Journal of Critical Reviews

ISSN- 2394-5125 Vol 7, Issue 11, 2020

APPROACH ON AUTOMATIC PROVISIONING OF SWITCHES IN DATA CENTER THROUGH SOFTWARE DEFINED NETWORK CONTROLLERS

Manikandan Ramanathan[1*], Kumar Narayanan[2]

[1]Reserach Scholar, Department of Computer Science & Engineering, Vels Institute of Science Technology & Advanced Studies (VISTAS). maniramphd@gmail.com
[2]Associate Professor, Department of Computer Science & Engineering, Vels Institute of Science Technology & Advanced Studies (VISTAS)

Received: 08.04.2020 ***Revised: 15.05.2020*** ***Accepted: 06.06.2020***

Abstract
In the emerging IT industry, the data center network consists of switches and routers which are high in cost, maintenance, load. To reduce these challenges, Software defined controllers can be implemented. In the paper we will be discussing the component of Software defined network and its sequences of configuration mechanisms through flow tables with minimal fields to be configured in network devices.

Keywords: Data Center, Software Defined Networking, Network Elements, Controllers, SDN Agent.

DOI: http://dx.doi.org/10.31838/jcr.07.11.101

INTRODUCTION

The switches in network is used to transfer the data in the network from core to distribution and from there to Access and vice versa. Real challenge comes on the speed in which the data transfers. Real world expects the very high speed of data transfer considering the current big data and increase in access devices. At present the switches has both data and control plane along with the network processors. Control plane in Switches and Routers are used for running the control plane protocols, control packet processing, configuration of the switches through various configuring terminals like CLI, SNMP, WebUI and Rest API [1-2].

Software Define Networks provides new ways to design the network for speedy network operations. Instead of implementing new different protocols, it is decided to separate the functions of control and data plane forwarding functionalities of switches and routers. However, the actual switches become programmable forwarding plane and other processing happens in central system.

The SDN architecture will have north bound and south bound interfaces [1]. The controller is the device which is responsible for processing control packets and dynamic execution of protocols like MAC learning, VLAN, IP routing, loop detection, security protocols like PNAC, DCHAP, etc. Once the protocol decides based on the standards and rules the forwarding tables in the actual switches are configured by the SDN controller on the forwarding decisions made by protocols.

Software to be designed such a way that the applications that interacts with SDN controller and the low-level interface modules that interacts between controller and switches. The Application interface to SDN controller is north bound interface and software used to interact between applications and switches are called north bound API and the switch or router connected to the SDN controller is called south bound interface and the software used to configure from controller to switch is called south bound API.

The control plane of the network elements in data center network is decoupled from the hardware and implemented as a separate software application. This architecture is modular and provides more programmable pluggable modules and it becomes ease of controlling to the network administrators.

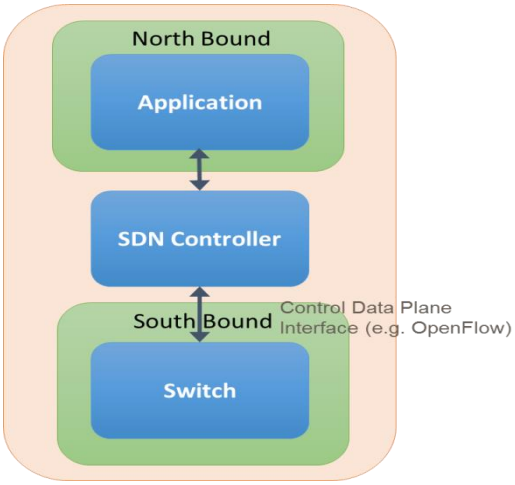

Figure 1: SDN Architecture

SOUTH BOUND INTERFACES

Following are the south bound interfaces [7],

- OpenFlow
- ForCES (Forwarding & Control Element Separation)
- PCE (Path Computational Element)
- XML

Open Flow

Open flow is a protocol defining an interface or communication mechanism between SDN Controller and network elements for communication (i.e., communication between control plane in controller to data plane in network elements) over the network. Switch functions as the OpenFlow Client. SDN controller functions as the OpenFlow Server [13].

OpenFlow completely replaces legacy control and management planes, All ports on the switch are controlled by the OpenFlow controller. All switches in the network/domain must be OpenFlow controlled.

Open Flow switch will have chain of tables also known as Multiple flow table which will have the actions for the packets

which are ingress and egress rules based on the SRC and destination MACs, VLAN IDs, IP, etc [13].

Open Flow controller

Provides a secure interface for configuring the OF switches. Provides API's to application software to manipulate the network switches and modify any topology. This maintains the configuration database for the OF switches and pushes it to the network switches. Configuration can be done by CLI, SNMP or through REST API to Open Flow controller [9]. Predefined Configuration files can be restored to Open flow switches. And, it maintains the forwarding table database which is also known as network topology formation information. Multiple virtual switches can be created in the flow controller for one physical network deployment. Open Flow controller also monitors the network health and can monitor all the control packets and can tweak the packets to understand the kind of packets getting transmitted.

