Software Defined Networking (SDN): A Survey on SDN Components and Controllers with Structural benefits for Scalability.

Pushpa. J[1,#], Assistant Professor, Department of BCA, Jain University, Bangalore.

Pethuru Raj[2], Chief Architect, Site Reliability Engineering (SRE) Division, Reliance Jio Infocomm Ltd, Bangalore, Karnataka, India.

[#]Corresponding Author, Email: pushpaj.mca07@gmail.com

Abstract

Network resources have to be optimally and organically leveraged to accomplish communication and data transmission requirements in an efficient manner. That is, the overwhelming and widely insisted mantra of "more with less" has to be linked up with networking also. Network resources have to be sliced accordingly and adaptively in order to reduce wastage, but at the same time, it is crucial to ensure business needs without any hitches and hurdles. Various network functions have to be accelerated and augmented with the application of a bevy of powerful automation technologies. Therefore, new disciplines such as software defined networking (SDN) and network virtualization (NV) have emerged and evolved fast in the recent past. Network resources and functions are being virtualized and network functions are being implemented and presented as microservices. Further on, there are standards-compliant SDN controllers and platforms, which facilitate the extraction of the most crucial network intelligence being embedded in network elements such as routers, switches, etc. In other words, the controlling logic gets separated out of routers and switches and is being presented as a centralized controller running on commodity servers. This meticulous abstraction goes a long way in bringing the much needed flexibility, extensibility, visibility, control ability, security, etc. in networking, which is becoming hugely complicated due to the leaning towards distributed and decentralized computing. This survey paper is prepared with the sole aim of telling all about the origin, the compelling features of SDN controllers, the unique capabilities,benefits of SDN and its topological behaviour.

Keywords: DSCP, RSVP-TE, GRE, SSH, OVSDB.

1. Introduction

Current technologies like the Internet of Things (IoT), edge computing and 5G networks are leading technologies for providing better service in terms of high availability, high speed access, scalability, dynamic configuration and security. Static nature of traditional networks is not fit for agile technologies to overcome this virtualization and datacenter. The traditional network has a high level of OPEX which is static in nature. Virtualization creates a virtual real-world physical object using software components whereas a data center is an in-house tool to store data within an organization's local network. An interface is required to communicate between the Virtual Network's (VN-VN), virtual network to hardware device and data center which SDN can do. So SDN can be defined as an interface between the physical object and virtual object. SDN brought tremendous changes in IT industries in 2011 when the first standard OpenFlow was developed based on Ethane a network architecture developed by Martin Casado a Ph.D. student of Stanford University, an OpenFlow is a communication protocol. SDN definition varies based on their perspective, ONF defines SDN as "The physical separation of the network control plane from the forwarding plane." VMware defines "SDN as reproducing a network environment using software". The key idea of SDN is to programmable managing the network by separating the forwarding devices and control system. Forwarding devices are the networking device for forwarding packets/frames such as switches and routers whereas the control system is a most important part of SDN[1-2], it acts as interface or intermediary between the application layer and forwarding device which is commonly named as a controller.

SDN is network centric by incorporating the software module to monitor, policy management and configure the network by programming whereas traditional network is device centric which needs to manage manually for each type of device makes the process tedious. If we look into the comparative benefit of SDN over traditional networks, multivendor support, dynamic configuration, secured communication using tunneling and data flow optimization are incredible services provided by SDN.

Several surveys on SDN differ with analysis [3], whereas in these papers we will concentrate more on SDN controller and structured based analysis. Before exploring the functionality of SDN, understanding the framework of SDN is very important. In section 2 will learn about architecture, section 3 will spread the light on benefits of SDN implementation in terms of use case and our journey of survey helps to understand the role of SDN on service model in section 4. Before moving to any part of SDN, analysing the notion of SDN controller is very important as it is a master controller of the entire system as seen in section[5]. Forwarding plane can follow the different mechanisms and structure which are discussed in section[6] and[7]. Every system/technology has its benefits and draws back, finding the solution for the challenges is research, the subsequent section tries to address some of the pitfalls in SDN.

In our survey we focused on topologies, Performance measurement of SDN controllers using ITZ on Internet zoo topology[7], Controller placement algorithms and we also demonstrated clustering of switches and controllers[4-5]. The custom topology which is developed by clustering the nodes in hierarchical methods and applying k-mean results in better performance.

2. A Brief on Software-defined Networking (SDN)

Software-defined networking(SDN) along with network virtualization brought a new paradigm in the networking domain. As demand increases for the internet in 2000 new challenges arise, especially to manage the networking devices, scaling, resilience and automation are the common factors integrated in it. Working structure of a traditional network is a tedious process for monitoring or managing any of the above factors. Many protocols are introduced to check the network stability such as SNMP,Netflow, ICMP and many more which still fail to automate the updated policy in networking devices due to manual setting up of individual devices.

SDN disaggregates the controlling decision from forwarding devices(router/switches) which is sprouted up with the concept presented in Forwarding and Control Element Separation" (ForCES). Though the decoupling principle is more effective the challenges arise about standard application interfaces between those components. OpenFlow protocol is introduced to overcome this drawback and is promoted by Open Network Foundation.

