ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1161032021.pdf
https://doi.org/10.30534/ijatcse/2021/1171032021

Implementation Based Approach to analyze MPLS and Segment Routing Traffic by Using ODL SDN Controller

Waqas Saeed [1], Davide Adami [2], Sajid Rehman Babar [3], Akbar Ali [4]

[1] Department of Computer Science, Sub Campus Gomal University Tank, Pakistan, waqas.researchers@gmail.com
[2] Department of Computer Science, Federal University of Pisa, Pisa, Italy, d.adami@iet.unipi.it
[3] Department of Computer Science, Sub Campus Gomal University Tank, Pakistan, srehmanbabar@gmail.com
[4] Department of Computer Science, Federal University of Technology, Islamabad, Pakistan, aakbarali18@gmail.com

ABSTRACT

Network infrastructure in the data center is getting more complicated due to heavy network traffic, by increasing a network load leads high storage consumption and high deployment cost.

We have proposed an effective traffic management mechanism by combining Segment Routing and Multipath Transmission Control Protocol TCP in a logically centralized controller environment. Segment Routing and Multipath TCP is used to define a logically centralized and physically distribution for Software Defined Network SDN.

Segment Routing SR technology is used for simplifying Multi-protocol Label Switching MPLS and software-defined network SDN networks. Segment Routing makes it easy to assign a dynamic traffic according to customer's requirements through the high-level application.

This work presents a comprehensive comparison between MPLS and Segment Routing Traffic Engineering, both technologies are implemented in an SDN network environment, based on GNS3 network Emulator as data network Layer, Open-Daylight as SDN Controller, and Python scripts for sending Segment IDs as a part of network Orchestrator in Application Layer.

Key words: Software-Defined Network, Segment Routing, MPLS, Open Daylight, Data Centre Network, Multipath TCP.

1. INTRODUCTION

Segment routing is used to control routing architecture paradigm. Router nodes forward the packet through segment routing policy list called "segments". Segments represent instruction, topological and service based. Segments have local semantic to node and global semantic domain.

Segment associated with different topological and instruction. Local segment and global segment, Local segment in which packet is forward via specific out going interface and Global segment in which packet is forwards and instruct by an SR domain via specific paths to their destination.

In segment routing packet should be processed by container and virtual machine (VM) which is associate with segment. Segments are associated with QOS and bandwidth e.g. (packet is receiver with segments Y MBLS which mean specified bandwidth) [1].

Segment Routing architecture supports different types of control plane:

1.1 Distributed Control Plane

In the case of distribution segments are signalled and allowed by OSPF, IS-IS and BGP and nodes forward the packet according to an SR Policies e.g. (pre-computed local protection) [2].

1.2 Centralized Control Plane

In the case centralized scenario, segments are instantiated by Segment Routing (SR) controller and SR controller will decide when and which node have to forwards the packets according to the source-routed policies. Segment routing (SR) architecture allow the controllers to discover SIDs and sets of local (SRLB) and global (SRGB) labels are available according to different node.

In the case centralized scenario, segments are instantiated by Segment Routing (SR) controller and SR controller will decide when and which node have to forwards the packets according to the source-routed policies. Segment routing (SR) architecture allow the controllers to discover SIDs and sets of local (SRLB) and global (SRGB) labels are available according to different node.

1.3 Hybrid Control Plane

In the case of hybrid scenario, distributed control plane with centralize controller e.g. (when the packet destination is outside of IGP domain then Segment Routing (SR) controller will compute the SR policies on the behalf of IGP domain.

In a modern Data centre services are deployed and distribute in data processing and by increasing

computation resource storage demand are increasing [3]. For the stable communication and computational resources thousands of severs are distribute in a data centre network (DCN)for resulting a network congestion. Rending is primary solution used to satisfy the network throughput simultaneously and for the demand of network transmission.

Different networks topologies have been used to solve the network congestion in data centre network (DCN). Multi rooted topologies are used with the implementation for the Equal Cost Multiple Path (ECMP) algorithm is used for selecting random path, load balancing and for the multiple paths between hosts. Such as, Fat tree [4], VL2 [5] and jelly fish [6] topologies.

During the time when demands and traffic requirement increase, random path selection become a problem because, it creates flows to collide on links and lead a congestion problem in network.

In this case SDN-based Multipath TCP (MPTCP) [7] implementation used instead of Equal Cost Multiple Path (ECMP). Multipath TCP (MPTCP)that split the flows then, SDN technique is used to control the flows in control plane, that control the network status and path allocation individually by links information. Then, forward the rules stored in switch flows table. This mechanism decreases the overload links to reduce congestion problem in network and to improve the network resource utilization.

However, this approach was not successful for a longer time. Because they need an extremely expensive resource, Ternary Content Addressable Memory (TCAM) to store a forwarding rule. (TCAM) is the dominant hardware used by Switches to stores the forwarding rules that offers the line rate parallel lookup and forward the packets at high speed. But it's so expensive and leads to power consumption.

For this solution SDN switches used, which have integrated (TCAM) but with a limited size and we can store 2k-20k forwarding rules in applications [8]. Therefore, in the case of (MPTCP) technique where we are using SDN-based (DCN) that generate more data flows and increase the TCAM resource consumption but limited (TCAM) resources obstacle on SDN granularity affect (DCN) scalability. It's not a good solution because it extremely expensive in the term of deployment cost.

2. Background Traffic Engineering Mechanism

In Segment routing we can also create tunneling according to the customers requirement and needs. We can create tunnels for increasing network preference and throughput by consider to customer SLA. Traffic Engineering is a useful tool for path deterministic and paths avoidance [8]. By using system identification number (SID) adjacencies we can specify a path flow though the network. As you can see in the picture below, use case of traffic engineering.

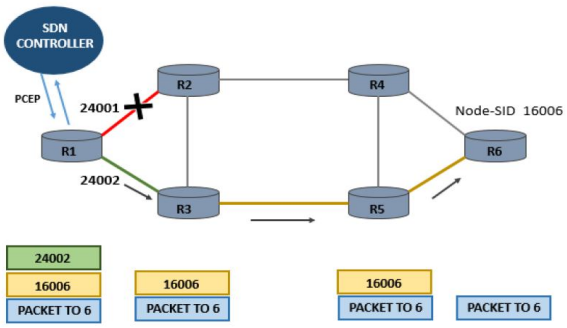

Figure 1: Traffic Engineering Classification of Segment Routing.

Lets spouse we want to send a traffic to R6 (Node-SID 16006). To push the node segment on the top of packet, it will forward the traffic according to the shortest path. In addition, Node-SID present the instruction for Equal-Cost Multi-Path (ECMP) to select the shortest path to reach R6 node.

We have two paths to forward the traffic by using (R1, R2, R4, R6 **OR** R1, R3, R5, R6) but if we have situation when in the case of (R1, R2, R4, R6 OR R1) path is overloaded because of others traffic and quality of service (QOS) drop down the performance.

