two tunnels. Dual-plane and service disjointness configuration have also been simplified. (Segment-routing 2017.)

REFERENCES

Aubry, F. 2015. Traffic duplication through segmentable disjoint paths. Available at: http://inl.info.ucl.ac.be/system/files/paper_9.pdf [Accessed: 5 December 2016].

Butler, B. 2016. SD-WAN: What it is and why you'll use it one day. Available at: http://www.networkworld.com/article/3031279/internet/sd-wan-what-it-is-and-why-you-ll-use-it-one-day.html [Accessed: 17 December 2016].

Cisco Systems, Inc. 2005. MPLS Label Distribution Protocol (LDP). Available at: http://www.cisco.com/c/en/us/td/docs/ios/12_4t/12_4t2/ftldp41.html [Accessed: 15 December 2016].

Cisco Systems, Inc. 2011. MPLS LDP Configuration Guide, Cisco IOS Release 12.4. Available at: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mp_ldp/configuration/12-4m/mp-ldp-12-4m-book.pdf [Accessed: 16 February 2017].

Cisco Systems, Inc. 2015. Segment Routing: Introduction and Value Proposition. Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/application-engineered-routing/white-paper-c11-734250.html [Accessed: 26 December 2016].

Cisco Systems, Inc. 2016. Application Driven Routing – Segment Routing. Available at: https://supportforums.cisco.com/document/12730586/application-driven-routing-segment-routing [Accessed: 29 December 2016].

Cisco Systems, Inc. 2017a. ASR9000/XR: Load-balancing architecture and characteristics. Available at: https://supportforums.cisco.com/document/111291/asr9000xr-load-balancing-architecture-and-characteristics [Accessed 24 April 2017].

Cisco Systems, Inc. 2017b. Release Notes for Cisco ASR 9000 Series Routers, IOS XR Release 6.2.1. Available at: http://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-2/general/release/notes/b-release-note-asr9k-621.html [Accessed: 15 March 2017].

Filsfils, C. 2015. Segment Routing Architecture. Available at: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-06#section-3.5 [Accessed: 12 December 2016].

Juniper Networks. 2012. Understanding MPLS Label Operations on EX Series Switches. Available at: https://www.juniper.net/documentation/en_US/junos15.1/topics/concept/mpls-label-operations-ex-series.html [Accessed: 17 October 2016].

Juniper Networks. 2015. Understanding Entropy Label for BGP Labeled Unicast LSP. Available at: https://www.juniper.net/documentation/en_US/junos/topics/concept/bgp-entropy-label.html [Accessed: 3 March 2017].

Juniper Networks. 2016. Understanding the RSVP Signalling Protocol. Available at: http://www.juniper.net/documentation/en_US/junos15.1/topics/concept/mpls-security-rsvp-signaling-protocol-understanding.html [Accessed: 20 December 2016].

Koivula, R. 2016. Core Hiding Migration. Available at: http://www.theseus.fi/bitstream/handle/10024/113707/Koivula_Riku.pdf?sequence=1 [Accessed: 21 December 2016].

Lerner, A. 2015. Predicting SD-WAN Adoption. Available at: http://blogs.gartner.com/andrew-lerner/2015/12/15/predicting-sd-wan-adoption/ [Accessed: 17 December 2016].

Linuxfoundation. 2015. Future of PCE. Available at: http://events.linuxfoundation.org/sites/events/files/slides/FutureofPCE.pdf [Accessed: 6 December 2016].

Mplsinfo. 2009. What is MPLS? Available at: http://www.mplsinfo.org/ [Accessed: 24 October 2016].

Nokia Oyj. 2016a. Fast reroute with segment routing. Available at: https://resources.alcatel-lucent.com/asset/185758 [Accessed 28 November 2016].

Nokia Oyj. 2016b. Segment Routing and Path Computation Element. Available at: http://resources.alcatel-lucent.com/asset/186949 [Accessed: 29 November 2016].

Nurmi, J. 2016. Implementation of Nested Virtual Laboratory System. Available at: http://www.theseus.fi/bitstream/handle/10024/107061/Nurmi_Jaakko_Thesis.pdf?sequence=1 [Accessed: 21 December 2016].

Open Networking Foundation. 2016. Software-Defined Networking (SDN) Definition. Available at: https://www.opennetworking.org/sdn-resources/sdn-definition [Accessed: 17 December 2016].

Previdi, S. 2015. IS-IS Extensions for Segment Routing. Available at: https://tools.ietf.org/html/draft-ietf-isis-segment-routing-extensions-06#section-2.1 [Accessed: 8 December 2016].

Rouse, M. 2014. Multiprotocol Label Switching (MPLS). Available at: http://searchenterprisewan.techtarget.com/definition/Multiprotocol-Label-Switching [Accessed: 19 December 2016].