SDN conjugated with Network Function Virtualization(NFV) bringing in a bevy of pragmatic yet frugal innovations in data centers, cloud computing and in all networking arenas. NFV spun up the virtual switches and the forwarding rules are centralized in the SDN Controller. As depicted in the macro level SDN Fig. 1, API(Application Programming Interface) is placed between the application and controller which are known as northbound API. Rest API is commonly used northbound API which interconnects with technologies such as data warehouses,management system,security policy and other Application based services to controllers. At the other end, southbound API such as OpenFlow protocol is used for connectivity between controller and switches. The rich set of API overcomes the constraints of ForCES and integrates the intelligence of all kinds of network appliances, dynamic policy monitoring, remote communication and many others which results in agile technology.

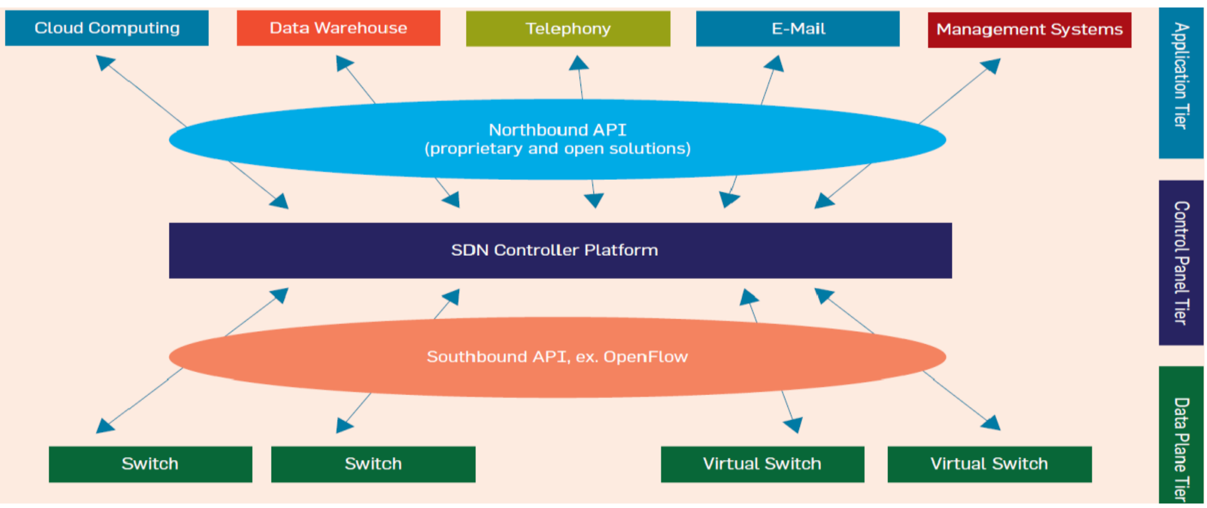

Fig. 1 – The Macrolevel SDN Architecture

SDN is amalgamated with most of IT technologies such as IoT, Cloud Computing, Digital Twining, 5G and many more. Most of them demand security, scalability and robustness. As shown in Fig. 2 distributed controllers can communicate and provide scalability for increasing demand in host devices. Controller planes centralize the rules to block any malicious device or to navigate the flooding traffic to avoid the downtime. As discussed earlier about the NFV which helps to create virtual switches to meet the demands and manages the traffic. Programmability, proactive/reactive flow management, interoperability, and security are integrated features of SDN which are promoted to embed in current inter networking.

As in the OSI model, the control plane manages switch and routing tables while the forwarding plane actually performs the Layer 2 and 3 packet filtering, forwarding and routing policy. In short, SDN Fig. 2 decouples the system that makes decisions about where traffic is sent (the control plane) from the underlying system that forwards traffic to the selected destination nodes. This well-intended segmentation leads to a variety of innovations to incorporate.

SDN orchestrates a 5G network to provide the ultra reliable low latency demanded by billions of devices. Instantaneous connection , high speed with low latency of 1 millisecond are the characteristics of 5G network which is indeed software driven with the supportive technologies of SDN, NFV and cloud server. The SDN controller plane is not only limited to a homogeneous network but also heterogeneous network which is one of the major requirements in the 5G network.

