

SOFTWARE DEFINED NETWORKS-A WORKING PRINCIPLES, OPERATIONS BY USING BOTTOM-UP APPROACH AND ALTERNATE METHODS

Abstract

SDN is a developing architecture that is dynamic, manageable, cost effective and adaptable making it ideal for the high bandwidth, nature of today's applications. SDN that aims to improve the control of network and flexibility. The architecture of SDN decouples the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The chapter concentrates the working of SDN, requirements of SDN, the responsibilities of SDN and the components interaction. The techniques bottom-up was utilized in the SDN device, controller and the Applications. The device is constructed by the API structure for the interaction with the layer protocol and it constitutes with the data packets to inculcate the function. The flow tables are the fundamental data structures in an SDN device, which allow the device to evaluate incoming packets and take the action based on the content of the packet that has been received.

The section I contributes the functions availability of the open SDN. Section II focus the working and the alternate principles of the SDN. The Section III demonstrate the accomplishment of the SDN for the better replacement with 2 classification SDN via API's and SDN via Hypervisor-Based Overlay Networks.

This chapter defines the basement characteristics in which the SDN works in various platform. It is emergence to notice that there is no basic alternative choices to implement between the hypervisor-based overlay network approach to SDN and open

Authors

Dr. Ninu SB

Associate Professor
Thiruthangal Nadar College
Chennai, Tamil Nadu, India.

Dr. V. Devi

Principal
Thiruthangal Nadar College
Chennai, Tamil Nadu, India.

SDN. The final conclusion added in this chapter are the SDN act as complete assurance strategy solution given for the physical and virtual addressing issues.

Keywords: SDN, Plane Separation, Flow Tables, SDN Controller, SDN via APIs, Hypervisor Based Overlay Networks

I. FUNDAMENTAL CHARACTERISTICS OF SDN

The SDN is categorized by the various five basic traits (i) Separation of Plane (ii) a Device Simplification (iii) Control Centralization (iv) Virtual setup and automation of network & virtualization (v) Sincerity.

- 1. Separation of Plane:** The separation of the forwarding and control planes is the first fundamental characteristic of SDN. Based on the properties MAC address, IP Address and VLAN ID the logic and tables are used to deal with incoming packets. The action of forwarding plane describes how it dispenses with arriving packets. The actions may be forward, drop, consume or replicate an incoming packet. To forward the device obtain the correct output port by lookup in the address table of hardware ASIC. Due to buffer overflow conditions a packet may be dropped or due to filtering which result Qos rate – limiting function.

The logic and algorithm are applied to program the forwarding plane which reside in the control plane. To require the global knowledge of the network the protocols and algorithms are used. The control plane determines how the tables and logic in the data plane should be programmed. The complete devices on the network which focus the synchronized parameter utilize the dispense table forwarded in the network which avoid the prevention of loops.

- 2. Simple Device and Centralized Control:** The integrated system is governed by the shortening of devices which run under management and control software. The software is placed on a centralized controller that manage the network .The primitive operations are provided by the controller to the devices in order to allow them to adopt fast decisions about how to deal with incoming packets.
- 3. Network Automation and Virtualization:** The SDN can be derived precisely by the abstraction feature of distributed state, forwarding and configuration. Under the aspects of global network the shared state provide the programmer, by its unique state integrating to rectify the world wide network problems. The forwarding abstraction makes the programmer to specify the forwarding behaviors without any knowledge of vendor – specific hardware. Atlast the configuration abstraction, are also known as specification abstraction to express the goals of the entire network without losing the details of how the physical networks will implement those goals. Working with the network by using the configuration is network virtualization as its basic level.
- 4. Openness:** The standard was remaining constant in open SDN and it should be documented in worthy manner and nonproprietary. The APIs adopt the enough control to deal with and govern the different options of the plane control. Researchers can take advantage of this capability to test new ideas. Enormous growth of network, the individuals and organizations are applied themselves, focusing good and improved design and work involvement of networks.

II. SDN OPERATION (BOTTOM –UP APPROACH)

The SDN characteristics and the process of SDN is direct involvement in concept level. The basic components of SDN are (i) the SDN devices (ii) the Controller (iii) the applications. The bottom –up strategy is applied to understand the operation of SDN.

In fig 1.1, the SDN device includes the forwarding functionality to decide what to do with the incoming packet. The upper left portion of each device flows defined by the controller actually represent its data. A flow describes a set of packets from one endpoint to other endpoint. A flow is unidirectional, and it is nominated as data entry flow in a device entry.