Then controller have possibility to send (push) the traffic dynamically on (R1, R3, R5, R6) instead of (R1, R2, R4, R6 OR R1) for avoiding busy links and finally reached to R6 node according to the shortest path.

But in the case of (Anycast-SID) we can define a different group on router or the flow of traffic to reach the destination. Service provider is a tool by using it we can express the macro policies such as "go via plane one of dual plane network" OR "go via European region" [9]. As you can see the below picture.

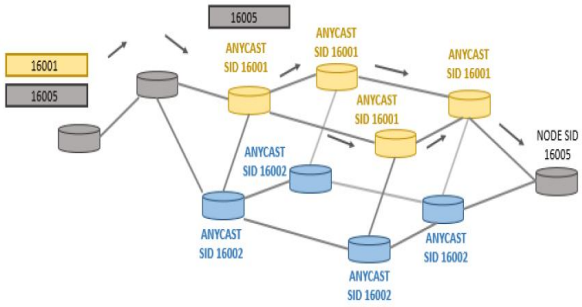

Figure 2: Classification of Traffic Engineering Dual Plane

In this network we can describe dual plane. First one (Yellow) and second is (Blue), traffic flow will continue through the ("yellow" or "blue") node to reach their destination by assigning labels (16001, 16005 **OR** 16002, 16005) as you can see in the picture.
In additional, Equal-Cost Multi-Path (ECMP) is support within a plain, which mean if there is a disjoint path in

the network, load will be balanced per-flow but for this Anycast-SID must have to be configured (One-Per-Network-Plane).

2.1 Fast Reroute Transmission

Segment routing (SR) is a service that support tight Service Level Agreement (SLA). So, we can use Topology Independent Loop-free Alternate (TI-LFA) within a segment routing (SR) network.

The time of convergence from Topology Independent Loop-free Alternate (TI_LFA) in any interior gateway protocol (IGP) in 50 msec. (TI-LFA) is easy to implement which is automatically pre-computed by interior gateway protocol (IGP) for the protection of paths. (TI-LFA) use post-convergence paths to support traffic rerouting in the case of primary path failure or path failure [10]. In the case of path failure in segment routing (SR) network, it reroutes path again and send the packet by attaching backup segments.

2.2 Multipath Transmission Control Protocol

(MPTCP) is progression of Transmission Control Protocol (TCP) [11]. By using TCP, we can effectively use different multiple paths within a single transport connection network [12], [13].

MPTCP split the incoming traffic flow. So, (the single flow propagates by multiples paths) simultaneously between the host and destination. In the case of failure, we can use alternate paths to send the traffic. In additional MPTCP is used to transport approaches in Data Center Network (DCN) [14], increase the throughput of packets during the transmission and to improve the performance of network [15] specially when data center is suffering from the huge traffic congestion problems [16] because of the lacks and effective utilization of the network such as, storages and computing speed across the multiple servers. Various data center network (DCN) topologies MPTCP is used because its provide a better load balancing and better traffic flow strategies when the network is suffering from congestion [17], [18], [19].

3. SOFTWARE DEFINED NETWORKS

Software-Define-Network (SDN) is used for routing & policy networking control functions definition in automated way for ensuring network reachability on Time. SDN provides a new level of programmability and abstraction to Network layer which is playing phenomenal role in Automated networks.

For this purpose, we need some information such as: IP address allocation, Bandwidth of different path Allocation, defining Policies, Routing changes for End-to-End Reachability.

These things are not gone be happened automatic way and for performing charges on every router and switches it will take a lot of time maybe (one week or few weeks for enabling end-to-end reachability for configuring a new path on required links)

3.1 SDN Framework and architecture

Software Define Network (SDN) architecture is an open-flow protocol for entire topology and provides southbound interface (open-flow) for the communicate with the data plane, for his we can use different devices such as: Juniper, Cisco, HP etc. OpenFlow works on standard rest API defined between control plane and Data plane or forwarding plane. OpenFlow is a (Brain or Controller) use for manipulating and change in the Routing-Tables (RT) and Routing-Algorithm (RA) is use by forwarding-plane [20].

it allows SDN-Controller as a remote configuration to forward the packets by modifying or adding a new path and allow to removing matching rules and actions (which is applied by routers).

Open-Flow (OF) needs supported by (Controller and Forwarding routers) used within a network. For example, Switches and Routers can receive instructions from 3rd Party which is basically Controller and behave as Data Forwarding plane according to controller.

Business applications by the help of rest APIs, provide a multiple functionality in the network according to the customer requirements for start their business.

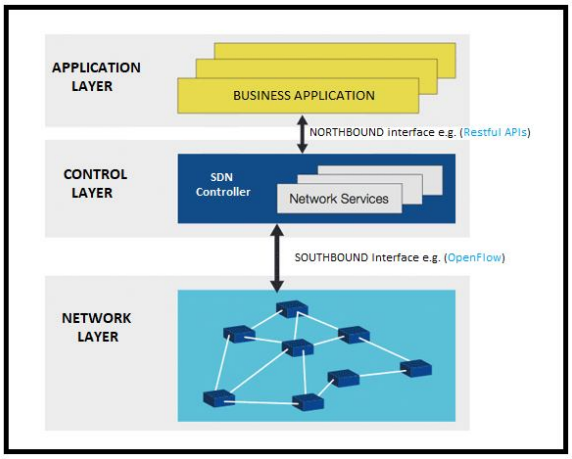

Figure 3: Architecture of SDN

Software-define-network (SDN) controller is behaving as a “brain” of the network and provide a centralize control to manages the flow of network devices and applications.

SDN architecture provide a (Forwarding plane or data plane) layer. Data plane layer is consisting of (switches and routers) devices (which are connected to each other by wired or WIFI connection). Network devices perform a forwarding operation and behave according to the controller instruction.

Communication between Software-define-network (SDN) controller and programmable devices (switches and routers) are enabled by southbound interface.

Southbound APIs, for example, SDN controller can remove and modify the entry of forwarding-table (FA) through (southbound interface).

The northbound interface (restful API) presents a network application that place on the top of SDN

controller, which basically enables network programmability through the application level. Northbound restful API is a critical part of SDN controller architecture, where the Northbound restful APIs connect SDN controller for automation and orchestration of the network platforms.

Application layer is a set of applications that provide a functionality by northbound API to implement or execute an operation from application level by monitoring a physical network such as: routing, load balancers, firewalls etc. Commands receive from the application layer are forward to the (southbound) instructions.

4. IMPLEMENTATION METHODOLOGY OVERVIEW

The implementation of the paper depends on three tools.

1) Open-Daylight is used for the SDN controller.
2) GNS3 Emulator used for “topology”.
3) Postman used as a high-level application.

4.1 SDN Controller ODL

We downloaded and installed the ODL Boron-SR2 SDN controller on Linux Ubuntu; Java-development-kit must be installed on the system. We have used the version of "JDK-8.1" on Ubuntu operating system.