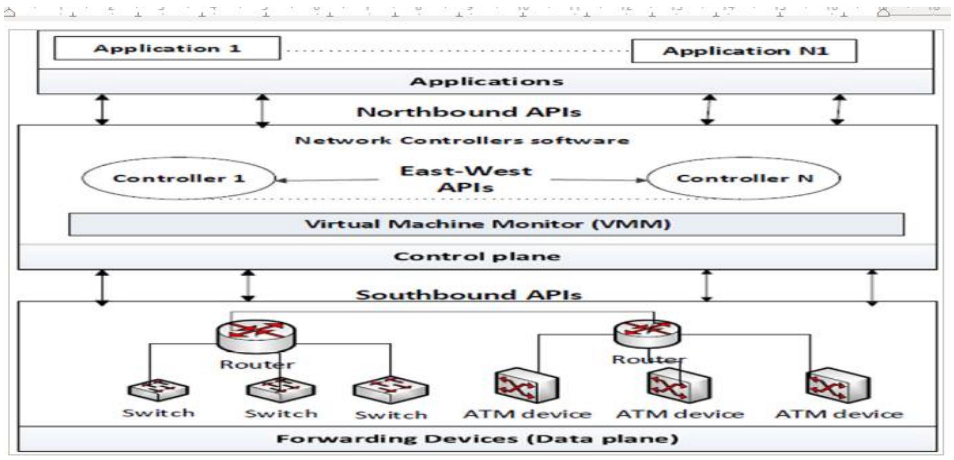

Fig. 2 – The Separation of Control and Data Planes

2.1.1. SDN architecture [13] is envisioned by ONF which is bundled with software modules. SDN is logically classified into three segments which are known as a plane, such as Application Plane, Control Plane and data plane. The framework of SDN is depicted below in Fig. 5, where each layer is abstracted from one other to make interaction easier with the help of protocols. OpenFlow [18] protocol and REST API are used to communicate between the planes. There are many other protocols that come into picture which are discussed in section 3. Let us discuss the SDN Planes in detail with the help of default SDN Architecture in Fig. 3,

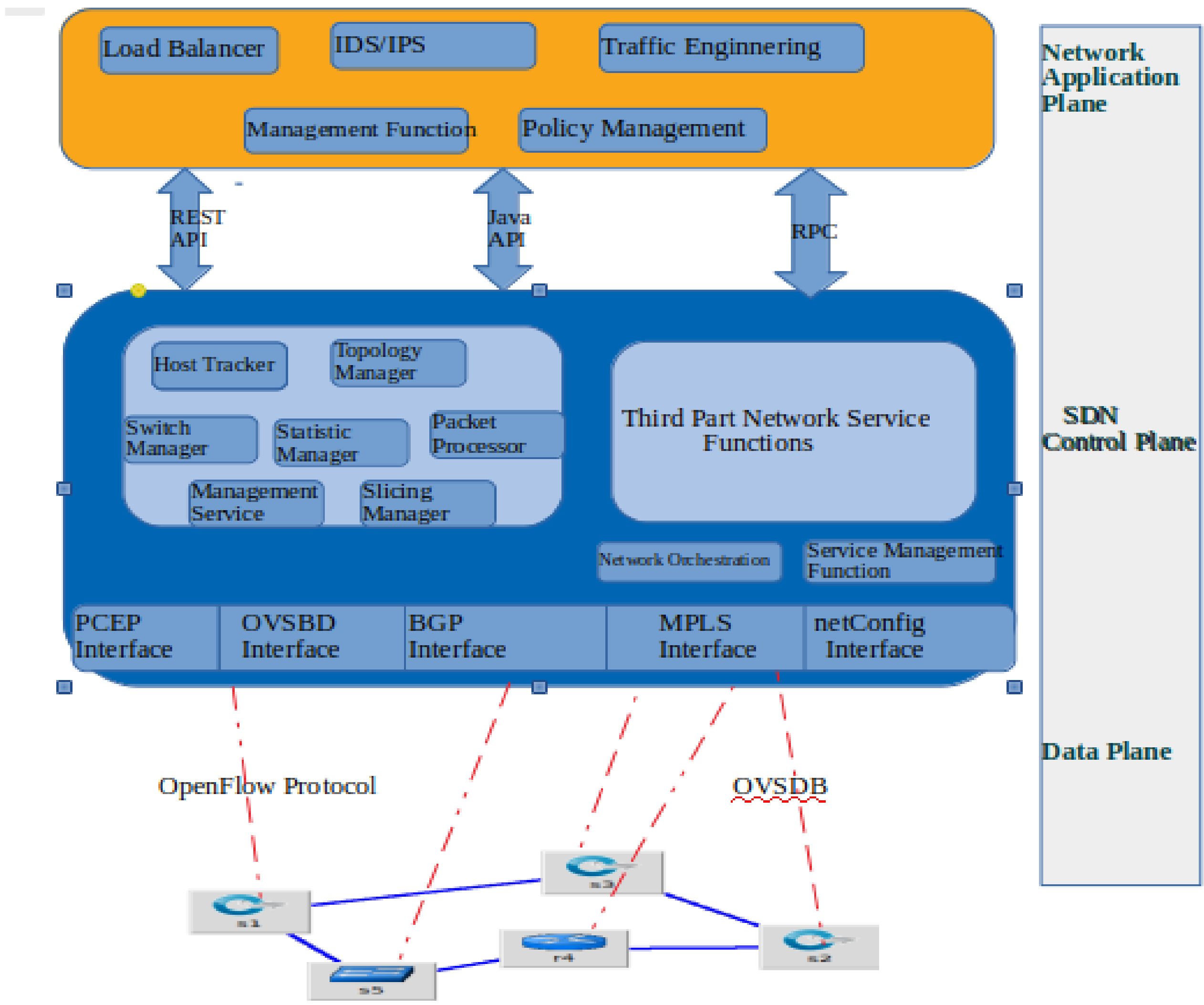

Fig. 3 - SDN Architecture

a. **Control Plane:** It is also called the brain of SDN which manages the flow control, Topology management, switch management, QoS Service, Host tracker and so on. It contains one or more controllers which take the decision for forwarding packets or not based on the different types of rules and policy as defined in application services. This forwarding decision is passed to routing devices in a data plane as flow rule. Flow rule is the set of instructions sent by the controller to update the flow table of an open virtual switch (OVS). OpenFlow protocol is used to communicate between switches and controllers. Controllers can be centrally located or distributed. If controllers are centralized then network devices are controlled and managed with common policy and monitored within the same domain but suffer from a single point of failure and inefficient performance in load sharing, whereas distributed controllers can overcome this drawback and provide better performance in load sharing but raises the challenges of policy management and intercommunication.