If match occurs in a packet during the flow in the device the flow table which fit on the network device which includes the flow entries and output to perform the action. The SDN discuss with the packet for the match. It deliver the packet if it match else it drop or pass the controller.

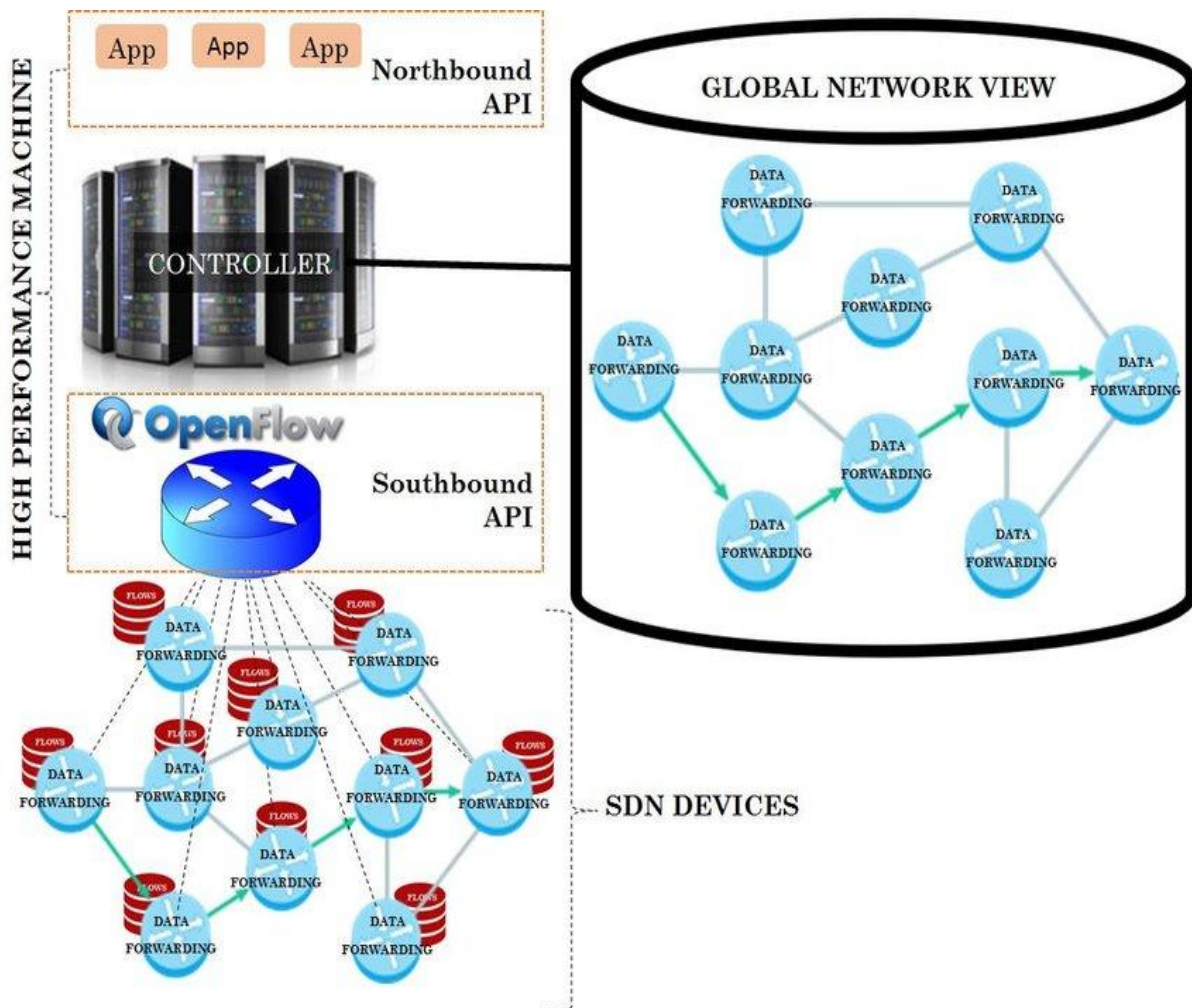


Figure 1: SDN Operation Overview

For the abstraction of SDN device the controller is in charge to the control of applications. The controller makes the application to respond the packets which are forwarded to the controller by the SDN devices. By predictable the controller estimates the optimal forward solution.

On the top of the controller the SDN applications are built. By the proactive flow the SDN interface and controller which implies to receive packets and has forwarded to controller. Proactive flow is also known as static flow will persist until some changes made in configuration. To establish the new flow on the device the Reactive flow will allow the device to respond to the packet.