4.2 GNS3 Emulator

Cisco Ios XRv image is used as routers import inside the GNS3 GUI virtual machine. GNS3 offers an appliance for importing Cisco IOS-XRv images, provided by VIRL. First, we Download the Cisco IOS image with the extension of “. ova” then convert it into a “.qcow2” extension and import it inside the GNS3 as a Qemu Virtual Machine.

4.3 Postman Application

Postman application is a necessary program to install on the Linux operating system as REST API. Open- Daylight use RESTful API northbound interface to communicate with upper layers. REST API Client is available online free as a RESTful web service for clients to access. REST API Client supports HTTP Request methods represented in JSON and XML format by specifying the raw and binary request body. Postman is an easy-to-use interface for sending HTTP requests by using different Operations, GET, POST, PUT and DELETE. In this work, Postman is used for retrieving information and for adding, update, delete multiple paths via selecting Get, Post, Delete for MPLS, and Segment Routing tunnels.

5. OPEN-DAY-LIGHT ODL

ODL is an open-source project by Linux Foundation [21]. SDN-Controllers is specially used for enabling the network and testing network virtualization. The open-source architecture consists of different modules to perform for pluggable network functions. Open source is an application to developing and customization projects available in the market.

Open-Daylight was introduced On April 08, 2013, by Linux Foundation to accelerate OpenFlow and Software Define Network development. Open-Daylight depends on Java programing language [22]. The founding for Open-Daylight projects was contributed by different companies, Big Switch Networks, Brocade, Cisco, Citrix, Ericsson, HP, IBM, Microsoft, Juniper Networks, NEC, Red Hat, and VMware [23].

There are different releases available in the figure below.

Release Name	Release Date
Hydrogen	February 2014
Helium	October 2014
Lithium	June 2015
Beryllium	February 2016
Boron	November 2016
Carbon	June 2017
Nitrogen	September 2017
Oxygen	March 2018
Fluorine	August 2018
Neon	March 2019
Sodium	September 2019

Figure 4: ODL Release Information

ODL technology support OpenFlow. The first project, Hydrogen, was released in February 2014 [23]. Implementation of the source code initially contributed by Big Switch Network, Cisco, and NEC.

5.1 ODL Architecture:

Open-Daylight architecture is Simplified. See figure 5 below [43].

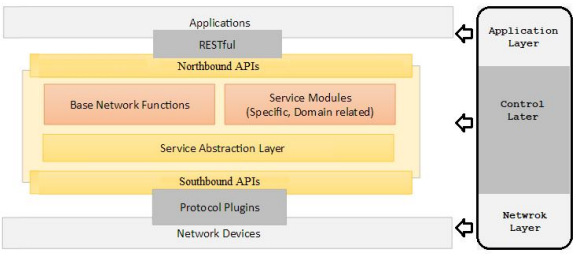

Figure 5: Architecture of Opendaylight

SDN-Controllers consists of three main parts

1. Southbound APIs
2. Control function layer
3. Northbound APIs

5.2 Southbound Interface

The southbound interface supports multiple protocols,

OpenFlow, BGP, LISP, SNMP, PCEP, etc. all modules are dynamically linked to a Service-Abstraction-Layer, which determines about the fulfil service requested by different applications according to requirements and protocol used between the controller and the network devices.

5.3 Control Layer

The main components of ODL are Layer and pluggable modules functions.

SAL is used to represents a bundle key between consumer and service producers. SAL provides registry services through APIs requests. When consumer call for entail participation through API service, SAL binds it into a "contract" consisting of two architecture.

The first one is Application-driven SAL, and the second is module-drove SAL.

We discussed before the features of open-source a pluggable-module that allows a specific function. Some preconfigured network functions come with ODL are:

1) Topology functions
   Represents the function of "Topology" for displaying a service for the network by subscribing to a network-link.

2) Statistics services
   Managing the information about the topology, nodes, flows, and queues, etc.

3) Forwarding services
   Managing the information about network flows, forwarding the principle, and SDN-Controller functionality.

5.4 Northbound Interface

Northbound interface, allows the component to communicate with a higher-level component, using the southbound interface REST APIs. REST APIs are used to runs HTML base request through the application. REST API is technologies based on HTML, JSON, and XML format to forward the request. Top-level application Interaction can be through HTTP request operations by using POST, PUT, GET, and DELETE. Transmission of the Data via REST API by run algorithms and forward the request to ODL-Controller to create new rules by specifying the BGP speaker inside the Network [23].

6. SETUP FOR VIRTUAL NETWORK MACHINE

A virtual Network is made by using virtual links to represent a non-physical connection between network devices.

Two types of virtual networks

1) First, One is based on a virtual Network via VLAN and VPN.
2) second is predicated on network virtualization by using virtual devices, VMware, and Hypervisor [24].

In the experiment, Virtual Network is made of Cisco IOS XRv virtual router image run inside the software of virtual machines placed on the identical Hypervisor.

6.1 Virtual Machine VMware

VMware is a desktop virtualization application software that runs as guests on operating systems at an identical time. The program is friendly and relatively easy to use. Using VMware several operating systems, use as a guest, like ubuntu, centos, fedora, Linux, cloud server, etc. [25]. VMware allows portability of the different guest operating systems. Where we can easily take backup or make a clone copy from the already installed guest operating system running on the virtual machines, and easily move to another location. By this, we can save us a lot of time instead of setting up a new virtual machine.

6.2 Features of VMware Network

VMware provides possibilities by choosing a different option to configure a virtual machine networking [26]. Allow us to use Network adapters inside the various virtual machine by selecting the following modes:

1) Bridged-Mode.
   Allow accessing virtual machines through the Internet. Ethernet adapter of the hosts is a best way of accessing Virtual Machine through internet service.

2) Network Address Translation NAT mode.
   The host machine and VMware machine shared a single IP and MAC address in the network.

3) Host-only adapter.
   The host machine and virtual machines adapter are connected to a public network. By using this approach, we can create an isolated virtual network environment.

4) Custom networking.
   Allow configuring of the connection of the network manually. It allows us to create external networks or can run only on the host computer.

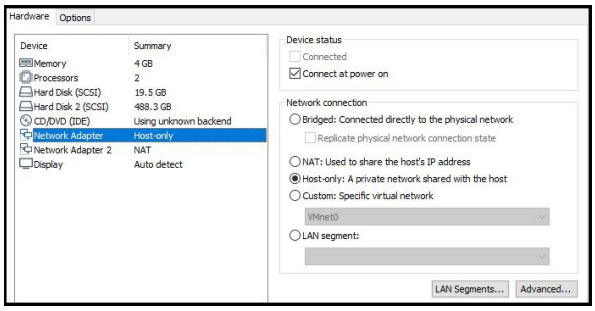

Figure 6: VMware Network Setup.