If we look into OpenFlow Networking, there are many functions that are discussed and described about the control plane. Basically services in controller is classified into two form they are,

- Network Base Services.
- Third Party Services.

In the network base services, the services such as topology manager, forwarding rule manager and ARP manager are used for gathering and managing the information of the network. Whereas third party services are proprietary based services such as network orchestration function and Event management services.

b. **Data Plane:** The main function of routing devices is to forward the data traffic generated by the user based on the forwarding policies within the network infrastructure. SDN Made the forwarding policies separate from the network device and placed it in a centralized plane which is known as the control plane. This structure is centralized in nature by placing the forwarding decision in the center which controls the devices in the form of rules. OVS will communicate with other OVS and also with traditional switches with the help of standards. There are many proprietary based switches available in industries which support openflow protocol. Each switch should maintain flow tables to hold the routing information.

c. **Application Plane:** It contains the service to define the behavior of the network. This layer also called Northbound API is based on the REST paradigm, Application plane can expand its functionality as required by binding the services which are nonpartisan from data plane devices. Primary service is to define the policy to instruct the forwarding devices. A northbound interface such as REST API is used to interact with the control plane [12] using a network programming language which eases and orchestrates the network.

2.1.2. Open vSwitches: A vSwitch in a virtual environment programmed to build its forwarding information base in a table known as flow table from SDN controller via openflow protocol. Flow Table contains flow entries obtained from the controller for the action to handle the packet. The fields of flow entry are given in Fig. 4.

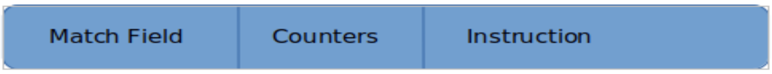

Fig. 4 - Flow Entry

Communication message between switch and controller is classified based on the intimation from whom? Usually the message will be from switches.

- Control-to-Switch: Message sent from controller to switch.
- Asynchronous: Message sent from switch to controller.
- Symmetric: communication message either from controller or switch.

2.1.3. Open vSwitch Database(OVSDB): is a protocol used to configure the ovs. Main function of OVSDB is to create/delete/modify the bridge,port and interface of OVS. We know that there are many logical paths between ovs and controllers referred to as bridges. A Management service requires information of brides which is stored in ovsdb and accessed through management protocol [RFC 7407].

2.1.4 SDN Protocols: Protocols are the standards of rules for interaction between two entities. Will discuss the protocols used in SDN[12][20].

a. OpenFlow: It is a standard for communication between data plane and controller. It is also called an an interface which updates the flow table of Open Virtual switches. OpenFlow basically parses the packet for matching the field to the destination address in the flow table. Many versions of OpenFlow evolve with the extension of new service and great arguable protocol MPLS is included in SDN protocol and named as MPLS-TP [13] OpenFlow protocol to support Super Packet Transport Network (SPTN) for OpenFlow Logical switches using Table Type Pattern (TTP) which make use of multiple tables.

b. BGP: The routing information along the path matrices and cost matrices are essential for any routing algorithm to provide the best path. It adapts path vector or distance vector routing algorithm within in network of the autonomous system between gateway hosts for exchanging the routing information.

c. PCE: Path Computation Element protocol is used to compute the path based on constraints applied to find the route between the device using LSP tunnel.

d. ForCES: Forwarding and Control Element Separation is one of the southbound interfaces used for interconnection between the logical functional blocks to modify the behavior of network devices.

e. OF-CONFIG: Configuration of OpenFlow switches such as queues and ports which are not performed by OpenFlow switches is accomplished by OF-CONFIG. It allows modifying queue properties for improving QoS.

f. MPLS: Multiprotocol Labelling Switching [14] reduces the forwarding tables in switches by replacing the IP address with the label and also provide more level of hierarchy for tunneling.

g. NETCONF: It is network management protocol for configuring and fetching the network devices information. This protocol performs better than Simple Network Management Protocol (SNMP) and Command Line Interface Protocol (CLI) in of Scalability and efficiency.

3. SDN Controller

SDN Controller: Control Plane is placed in the center of SDN architecture which orchestrates, manages the data plane and interact between northbound and southbound application to provide efficient logistics to handle the networks. It is logically centralized and manages the virtual resources and is stated as blackbox. SDN controller[20] is encapsulated with Network service function, Network orchestration function and service management function.
Network Service function contains topology management, state Management, slicing manager and switch manager.

Network orchestration Function in the way of creating the overlay and link to other management elements in support to configure with other entities such as northbound APIs, southbound entities, OVSDB, VxLAN and STT.

Management function services are infrastructure management, traffic analysis, Configuration, service persistence and software distribution/ upgrade due to the global view of SDN.