Sánchez-Monge, A. 2015. MPLS in the SDN Era. Available at: https://books.google.fi/books?id=pBooCwAAQBAJ&lpg=PA722&dq=ti-lfa%20srlg%20protection&hl=fi&pg=PA722#v=onepage&q=ti-lfa%20srlg%20protection&f=false [Accessed: 1 December 2016].

Segment-routing. 2017. Segment Routing Traffic Engineering (SRTE). Available at: http://www.segment-routing.net/tutorials/2017-03-06-segment-routing-traffic-engineering-srte/ [Accessed 23 March 2017].

Singh, D. 2016. Yet Another Blog About Segment Routing, Part2: TI-LFA. Available at: http://packetpushers.net/yet-another-blog-about-segment-routing-part2-ti-lfa/ [Accessed: 29 December 2016].

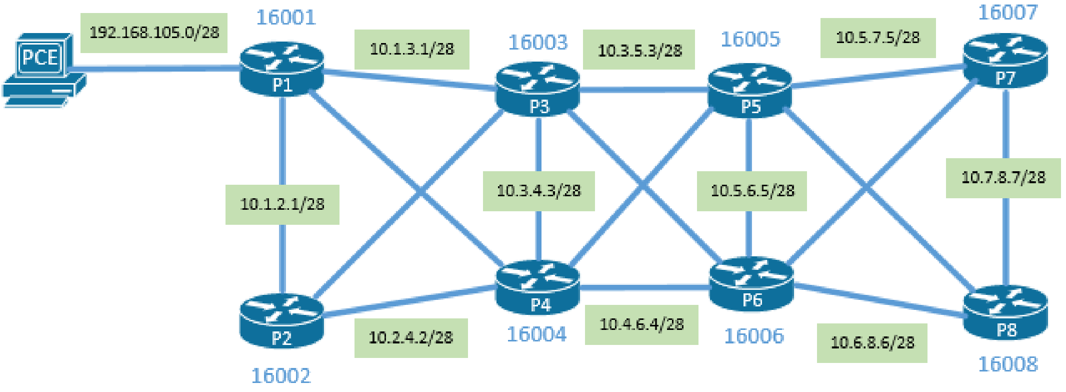
Trate, F. 2016. Bringing Segment Routing and IPv6 together. Available at: http://blogs.cisco.com/sp/bringing-segment-routing-and-ipv6-together [Accessed: 5 January 2017].

APPENDICES

SEGMENT ROUTING

CASE STUDY

06.03.2017



IP addressing is in the form of 10.x.y.z where x is the lower router number, y is the higher router number and z is the router the cable is attached to.

BASIC CONFIGURATION

1. Configure the IP addressing

    For example, a link from P1 to P3 is 10.1.3.1 on P1 and 10.1.3.3 on P3. Loopback addresses are 10.0.0.x according to the router number. Make sure that the interfaces are configured as point-to-point.
2. Configure the IS-IS instance.

    Use net 49.0000.0000.000x.00 where x is the number of the router.
    Add all required interfaces to the IS-IS instance.
    Use *metric-style wide*
3. Verify connectivity between the routers.
4. Use *show isis database* and *show isis topology* for detailed information about the IS-IS instance.

ENABLE SEGMENT ROUTING

1. Enable segment routing

    Firstly, configure *segment routing* in global configuration mode. Issue the command *segment-routing mpls* under the *address-family ipv4 unicast* in IS-IS.
2. Configure a prefix-SID for Loopback0

    Prefix-SID can be configured by either using an index value (16000 + 1) or absolute value 16001. *Prefix-sid index 1 / prefix-sid absolute 16001.*
3. Verify the prefix-SID configuration

    Show isis database verbose P1. Prefix-SID can be found as an index value.

CONFIGURE TI-LFA

1. Enable TI-LFA link protection
   TI-LFA can be configured by configuring the IS-IS instance or the interfaces inside the IS-IS instance. For example, *interface g0/0/0/1 > address-family ipv4 unicast > fast-reroute per-prefix > fast-reroute per-prefix ti-lfa*
2. Verify link protection
   show isis ipv4 fast-reroute 10.0.0.2/32 detail. Link protection can also be verified by generating traffic on a link and then shutting the link down.
3. Enable TI-LFA node protection (**Not functional, to be implemented in 6.1.1**)
   Node protection can be configured by adding the following line to already existing TI-LFA configuration: *fast-reroute per-prefix tiebreaker node-protecting index 100*.