7. CISCO IOS ROUTER IMAGE XRV

Cisco Router IOS is available on the cisco website with version 32 and 64-bit, for implementing the cisco image, we can use VMware on QNX microkernel to allow QNX users to send messages and allow to communicate with microkernel remotely [27]. XRV-VM contains a Route Processor functionality to communicate with the control plane, in figure 4.8 below. Cisco IOS operating system provides flexibility to configure without having a cisco IOS hardware.

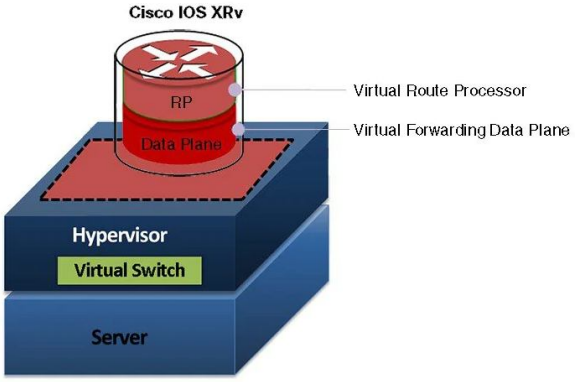

Figure 7: Cisco Router.

8. ENVIRONMENTAL SETUP

8.1 Network-Topology Build-Up

Network topology is built using GNS3, and Routers is configure using CLI. Each router has a management interface that would be used to communicate with SDN-Controller.

For connecting the router management interfaces in GNS3 to the Linux host and ODL, GNS3 offers the possibility to choose a cloud node. Then selecting the defined bridge interface card on the host, one the other side, connecting the cloud to a switch appliance available in GNS3, we can make the gateway of the whole Network in GNS3 to the external network. Now we should connect the management interface of the routers to the switch inside GNS3. See figure 8.

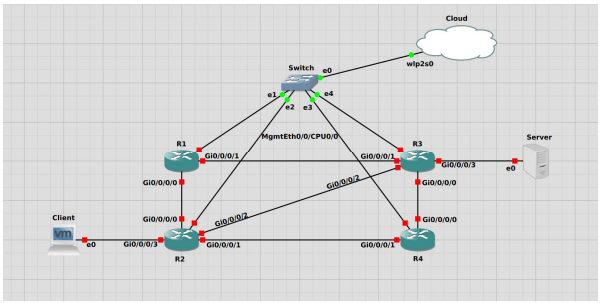

Figure 8: Topology of GNS3

Cisco ios XRv routers support the netconf YANG data model, Netconf protocol, ISIS and IGP routing protocol, BGP, PCEP, and Segment routing. Firstly, we configure the management interfaces address and add protocols (IS-IS, MPLS, BGP, and PCEP) configuration. Finally, we add extra configuration for the ODL modules, BGP and PCEP to allow communication with BGP speakers for each router.

We assigned the IP address of Management interface, Ethernet gigabyte interface, Loopback interface, and Node SIDs. You may be able to find the network configuration below in table 1.

Table 1: Configuration of Routers IP address

Router ID	Loopback Address	Management interface address	Gigabyte Ethernet interfaces		Node SID
R1	10.10.1.1	192.168.100.1	G/0 G/1	10.10.12.1 10.10.13.1	17000
R2	10.10.2.2	192.168.100.2	G/0 G/1 G/2 G/4	10.10.12.2 10.10.14.1 10.10.15.1 10.10.100.1	18000
R3	10.10.3.3	192.168.100.3	G/0 G/1 G/2 G/3	10.10.16.1 10.10.13.2 10.10.15.2 10.10.200.1	19000
R4	10.10.4.4	192.168.100.4	G/0 G/1	10.10.16.2 10.10.14.2	20000

8.2 Setup for Opendaylight ODL

The first thing is to download the ODL from the ODL official website [48]. We used the Boron-SR2 distribution version. The rationale for selecting this release made for SWAN Network Orchestrator, which already contains many applications compatibility. By Installation of ODL, we can customize the environment. After installation, we can add some features using the following commands mention below. For the Installation of ODL, we followed the guide [28].

• Setup for IP address of ODL, we assign the address **192.168.100.100** through the terminal by using following command, see figure 10.
" sudo nano /etc/network/interfaces "

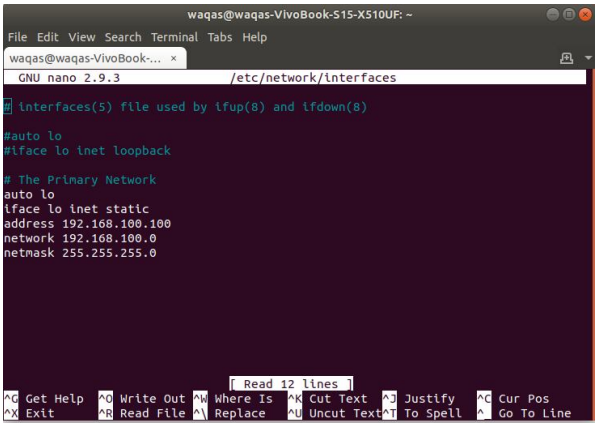

Figure 10: IP address setup for ODL

Commands for Running the ODL SDN-controller

• To run the karaf container, first enter in the directory of ODL use the following command.

"cd distribution-karaf-0.5.2-Boron-SR2/"
Then use the command.
"./bin/karaf"

• Add all necessary features by typing the following command below [29]. See figure 11.

" feature:install odl- bgpcep-pcep-all install, odl-bgpcep-bgp-all, odl-restconf-noauth install odl-dlux-all "

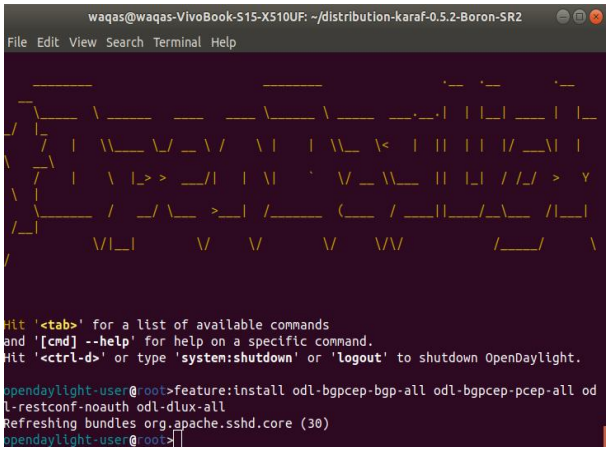

Figure 11: IP Address Setup for ODL

• After installing features, different files are created with an XML extension, you can see in the following directory [30]. See figure 12.