The goal of the SDN controller is to control and manage the network element by logically placing the forwarding decision in center as shown in the Fig. 5 and performs better than traditional network by handling traffic flow. Basically, SDN switches flow table is not having any flow rule, it requests the flow action from controller and updates its table. OpenFlow protocol is used to communicate between switches and controllers. At the same time northbound APIs allows to interact with third party application

SDN controller can be either distributed or centralized, as we discussed SDN controller is logically centralized by obtaining the global view of network which helps to calculate the path computation, policy management and shortest distance calculation using distance as the weight and queue length. But it faces challenges such as scalability, reachability and traffic management for the wide area network and growing traffic in mobile networks. This can be overcome by a distributed controller by creating the zone and interconnecting between the controllers of the zone. Distributed controller in the figure [6] solves the problem faced in a centralized network but throws the challenges of interconnection between the different controllers and policy management in the intrazone. Hence, in many research hybrid controllers is the outcome as solutions [16] [17] [18].

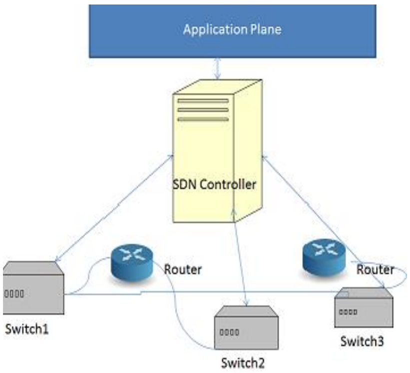

Fig. 5: Centralized Controller

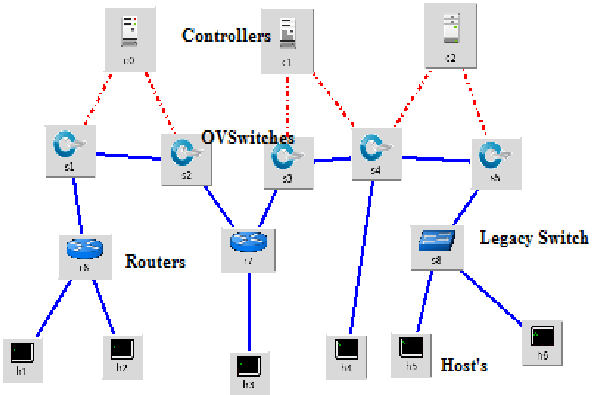

Fig. 6 : Distributed Controller Below table

From the table 1 shown below are some of the key features of SDN Controllers and compared between centralized and decentralized controllers[20][22] [26]. The table from[22] information concluded that some of the centralized controlle such as POX and NOX does not support multithreading, REST API and not compatible with OpenStack.

Features	**NOX**	**POX**	**FloodLight**	**ODL**	**Open Contrail**
Platform Support	Mac	Linux,Mac, Windos	Linux,Mac,Windos	Linux,Windos	Linux, Windos
Compatible OpenStack	No	No	Yes	Yes	Yes
Language	C++ Only	Python	Java,Python and language supported by REST API	Java	Java,Python
Southbound API	OpenFlow	OpenFlow	Open Flow	OpenFlow and other SB Protocols	XMPP
Northbound API	Switches	API	Northbound (java, REST)	Northbound (java,RPC, REST)	BGP,Netconf
Open Flow Support	1.0,1.3	1.0,paritial support for 1.1.0	1.0,1.3,experimental support for 1.1,1.2	1.0,1.3	-(It uses its own protocol for Managing master table)
REST API	No	Yes	Yes	Yes	Yes
Distributed/Centralized	C	C	D	D	D
Multithreading	No	No	Yes	Yes	Yes

Table 1: Comparison of SDN Controllers

3.1 SDN Models: OpenFlow support two models for forwarding the information and set up the path across the network.

They are Proactive and Reactive Model used to build the flow entry in switches. Switches use ASICs for hardware traffic forwarding. Controller should place nearer in reactive model than in proactive model.

a. Reactive Model: Switches learn the flow entries dynamically as the packet arrive. In this model switches will be holding no flow entry before any traffic. Once the packet arrives to switches it send packet in message to controller and intern controller sends packet out with message of carrying the action as flow entry to update the table. By default, SDN controller flow reactive flow.

b. Proactive Model: In this model the flow tables are filled with flow rule in prior. Once the packet arrives to switch it lookup into flow table and process the packet instead of forwarding to controller. In mininet we use flow dump services to insert the rules into table as shown below.

ovs-ofctl add-flow s1 in_port=1, actions=output:2 # ovs-ofctl add-flow s1 in_port=2,actions=output:1

Using Proactive model will forward the packets faster than reactive but is not applicable in distributed network or mobile network which keep vary in network connectivity.

3.2 Structural Benefit:
In our previous experiment on topology-based performance measurement we compared single, tree and linear topology and result show the limitation of the structural arrangement of switches. Our study results the benefit of hierarchical structure of tree topology gives low latency than other two -topologies. The resultant graph is given in Fig. 7.