OVSDB- Configuration Database

OVSDB is Open virtualized switch data base, this is maintained in the switches to hold the configuration database. OSVDB can be maintained at SDN controller and on the real switch based on the critical deployments, if the deployment is not critical OSVDB can be placed alone in SDN controller itself. The OSVDB manager will interact with CLI and other configuration management and configures to OSVDB data base. Configuration to be stored on the storage, it survives after a reboot [9].

PROVISIONING AND MONITORING OF SWITCHES THROUGH SDN CONTROLLERS

North bound Applications such as Virtuora NC, Kubernates are collection of applications, automation framework, management deployment models can be used as a management center for the switches and routers [11].

These north bound applications connect to the SDN controller (eg, Volta network's controller) through a dedicated data communication channel [11]. The communication mechanism is through Yang models or by using Netconf protocol [9]. These controller's act as a single point of contact for all the switches and routers in the network. This controller will have the protocol stack such as Layer 2 and Layer 3 stack. This controller will act as a source to configure the switches. The switches and routers are the network elements which will have the agent in their software architecture. These controllers will send messages and communicates to network elements through these agents in network devices.

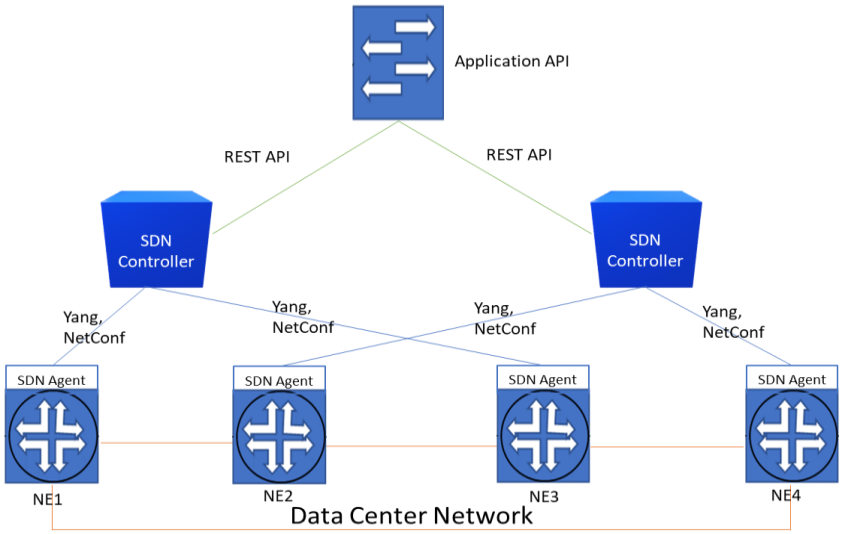

Figure 2: SDN Data Center Topology

Provisioning flows from the north bound APIs to controllers, controllers process them and configures the action of the protocols to network elements. Monitoring is performed by sending statistics, counters, performance monitor actuals, status of the device, ports, links are sent to north bound APIS through the reverse path – Network element HW sent to the agent, from agent to controller, controller to application APIs.

FLOW TABLE CONFIGURATIONS

The flow tables are maintained in network elements. The match fields are identified in various network layers [2] and the data traffic parses through various tables which are mentioned below,

A. Input Port table

The input port is the port where the data enters and if the input port is matched in data, it sends the packet to L2 Table, if it does not match it drops the packet from further processing.

Table 1: Input Port Table

Match Fields	Instructions
Input_Port	Goto L2 Table
Table Miss	Drop

B. L2 table

Once the input port table is matched, the packet sent to L2 table. Here following matches takes place, it can have 128 entries or more based on the HW capability. Match field combinations are as below

- Input_Port, ETH_TYPE
- Input_Port, ETH_SRC, ETH_DST, VLAN-ID
- ETH_SRC, ETH_DST, VLAN-ID can be wildcards

Priority is applicable for this table

Table 2: Layer 2 Table

Match Fields					Priority	Actions	
Input Port	ETH_SRC	ETH_DST	ETH_TYPE	VLAN-ID	Priority	Write-Action <To CONTROLLER/ LOCAL/Drop>	Goto L3 Table
	Table Miss						Goto L3 Table

If the match is successful or not it is sent to L3 / L4 table based on the configuration.

C. L3 table

Once the packet received for L3 table parsing, following matches takes place,

- Src-IP, Src-Port is used in case of upstream
- Dst-IP, Dst-Port is used in case of downstream
- VLAN Priority is marked

It is then pointed to Egress table

Table 3: Layer 3 Table

Match Fields						Actions	
Input_Port	Src-IP	Src-Port	Dst-IP	Dst-Port	L4-Proto	Src-IP/Dst-IP, srcPort/DstPort, VLAN Pri, Decr IP TTL Controller/Local	Goto Egress Table
Table Miss							Goto - Forwarding Table

D. Egress Table

Data is received in Egress table, this table parses and sets Source & Destination and it modifies or removes the VLAN ID based on the action, if the match is not there, drop the packet.