" distribution-karaf-0.5.2-Boron-SR2 /etc/opendaylight/karaf "

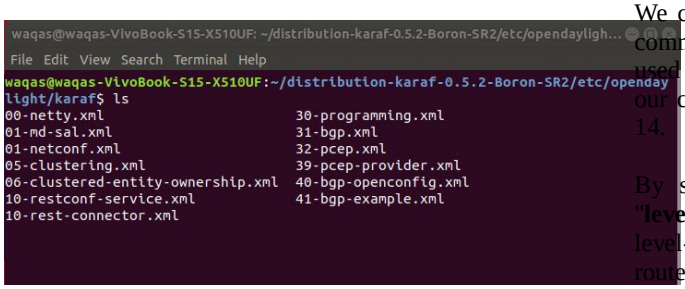

Figure 12: Directory of ODL

8.3 Web-Based REST-API client Postman:
ODL uses a RESTful API northbound interface to communicate with upper layers. REST API Client is available online free as a RESTful web service. REST API supports HTTP Request methods represented in JSON and XML format by specifying the raw and binary request body.
Postman is an easy-to-use interface for sending HTTP requests by using different Operations, GET, POST, PUT and DELETE. In this work, we used Postman for retrieving information and to add, update, delete multiple paths via selecting Get, Post, Delete for MPLS, and Segment Routing tunnels.

We used the Postman application within the network by posting an XML or JSON format request to the SDN-Controller through the Postman application interface. The creation of a tunnel requests is presented within the subsequent Paragraphs below.

8.4 Configuration of Cisco IOS Router
Firstly, we have done the Configuration of Loopback address interfaces, Gigabit Ethernet interfaces, and Management interface on each router. For communication with the SDN-Controller, the Management interface is used. See figure 13.

Interfaces Configuration
Loopback interface:

```
interface Loopback0
 ipv4 address 10.10.2.2 255.255.255.255
!
```

Management interface:

```
interface MgmtEth0/0/CPU0/0
 ipv4 address 192.168.100.2 255.255.255.0
!
```

Gigabit Ethernet interfaces:

```
interface GigabitEthernet0/0/0/0
 ipv4 address 10.10.12.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1
 ipv4 address 10.10.14.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
 ipv4 address 10.10.15.1 255.255.255.0
!
interface GigabitEthernet0/0/0/3
 ipv4 address 10.10.100.1 255.255.255.0
!
```

Figure 13: Interfaces Configuration of Routers

8.5 ISIS Configuration
We configuration IS-IS instance, which is an important command for starting the routing process. An area tag is used to identify the instance was assigned. However, in our case, we used area 10 to clarify in the figure below 14.

By specifying the command "**is-type**" configured as "**level-2-only**", meaning that all routers are configured as level-2 to communicate with each other on level 2. All router in the domain is configured as level 2. The next command, "**net**" is used for the Routing Process. It is an important configuration, in the case if you are buildup "**multi-area-domain**".

By using command **"log-adjacency**" to keep updated to all other router if the neighbor goes up or down. It will enable log-adjacency-changes.

Inside the address family, by specifying unicast ipv4 "**address-family-ipv4**", first we configured "**fast-reroute**" meaning provide a fast reroute functionalities in the case of link failure. By using **"metric-style-wide**", meaning that IS can retrieve new style TLV "**new-style-TLVs**" always. Then incremental shortest path first, ISPF is used to recompute a path if any changes occur in the network.

To enable MPLS Traffic Engineering, we configured "**IS-IS-level**" to get link information through ISIS level. In our case, we configured to distribute the MPLS links on "**level-2-only**" only, and We are specifying the Loopback address as a (Router ID). Then we also specify the preference of the Segment Routing label over the LDP.

Finally, by configuring the interfaces that are advertised by "IS-IS". First, we configure the loopback interface advertisement on "passive mode", which means We just want to announce in the network. We do not want any traffic on the loopback interface. Here We also specify the node-SID "Prefix-SID" of each Router that will bind to the Loopback-address of routers. Then other GigabitEthernet interfaces configure as a point-to-point link, and family address unicast between the router.

Figure 14: ISIS Configuration

8.6 MPLS, Segment Routing, and PCEP Configuration

We entered the MPLS configuration mode by typing "Mpls oam" for the MPLS Operation and Management. Firstly, we enabled the interfaces for the MPLS traffic flow as you can see, the figure below 15.

Then by entering in "PCE" mode configuration, we specify the IP addresses of the source (which is the management interface of router R2) and define the IP address of SDN-Controller, which is (192.168.100.100) as a peer to communicate within a network.

We configured the Segment Routing mode for path computation. Then, specify the "stateful-client" for adding a new session using the stateful capabilities of TLV. Moreover, we configured the "delegation" of all "active" tunnels to PCE. It allows SDN-controller to change the LSP while computing a new path that can reoptimize tunnel traffic.

Then, we specify the Speaker ID, as the Loopback-address of the Router, for defining the range of tunnels, stateful PCE used to specify the minimum and maximum range of tunnels IDs. Finally, specify "reoptimization" which means the new labels will be used after the tunnel re-optimization according to the specified time in seconds.

Figure 15: MPLS, SR and PCEP Configuration

The present configuration is valid for all Routers in th e network. Just IP addresses should be changed for ea ch router. Given In table 5.8 above, you can see the I P address of all the routers.

8.7 BGP Speaker Configuration

In the BGP configuration, we select R2 router as a BGP speaker to redistribute all IGP information to ODL. Firstly, BGP instance should be open with a specifying area tag 10. Then, we specified the router loopback address as a router ID. BGP neighbor is SDN-Controller, the management interface is used to send the update

about all link-state information to the SDN-Controller. BGP configuration, you can see figure 16.

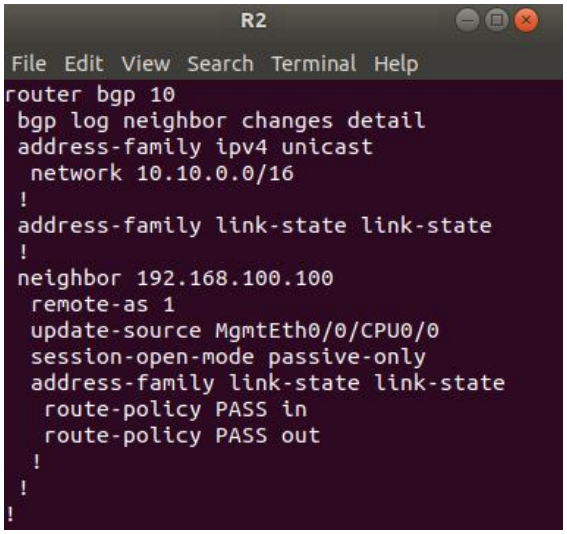

Figure 16: BGP Speaker Configuration

8.8 Setup for BGP Speaker in ODL

In ODL, BGP module is placed in " distribution-karaf-0.5.2-Boron-SR2/etc/opendaylight/karaf" in file (41bgp-example.xml). it should be reconfigured as you can see in figure 17 below by specifying the IP Address of the management interface of Router R2.