Type	min	avg	max	mdev
Tree	2.53	32.252	278.558	82.267
Linear	0.975	74.51	721.9	215.836
Single	0.711	42.114	410.529	122.802

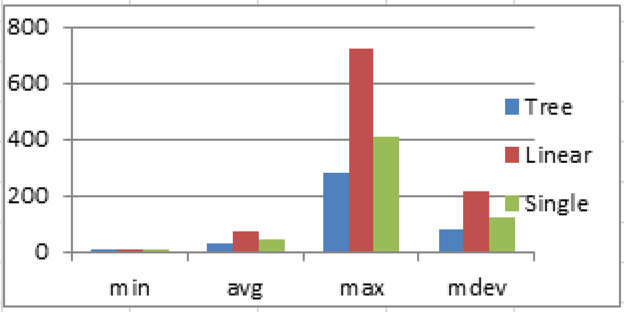

Fig. 7 : RTT of FloodLight

3.3 Scalability through clustering: SDN is matured enough to adopt in new technology due to its intelligent management of forwarding devices and controlling the data flow. But scalability of SDN controller for maximum utilization of available resource for achieving low latency and throughput created a space for researcher to explore the area on scalability. As host increases switches will also grow to forward the data in network[21]. SDN controller manages the switches efficiently to provide efficient service for the customer, but managing the growing switch is not compatible in the present SDN. Incorporating the algorithms as northbound APIs which provide the solution for scalability. In our survey on scalability [19][20][21][22]

Our experiment shows the significant growth in SDN scalability using clustering. Clustering is an approach to group the similar data point, here switches are considered as data point for grouping. Clustering helps to communicate between the switches which are closer, load balancing and reduce the traffic. It also helps to find the position to place the controllers and number of controllers required. Consider the example is demonstrated using dendrogram to show the result of hierarchical clustering for the delay matrix which clusters the nearest switches in hierarchical structure and the switch connectivity is shown in Fig. 8.

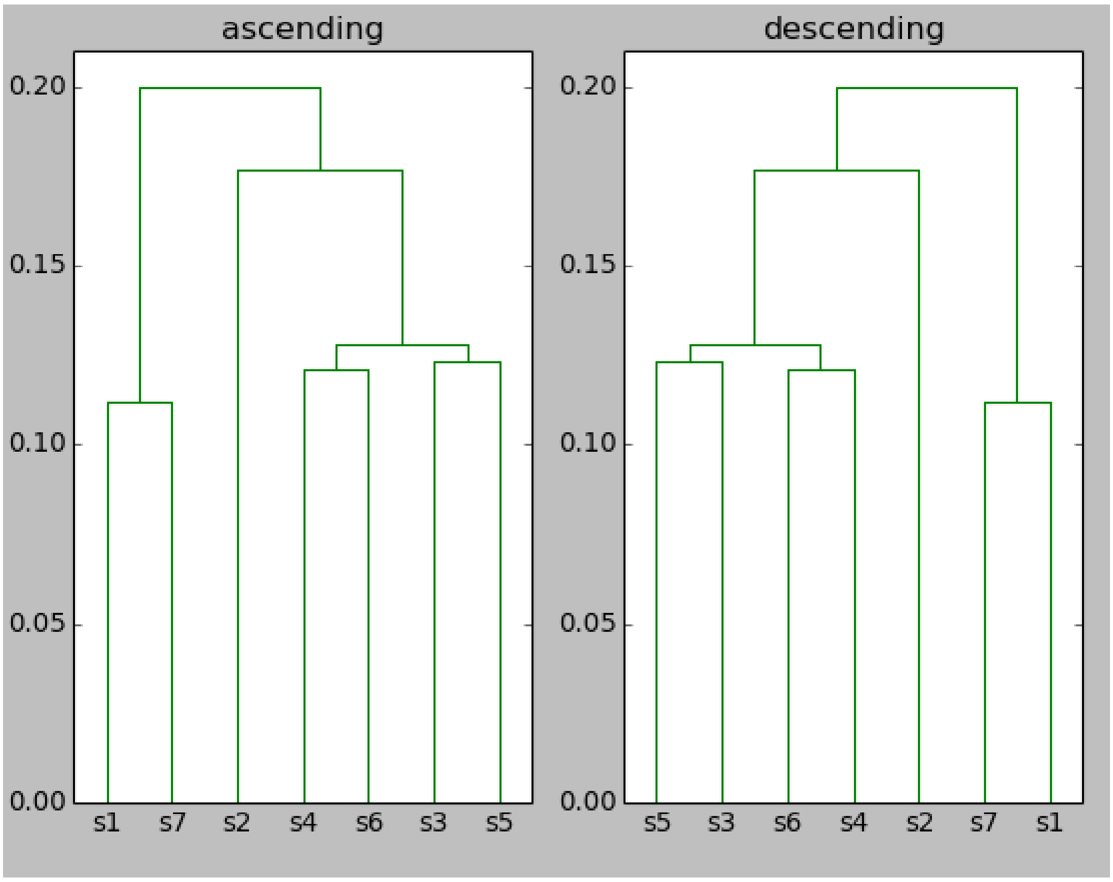

Fig 8: Dendrogram Hierarchical Clustering

Algorithms used in our experiments are kmeans clustering, switchbin placer, random placer and hierarchical clustering algorithm for comparing the round trip time to find the efficient algorithm executed on two VM as shown in Fig. 7.