Table 4: Egress Table

Match Fields	
Data/ Upstream/ Downstream	ETH_SRC, ETH_DEST, VLANID VLAN-ID Remove / Modify Output Port
Table Miss	Drop

E. Forwarding Table

Used by network element for port forwarding and destination point filtering. Matching fields used are Src-IP, Dst-IP, etc. The action is either drop, trap to SDN controller or to local ports.

Table 5: Forwarding Table

Match Fields			Priority	Instructions	
Input_Port	Src-IP	Dst-IP	Priority	Action - Drop / Output <Controller/ Local >	Modify-data
Table Miss				Drop	

1. ARP Handling:

ARP handling is depicted with the tables in the figure, when an ARP packet comes to Ingress port, it is sent to L2 table. From L2 Table it is sent to controller and then it is forwarded and update the Egress table to update the Egress ports.

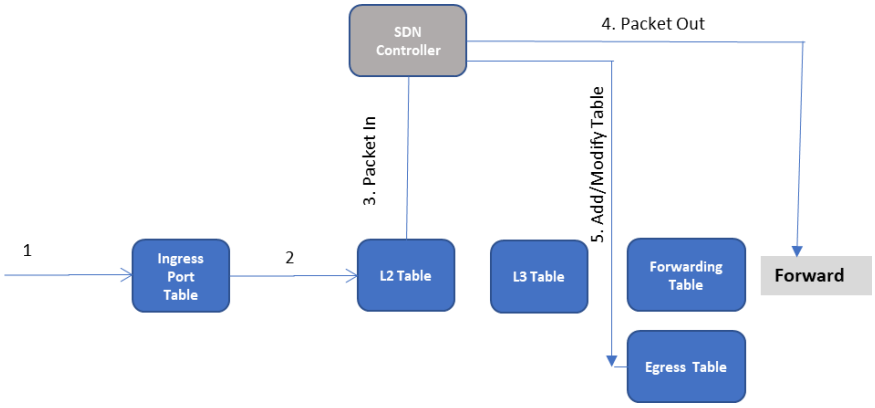

Figure 3: ARP Handling - Tables in Network Elements

2. Benefits of Migrating to SDN and Future Use

Following are the benefits of migrating the data center to SDN networks [9] [11] [5],

1. Cost effective
2. Provisioning and monitoring of the switch
3. Operation, Administration and Maintenance will be ease
4. High Data Traffic steering in Data centers

Similar to ARP handling, tables that are used for routing and multicast data traffic also to be depicted. SD WAN is a networking solution to connect the data center and client offices using SDN architecture. The configuration tables with minimal fields to deploy VPN to aggregate the data traffic remote data centers. Implementation of MPLS with these tables can be explored further.

CONCLUSION

SDN components and theory of working is also understood and explained in this paper. The forwarding of data traffic alone happens in the network elements with the sequence of flow tables depicted in this paper. ARP handling mechanism is depicted in this paper. Advantages and future work is also documented in this paper. We can achieve the efficiency, high sport speed, seamless data traffic within data centers.

REFERENCES

1. Nunes, Bruno AA, et al. "A survey of software-defined networking: Past, present, and future of programmable networks." Communications Surveys & Tutorials, IEEE 16.3 (2014): 1617-1634.
2. Manikandan Ramanathan, Kumar Narayanan, “Analysis of substantial growth on Data Center in virtualization local

area network applications", IEEE Xplore Part Number: CFP19P17-ART; ISBN:978-1-7281-2119-2
3. S. Paul, J. Pan, R. Jain, Architectures for the future networks and the next generation internet: A survey, Comput. Commun. 34 (1) (2011) 2–42.
4. M. Kind, F. Westphal, A. Gladisch, S. Topp, Splitarchitecture: Applying the software defined networking concept to carrier networks, in: World Telecommunications Congress (WTC), 2012, 2012, pp. 1–6.
5. Jammal, Manar, Taranpreet Singh, Abdallah Shami, Rasool Asal, Yimming Li. "Software Defined Networking: State of the Art And Research Challenges". Elsevier Computer networks 72(2014)74-98.
6. Wickboldt, Juliano Araujo, Wanderson Paim de Jesus, Pedro Heleno Isolani, Cristiano Bonato Both, Juergen Rochol, and Lisandro Zambenedetti Granville "Software-Defined Networking: Management Requirements and Challenges", IEEE Communications Magazine, January 2015
7. Paul Goransson and Chuck Black, —Software Defined Networks: A Comprehensive Approach‖, First Edition, Morgan Kaufmann, 2014
8. E. Haleplidis, S. Denazis, K. Pentikousis, J. H. Salim, O.Koufopavlou, Sdnlayers and architecture terminology, Tech. Rep. 01 (2013).
9. Thomas D. Nadeau, Ken Gray, —SDN: Software Defined Networks‖, O'Reilly Media, 2013.
10. J.R.W. Liu, J. Wang, A unified framework for software defined information-centric network, Tech. Rep. 00 (2013)
11. https://www.sdxcentral.com/
12. Scalable adaptive graphics environment. URL http://www. Open flow.org/wp/gui/
13. OpenFlow switch specification v1.0.0, [online] Available: http:// archive.openflow.org/documents/openflow-spec-vl.0.0.pdf