CONFIGURE MAPPING SERVER

1. Configure local advertising
   Local advertising can be enabled with the command *segment-routing prefix-sid-map advertise-local* in the IS-IS configuration address-family ipv4 submode.
2. Configure a mapping server
   In global configuration mode: *segment-routing > mapping-server > prefix-sid-map > address-family ipv4 > 20.1.1.1/32 10 range 200.*
3. Verify the mapping server configuration
   Show segment-routing mapping-server prefix-sid-map ipv4 detail.
4. Configure mapping clients
   By default, routers are configured to receive prefix-sid-maps. The functionality can be disabled by going to *router isis SRLAB > address-family ipv4 unicast > segment-routing prefix-sid-map receive.*

CONFIGURE SR-TE

1. Alter the IS-IS configuration
   Remember to type *segment routing* in global configuration mode if you have not done it yet. The following lines of configuration must be added under the address-familly ipv4 unicast in router IS-IS configuration: *mpls traffic-eng level-2-only* (make sure this matches the IS-IS level) > *mpls traffic-eng router-id Loopback0.*
2. Configure MPLS-TE
   In global configuration mode: *mpls traffic-eng > interface g0/0/0/0-3.*
3. Configure explicit path from P1 to P2
   Explicit-path name LABEL_PATH > index 1 next-label 16003 > index 2 next-label 16005 > index 3 next-label 16007 > index 4 next-label 16008 > index 5 next-label 16006 > index 6 next-label 16004 > index 7 next-label 16002.
4. Configure an SR-TE tunnel
   Interface tunnel-te150 > ipv4 unnumbered Loopback0 > autoroute announce > destination 10.0.0.2 > path-selection > segment-routing adjacency protected > path-option 1 explicit name LABEL_PATH segment-routing.
5. Verify the SR-TE tunnel configuration
   Traceroute 10.0.0.2. If the tunnel has been configured correctly, the traceroute should show all the steps configured in the explicit path before finally reaching P2. *Show mpls traffic-eng tunnels*.

CONFIGURE PCC PARAMETERS

1. Delegate tunnel-te150 to PCE
   interface tunnel-te150 > pce > delegation.
2. Configure the PCC
   Mpls traffic-eng > pce > peer source ipv4 192.168.105.1 > peer ipv4 192.168.105.2 > segment-routing > stateful-client > instantiation > reoptimize 60 > auto-tunnel pcc > tunnel-id min 1 max 100.
   In global configuration mode: *ipv4 unnumbered mpls traffic-eng loopback0*
3. Verify PCE peer and tunnel delegation
   Show mpls traffic-eng pce peer, show mpls traffic-eng tunnels

CONFIGURE PCE SERVER (ODL)

1. Install Ubuntu 16.0.4 or similar that can run OpenDaylight Beryllium.
   Wget https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight /integration/distribution-karaf/0.4.4-Beryllium-SR4/distribution-karaf-0.4.4-Beryllium-SR4.tar.gz
2. Install Java
   Apt-get install default-jdk
3. Run OpenDaylight and install PCEP features.
   ./bin/karaf in OpenDaylight folder to run the program. *Feature:install odl-bgpcep-pcep, feature:install odl-restconf-all, feature:install odl-dlux-all*

CONFIGURE PCE SERVER (PYTHON)

1. Install python3, gevent, python-pip
   Sudo apt-get install python3/python-gevent/python-pip
2. Use pip to install protobuf
   Sudo pip install protobuf
3. Clone https://github.com/Dipsingh/mpls-pce
   Git clone https://github.com/Dipsingh/mpls-pce
4. Verify that PCE can be launched
   python ./pce_controller.py in the mpls-pce folder. Some troubleshooting tips in chapter 5.7.1.
5. Configure PCE to impose SR-TE tunnel on the PCC
   Configuration file is in PCE_config.json. Configurable parameters are tunnel name, setup and hold priority, FRR, tunnel source and destination, SR ERO list.
6. Configure a tunnel P1 – P4 – P5 – P8
   Set tunnel source to 10.0.0.1 and tunnel destination to 10.0.0.8.
   Configure SR-ERO list with the required hops, for example *“10.0.0.1” = 16001* for P1.
7. Verify the SR-TE tunnel
   Turn on the PCE again and verify that the PCC gets the tunnel with *show mpls traffic-eng pce peer, show mpls traffic-eng tunnels.*

CONFIGURE A DUAL-PLANE NETWORK

1. Configure loopback1

   10.0.0.100 for P3 and P5, 10.0.0.200 for P4 and P6.
2. Add loopback1 to IS-IS instance
3. Configure anycast-SID

   Configure loopback1 inside the IS-IS instance with the following: *prefix-sid absolute 16100 n-flag-clear*.
4. Make sure two traffic flows never take the same path under normal circumstances.

   Configure cross-links in the core with higher metric values.