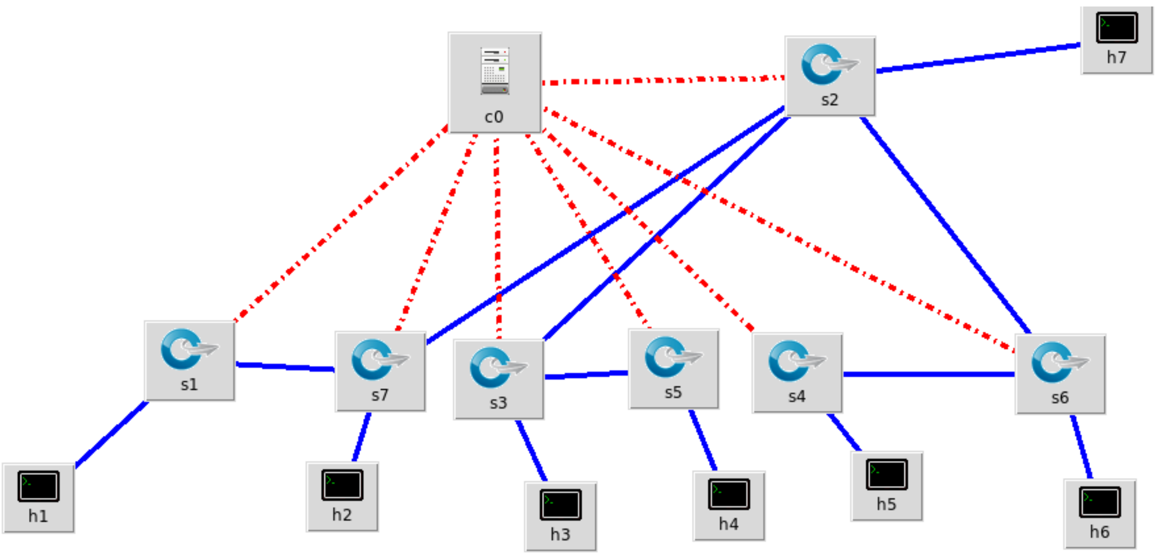

Fig. 9: Hierarchical clustering of SDN OVS.

SwitchBin Placer algorithm[25] is used to place the switches evenly which are close to the host. It won't consider the high priority hosts for clustering which leads to high round trip time but efficient than Random placer clustering which follow round robin scheduling algorithm. Kmean clustering[21][23] is more suitable than hierarchical clustering in distributed network and also with efficiency of algorithm is O(n) and efficiency of hierarchical clustering is O(n^2). kmean algorithm fails in interconnecting switches to intermittent point to achieve optimal clustering over hierarchical clustering[25]. Hence, our custom topology with conjugating with heirarchical and k-mean results the less delay and is measured using round trip time.

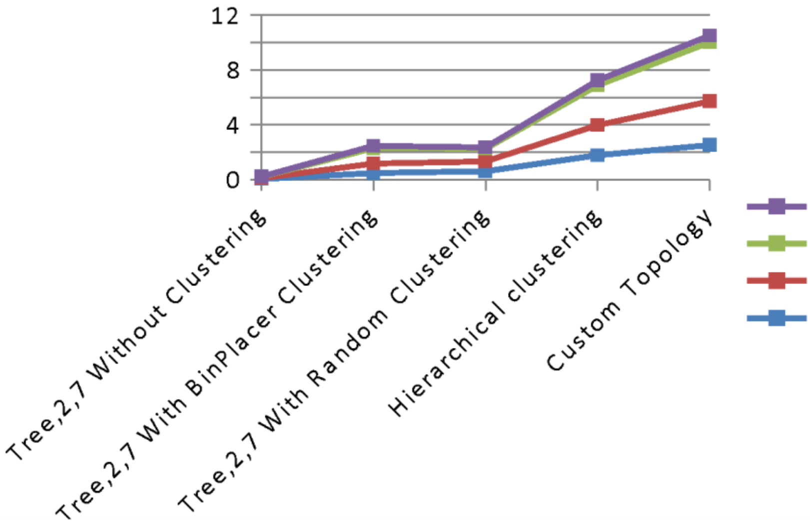

Fig. 10 : Comparative study on Scaling.

4. The Distinct Benefits of Software-defined Networking

The benefits of SDN are definitely diversified and gradually enlarging. SDN is widely implemented in cloud computing for the following featural benefits

a. The centralized network control and programmability – SDN avoid manual updation by centralizing the controller plane and mange via programmable interface. It also provide the flexibility to modify the controller application for current scenario.

b. Dynamic network segmentation – NFV will aids to manage the network effectively by segmenting the network. VLANs provide an effective solution to logically group servers and virtual machines at the enterprise or branch network level. Managing the VLAN by grouping them to provide the privileges can be programmatically achieve via SDN.

c. High Visibility of VMs – Hypervisor creates Virtual Machine (VM) n which vswitch can be run which definitely resolves the scaling problem but fails to provide the global visibility which can be alleviated using SDN embedded hypervisor.

d. Capacity utilization – SDN supports both centralized and distributed data center which results in load balancing or resources utilization .

e. Network capacity optimization - The spine leaf architecture poses inefficient traffic flow due to redundant connectivity. There are innovative solutions such as link aggregation, multi-chassis link aggregation, top-of-rack (ToC) switches and Layer 2 multipath protocols. These are able to fulfill load-balancing, resiliency and performance requirements of dense data centers. However, these are found to be complex and difficult to manage and maintain. The SDN paradigm enables the design and maintenance of network fabrics that span across multiple data centers.

f. Distributed application load balancing - With SDN, it is possible to have the load-balancing feature found in many controllers which are distributed in nature such as ONIX. Multiple SDN controllers distributed geographically but connected to each other to share the networking view and also share the load in terms of network traffic.