III. SDN DEVICES

In fig 1.2, SDN Devices implies that the data packets delivery by the switch techniques which embedded in the hardware.

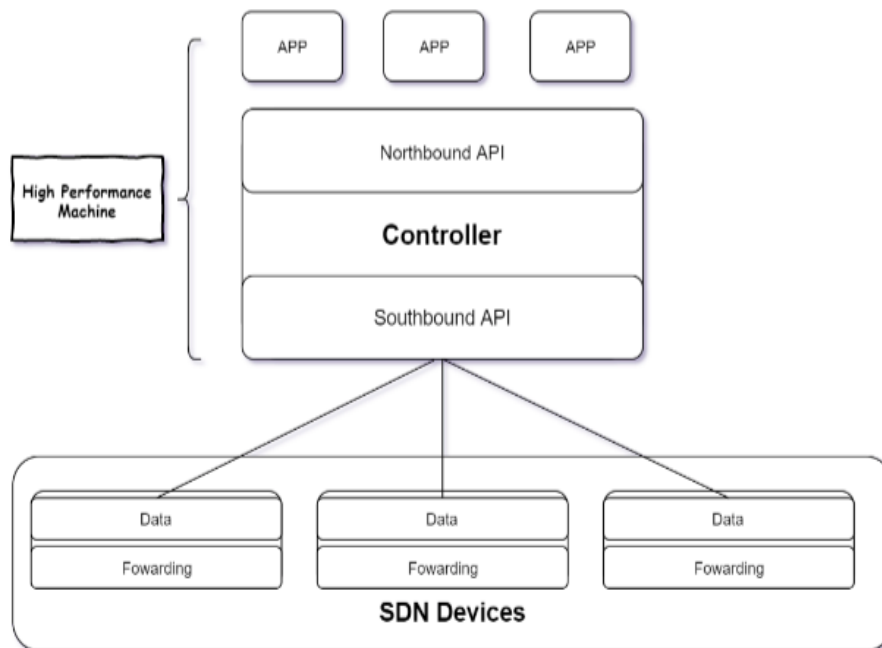


Figure 2: SDN Devices

The abstraction layer includes one or more flow tables. The mechanism which applied by the packet processing logic is to take actions based on the results of incoming packets and find the highest priority match. If match is found, the incoming packet is processed locally. If else, for alternate operations the data packets may be copied to the controller.

- 1. Flow Tables:** Flow table is a fundamental data structure which allows the device to evaluate incoming packets and placed the proper action. The action may be forward, dropping and flooding. It includes prioritized flow entries, which consist of two components (i) match fields (ii) actions. Based on priority the incoming packet compared with match field and the selection is made with proper action.

2. **SDN Software Switches:** The Hyper visions of the software oriented network include the SDN device. This Phase connects the different categories of digital machines to the digital network by the software switch. The SDN Software Switches is controlled by a centralized management system.
3. **Hardware SDN Devices:** For the better understand of the working principles of how this objects will translated into hardware ,we trace over the hardware equipments of network devices. The layer 2 and the layer 3 were implemented by the content-Addressable memories (CAMS) and Ternary content-Addressable memories (TCAMs).For the path finding (Routing) the layer 3 is used. It traces out the path table by the action with matching the Receiving IP address.

A series of challenges are focus here,

- The translation from flow to hardware entries.
- How to handle the flow entries in hardware.
- Dealing of hardware action limitations.

IV. SDN CONTROLLER

The SDN controller implements policy decisions, controls all the SDN devices that includes the network infrastructure .The modules involved in the Controllers are learning switch, a router, a firewall and a simple load balance.

1. SDN Controller Core Modules:

- **End-user Device:** End users devices are laptops, desktops, printer, Mobile etc.
- **Network Device:** The network devices are switches, routers, wireless access points etc.
- **Topology:** Concentrates on physically and logically laid out of the network devices.
- **Flow Management:** The synchronized is achieved at the database level.

2. **SDN Controller Interface:** For accessing the network the API act as key function which provide by SDN controller.

3. Issues in SDN Controller:

- Suffers towards the adaptation of new technology.
- Demanding network will be high in real-life applications.
- Require more heterogeneous mix of equipment types.
- The limitations make the coordinator to risk factor by the lack of standard Northbound API and flow prioritization.