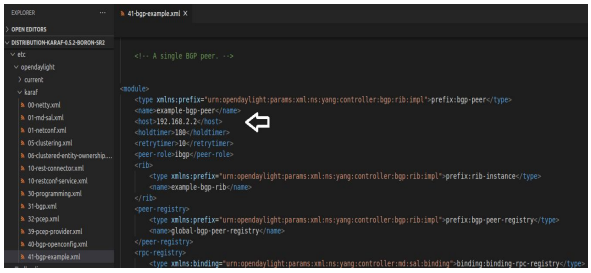

Figure 17: BGP Speaker IP address

9. EXPERIMENT FOR TUNNEL SETUP

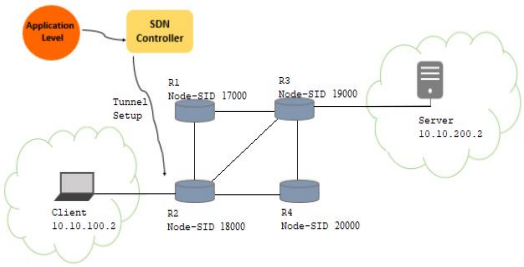

Figure 18: Topology and SDN Controller View

Now, we are going to create a tunnel by using the Postman application as a RESTful web API service. ODL uses a RESTful API northbound interface to communicate with upper layers. It will receive the request from the application and forward it to the BGP speaker R2 router.

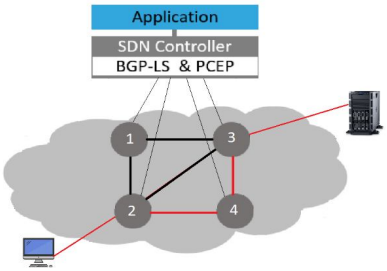

Figure 19: Tunnel

We add one tunnel from the router R2 as a source to router R3 as a destination by forcing the tunnel to pass through R4 via a specified route path in a packet header. Segment Routing traffic separates the flow into many segments, and the packet transmission is accomplished by a segment label list, which is stored in the packet header. SR technology reduces the number of forwarding rules and solves the TCAM utilization problems effectively.

We have implemented three traffic engineering approaches.

1) Traffic-Engineering design with MPLS.
2) Traffic-Engineering design with Segment Routing.
3) Traffic Engineering design by Specify Metrics on router interfaces.

9.1 MPLS Traffic-Engineering Tunnel Setup by using Postman.

To establish an MPLS tunnel, we send a REST API command to the PCEP module inside ODL to add LSP. Inside postman application by defining parameters for connection to ODL and sending the right data model in XML or JSON format. The PCEP tunnel creation is shown in the figure below:

Connection Parameters should be defined as follow:

• **Authorization**: username and password of ODL, which is “admin” by default.

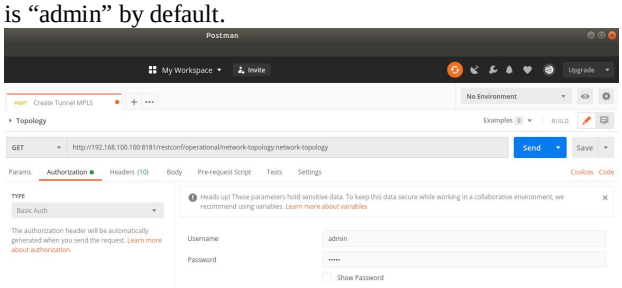

Figure 20: Set Username and Password “admin”.

• **Header**> content type: application/xml
• **Header**>accept application/xml

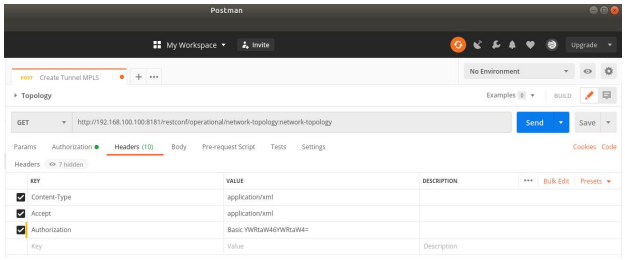

Figure 21: Set Content Type and Accept to “xml”

Then we should go to the body part to specify our code and select "POST" action. This action should send the request to the ODL module: see figure 22.
“http://192.168.100.100:8181/ restconf / operations / network -topology-pcep: add-lsp”
Then we will fill out the body part of our message. For doing this, define ingress router, egress router, and the ERO Explicit route objects meaning, nodes passed through the LSP. For adding an LSP tunnel from R2 to R3 router, explicitly pass through R4 (in line 20).

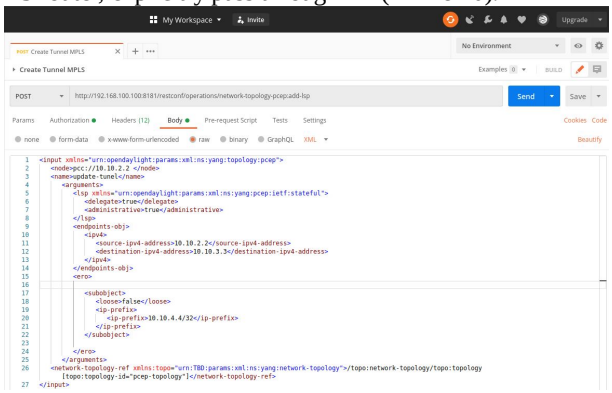

Figure 22: Setting Up a Body Part for MPLS.

Line 2 defines the IP address of the PCEP node, in which ODL will enforce relevant configuration. In line 6, delegates the Control of LSP to the PCE, ODL, so this tunnel cannot be deleted or modified by CLI on the router.
To check the tunnel creation, we can check by inserting the command on the router, see figure 23:
"show mpls traffic-engineering tunnels brief",

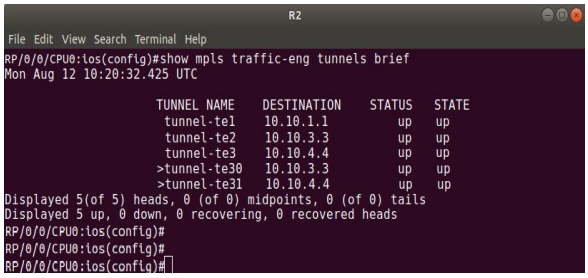

Figure 23: Details About Tunnels Created.

It shows that "**tunnel-te30**" is created from R2 to R3 through the Explicit route, defined in ERO. The character ">" before the name indicates that the creation of this channel is made outside the router enforced by ODL. The reason for having "**30**" as tunnel ID is that we already configured by "auto tunnel PCC" command to have tunnel numbers in the range of 30 to 60.

9.2 Segment Routing-TE Tunnel Setup by using Postman.

For adding a new Segment Routing tunnel, which is similar to MPLS tunnel creation, by using Postman with connection parameters, we will send the configuration to the same PCEP module of ODL. The difference is that, for adding SR tunnels, we need to define the route by specifying the SID numbers.
We added one more SR tunnel from R2 as a source to R3 as a destination, by forcing the tunnel to pass through R4 explicitly. See figure 22.