5. Results

The experimental result on structural benefits in section 3 has proved that as host/nodes increase tree topology which is hierarchical in structure is having less RTT and reduces the latency.

In section 4, considering the topological behaviour and its result, the custom topology with combine the k-mean and heirarchical algorithm results the best performance for scalability.

6. Conclusion

SDN can be called as game-changing technology for next-generation IT environments which easily integrate with other technologies. It eliminates the network complexity with simple commands remotely in the midst of multiple heterogeneous network elements. Significant impact on controller placement with popular algorithms evidence the difference in response time. The Structural behavior and controller placement of SDN also play a vital role for better performance. Hence supporting technologies such as cloud computing and NFV should be integrated for clustering and virtualization the resources for scalability and efficiency.

References

1. In Open Networking Foundation, "SDN Architecture Overview", Version 1.0 December 12,pp. 14-18, 2013.
2. Slavica Tomovic, Kenji Yoshigoe, Ivo Maljevic, Igor Radusinovic(2016), " Software-Defined Fog Network Architecture for IoT", Springer Science,Business Media New York, pp. 3-5, DOI 10.1007/s11277-016-3845-0.
3. Jen English and Stan Gibilisco, "SDN controller (software-defined networking controller)", searchnetworking.com.
4. Peng Xiao, Wenyu Qu, Heng Qi, Zhiyang Li, Yujie Xu, "The SDN controller placement problem for WAN", 2014 IEEE/CIC International Conference on Communications in China (ICCC).
5. Brandon Heller, Rob Sherwood, Nick McKeown(2012), "The Controller Placement Problem", HotSDN'12, pp. 2-5, August 13, 2012, Helsinki, Finland.
6. Document from Microsoft, "Skype for Business Software Defined Networking (SDN) Interface", Applies to: Lync Server 2013 | Skype for Business 2015.
7. S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The internet topology zoo.
8. OpenShift from website: https://docs.openshift.com/container-platform/3.3/architecture/additional_concepts/sdn.html.
9. Chapter 5: "Introducing SDN Control in MPLS Networks", Stanford.
10. Krzysztof Grzegorz Szarkowicz, Antonio Sanchez Monge, "MPLS in SDN Era", ch 19: Transit fast restoration based on RSVP-TE.
11. Brent Salisbury's Blog, "Open vSwitch GRE Tunnel Configuration", Network Static, Arp 2012.
12. OpenFlow Switch Specification Version 1.0.0 (Wire Protocol 0x01) December 31, 2009.
13. Howtoforge, "Software Defined Networking (SDN) - Architecture and role of OpenFlow".
14. Nathan Cranford(2017), "How SDN can benefit healthcare", RCRWireless News.
15. Kok-Kiong Yap , Murtaza Motiwala , Jeremy Rahe , Steve Padgett , Matthew Holliman , Gary Baldus , Marcus Hines , Taeeun Kim , Ashok Narayanan , Ankur Jain , Victor Lin , Colin Rice , Brian Rogan , Arjun Singh , Bert Tanaka , Manish Verma , Puneet Sood , Mukarram Tariq , Matt Tierney , Dzevad Trumic , Vytautas Valancius , Calvin

Ying , Mahesh Kallahalla , Bikash Koley , Amin Vahdat, Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering, Proceedings of the Conference of the ACM Special Interest Group on Data Communication, August 21-25, 2017, Los Angeles, CA, USA[doi>10.1145/3098822.3098854].
16. Carla Mouradian, Diala, Sami(2017), “A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges”, DOI: 10.1109/COMST.2017.2771153
Publisher: IEEE,page(s): 416 - 464.
17. ATOS Apprenda, “PAAS AND SDN: A MATCH MADE FOR THE CLOUD”, Blog.
18. Wolfgang Braun, Michael Menth(2014), “Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices” , Future Internet 2014, 6, 302-336; doi:10.3390/fi6020302.
19. Michelle McNickle(2014), “Five SDN protocols other than openflow”, TechTarget.
20. J. Pushpa,Pethuru Raj, “ Performance Enhancement of Fog Computing Using SDN and NFV Technologies”, chapter 6, pp. 8-13, Book: Fog Computing, Editor:Zaigham Mahmood,Springer(2018).
21. POCO: Pareto-Optimal Controller Placement, https://github.com/lsinfo3/poco.
22. Pushpa J, Pethuru Raj,. (2018, February). Topology-based analysis of performance evaluation of centralized vs. distributed SDN controller. In Current Trends in Advanced Computing (ICCTAC), 2018 IEEE International Conference on (pp. 1-8). IEEE.
23. MuratKarakus, Arjan, “A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN)”, Elsevier,Volume 112,15 January 2017, Pages 279-293.
24. Scikit-learn: Machine Learning in Python, Pedregosaet al., JMLR 12, pp. 2825-2830, 2011.
25. Othmane Blial,Mouad Ben Mamoun,andRedouane Benaini,”An Overview on SDN Architectures with Multiple Controllers”, Journal of Computer Networks and Communications ,Volume2016, Article ID9396525, pp. 2-4 .

26. Zhu, L., Karim, M.M., Sharif, K., Li, F., Du, X., Guizani, M.: SDN Controllers: Benchmarking and Performance Evaluation. arXiv:1902.04491v1. [cs. NI] (2019),pp. 4-5.