In line 16, the value should be set to "1" to specify this LSP as an SR-TE LSP (In line 18 and 26), the SR-ERO sub-object is present with SID. See figure 5.23.
We can see the created tunnel by inserting the command "show mpls traffic-engineering tunnels segment-routing brief" on the router. See figure 24.

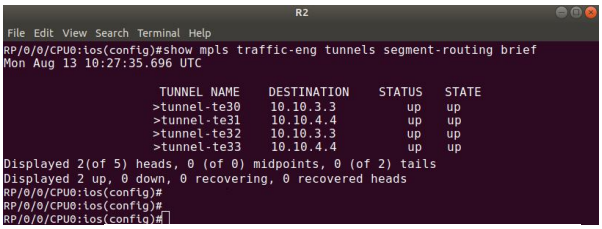

Figure 24: Network Try-out

The network Reachability, by pinging from R2 to R3 router. See figure 25 and 26 below.

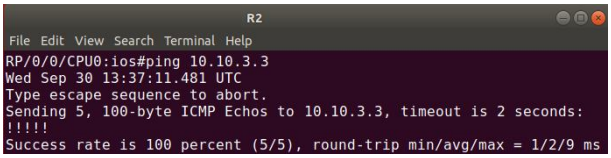

Figure 25: Pinging for Check Connection.

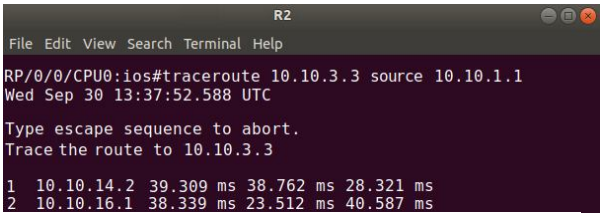

Figure 26: Traceroute of Tunnel.

Traffic is Forward through the router R4 because, in the ERO section, we specify the SID label of route R4. Send traffic to Router R3 destination, by forcing the tunnel to pass through R4 explicitly as you can see the figure 22 above.

9.3 Traffic Engineering approach by Specify Metrics on router interfaces.

The proposed solution can also improve by defining metrics on router interfaces and metrics used by a router to make routing decisions. A metric is typically one of many fields in a routing table. Router metrics help the router to choose the best route among multiple feasible routes to a destination. The route will go in the direction with the lowest metric. A router metric is typically based on information, path length, bandwidth, path cost, delay, and communications cost. See figure 27 below.

Figure 27: By Specify Metric on Router Interfaces.

Routing decisions are based on metrics to forward the traffic through the lowest metric "1" configured on the router interface. The router metric is used by a router to make routing decisions.
The route through Router R1 has a metric value of 1, so it is the primary route. The route through Router R4 has a higher metric value of 2, so it is the backup route. If the server stops responding, connections failover to the route through Router R1. Each interface reaches to the destination through different routers.

Network try-out
The network Reachability, by pinging from R2 to R3 router. See figure 28 and 29 below.

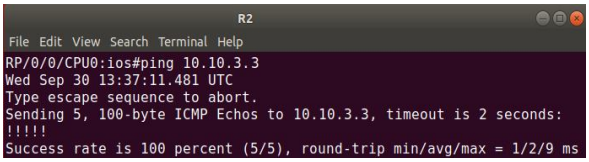

Figure 28: Pinging

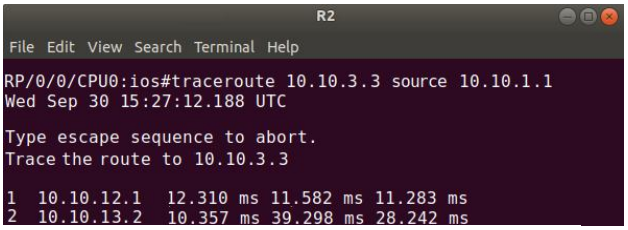

Figure 29: Traceroute.

Traffic is Forward through the R1 router because decision according to the router and the router forwards the traffic according to the lowest metric "1" configure on the router interfaces.

Comparison between MPLS and Segment Routing Tunnels

One of the elemental premises of fragment directing is to dispose of the requirement for extra flagging and name dispersion conventions, such as LDP and RSVP-TE, and use the steering convention itself (IGP, BGP) for name conveyance. Let us take a see at how this compares with LDP and RSVP-TE.

Comparison with LDP

Segment Routing dissemination within the IGP and LDP label distribution is comparable within the sense that they are both "plug and play". Consequently, they are simple to design, names are consequently promoted among the routers when a contiguousness peering is shaped and there's no name exchanged ways (LSPs) to physically arrange. Furthermore, both LDP and SR shape stateless MP2P LSPs determined naturally for each hub. It is this effortlessness in sending that made LDP well known and is one reason portion directing is being embraced so promptly.

Comparison with RSVP-TE

Traffic Engineering and Fast re-route (FRR) are the most reasons that RSVP-TE is broadly conveyed in systems nowadays. It is the made strides activity building versatility and adaptability given by portion steering whereas tending to a few of the modern SDN prerequisites that are driving its appropriation. Activity Building with RSVP-TE empowers the taking after:

Computation of ways utilizing limitations such as transmission capacity, shared interface hazard bunches and flagging of such express ways, which don't essentially get to take after the IGP most brief path.

Bandwidth reservation on the computed way and following of accessible connect transfer speed, permitting for moved forward utilization on the long-haul links.

Fast link connection and hub assurance on disappointment utilizing switchover to pre-computed reinforcement ways utilizing FRR capabilities.

10. CONCLUSION

The purpose of this work is to analyze the source Routing-Paradigm called Segment Routing traffic engineering. Initially, I explain the overall overview of the underlying technologies of Segment Routing and its benefits in the term of SDN-controller.

Substernal benefits of the Segment Routing could be achieved by orchestrated through high-level applications with the assistance of an SDN-Controller. Segment Routing can easily improve the performance of the existing infrastructure network.

We discuss the implementation and how available load distributed among available resources by creating tunnels via specifying route in the packet header. Moreover, Segment Routing reduces the load on the network and reduces the forwarding table strategy, leading to simplified hardware resources and reduced processing time.

Further, it allows the network providers to add dynamical traffic according to the customer requirements could be a great advantage to save time.

Finally, SDN-Controller and Segment Routing technology improve services to the ultimate users in certain Networks. By using SDN-Controller, we can easily optimization the network. Segment Routing brings an advanced traffic engineering logic that is more elegant and efficient in an SDN environment.

SR traffic engineering separates the flow into many segments, and the packet transmission is accomplished by a segment label list, which is stored in the packet header. SR technology reduces the number of forwarding rules and solves the TCAM utilization problem effectively.

10.1. Future Work

Creation of a Graphical application as a network orchestrator in which we can use these SIDs information to maintain segment routing tunnels. ODL Controller is one of the popular solutions to improve some lack of functionalities.

We aimed to analyses the benefits of Segment Routing brings, however, due to some technical limitation. One can continue the work on Fast Rerouting. According to the standard, Segment Routing should provide FRR inherently by using the post-convergence path as a secondary path. At this moment, ODL does not support protected tunnel set up dynamically. The tunnels pushed from ODL are not protected, and option cannot be changed from the router CLI since all the tunnels are delegated to the controller.

Lastly, work can be used to integrate Networks by setting up the segment routing tunnels through GUI. We must specify all the necessary parameters and requirements.

REFERENCES

1. David P 2018, Introduction to Segment Routing, VIP Perspectives, viewed **5 august 2019,** < https://learningnetwork.cisco.com/blogs/vip-perspectives/2018/03/23/introduction-to-segment routing >
2. A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, TCP Extensions for Multipath Operation With Multiple Addresses, RFC 6824, IETF, 2013.
3. Y. Lu, ``SED: An SDN-based explicit-deadline-aware TCP for cloud data center networks,'' Tsinghua Sci. Technol., vol. 21, no. 5, pp. 491_499, Oct. 2016.
4. M. Al-Fares, A. Loukissas, and A. Vahdat, ``A scalable, commodity data center network architecture,'' SIGCOMM, vol. 38, no. 4, pp. 63_74, 2008.
5. A. Greenberg et al., ``VL2: A scalable and _exible data center network,'' in Proc. ACM SIGCOMM Conf. Data Commun., 2009, pp. 51_62.
6. A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers randomly. In NSDI, 2012
7. B. Stephens, A. Cox, W. Felter, C. Dixon, and J. Carter, ``PAST: Scalable ethernet for data centers,'' in Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT), New York, NY, USA, 2012, pp. 49_60.
8. C. Filsfils, S. Previdi et al. "Segment Routing interoperability with LDP", IETF draft-ietf-spring-segment-routing-ldp-interop-00, October 2015
9. C. Filsfils, P. Francois, "Segment Routing Use Cases", IETF draft-filsfilsrtgwg- segment-routing-use-cases-02, October 2013
10. P. Sarkar, H. Gredler et al. "Anycast Segments in MPLS based Segment Routing", IETF draft-psarkar-spring-mpls-anycast-segments-01, October 2015
11. Pierre Francois, Clarence Filsfils et al."Topology Independent Fast Reroute using Segment Routing", IETF draft-francois-rtgwg-segment-routingti- lfa-00, August 2015
12. Coudron, Matthieu, et al. "Crosslayer cooperation to boost multipath TCP performance in cloud networks", 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet), 16 January 2014,

DOI. 10.1109/CloudNet.2013.6710558, ISBN 978-1-4799-0568-3

13. M. Sandri, A. Silva, L. A. Rocha, and F. L. Verdi, “On the Benefits of Using Multipath TCP and OpenFlow in Shared Bottlenecks,” 30 Apr 2015, IEEE 29th International Conference on Advanced Information Networking and Applications, DOI. 10.1109/AINA.2015.159, ISBN 978- 1-4799-7905-9, ISSN 1550-445X
14. Long Li, Nongda Hu, Ke Liu, "AMTCP: an adaptive multi-path transmission control protocol", 12th International Conference on Computing Frontiers, ACM, Ischia, Italy 2015, DOI. 10.1145/2742854.2742871
15. G. Detal, C. Paasch, S. van der Linden, P. Mrindol, G. Avoine, and O. Bonaventure, “Revisiting Flow-Based Load Balancing: Stateless Path Selection in Data Center Networks,” Computer Networks, vol. 57, no. 5, pp. 1204–1216, April 2013
16. Paasch Christoph, Ramin Khalili, and Olivier Bonaventure. "On the benefits of applying experimental design to improve multipath TCP", CoNEXT '13 Proceedings of the ninth ACM conference on Emerging networking experiments and technologies, DOI. 10.1145/2535372.2535403, ISBN 978-1-4503-2101-3, pp. 393-398
17. G. Detal, C. Paasch, S. van der Linden, P. Mrindol, G. Avoine, and O. Bonaventure, “Revisiting Flow-Based Load Balancing: Stateless Path Selection in Data Center Networks,” Computer Networks, vol. 57, no. 5, pp. 1204–1216, April 2013
18. Simon Jounet, Colin Perkins, Dimitrios Pezaros, “OTCP: SDN Managed Congestion Control for Data Center Networks” IEEE, Network Operations and Management Symposium, Istanbul, Turkey, 25-29 Apr 2016, pp. 171-179. ISBN 9781509002238, DOI: 10.1109/NOMS.2016.7502810
19. Savvas Zannettou, Michael Sirivianos, Fragkiskos Papadopoulos, “Exploiting Path Diversity in Data centers Using MPTCP-aware SDN”, 18 Aug 2016, IEEE Symposium on Computers and Communication (ISCC), DOI.10.1109/ISCC.2016.7543794, ISBN 978-1-5090-0679-3
20. Yifei Lu, Shuhong Zhu, “SDN-based TCP Congestion Control in Data Center Networks”, 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), 18 February 2016, DOI.
21. https://www.sdxcentral.com/resources/sdn/sdn-controllers/opendaylightcontroller/
22. Khattak, Zuhran Khan, Muhammad Awais, and Adnan Iqbal. "Performance evaluation of OpenDaylight SDN controller." Parallel and Distributed Systems (ICPADS), 2014 20th IEEE International Conference on. IEEE, 2014
23. "Industry Leaders Collaborate on OpenDaylight Project, Donate Key Technologies to Accelerate Software-Defined Networking"*(Press release). April 8, 2013*. Retrieved November 18, 2013.
24. Hopps, Christian E. "Analysis of an equal-cost multi-path algorithm." (2000).
25. Sundararaj, Ananth I., and Peter A. Dinda. "Towards Virtual Networks for Virtual Machine Grid Computing." Virtual machine research and technology symposium. 2004
26. Ward, Brian. The book of VMware: the complete guide to VMware workstation. Vol. 1. San Francisco: No Starch Press, 2002.
27. https://www.vmware.com/support/ws55/doc/ws_net_configurations_common.html
28. Cisco, Cisco IOS XRv Router Overview. http://www.cisco.com/en/US/docs/ios_xr_sw/ios_xrv/install_config/b_xrvr_432_chapter_01.html
29. Savvas Zannettou, Michael Sirivianos, Fragkiskos Papadopoulos, "Exploiting Path Diversity in Data centers Using MPTCP-aware SDN", 18 Aug 2016, IEEE Symposium on Computers and Communication (ISCC), DOI.10.1109/ISCC.2016.7543794, ISBN 978-1-5090-0679-3
30. Yifei Lu, Shuhong Zhu, "SDN-based TCP Congestion Control in Data Center Networks", 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), 18 February 2016, DOI.