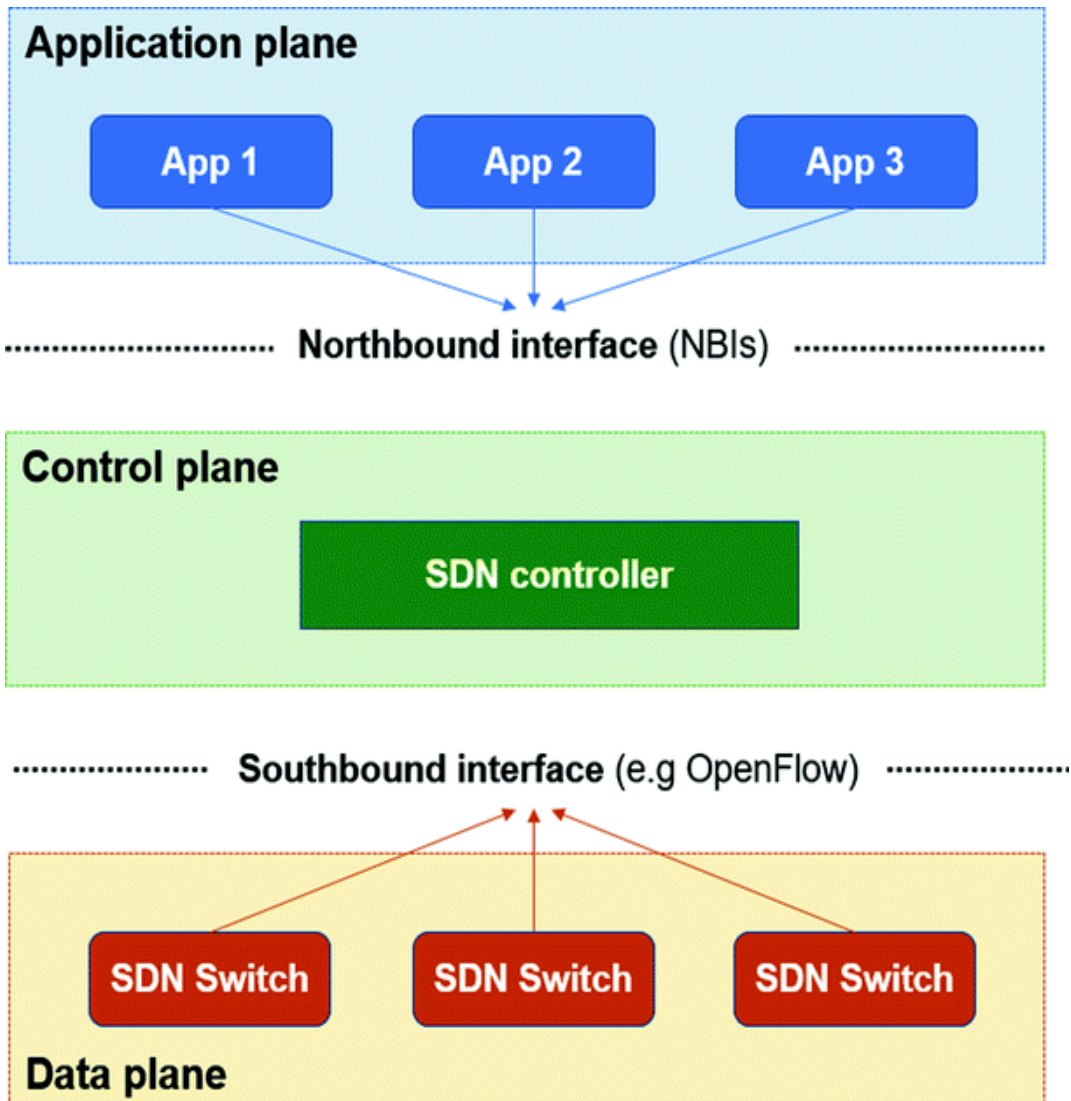


Figure 3: SDN Controllers

In fig 1.3, the events of the network were intimated by the controller. For the operation of the network the application uses different methods.

V. SDN APPLICATIONS

Above of SDN Controller there is an SDN applications act as intermediate to the network by the northbound API. To operate the flow entries the high respond which focus on SDB applications. By using this Application able to,

- Find the best path and configure the flows to route packets between the endpoints.
- For Multiple paths, it balances the traffic.
- React to the dynamic topology.
- Redirect the traffic.

Responsibilities:

- To perform appropriate function.
- Application spends its processing time to respond the events.
- Affects the network by responding the events.
- The method will be invoked when event occurs.
- The SDN applications capture the Incoming packet events.

VI. ALTERNATE SDN METHODS

The two alternate SDN methods are (i) SDN via APIs (ii) SDN via Hypervisor-Based Overlay Networks.

1. SDN via APIs: The various levels of APIs which included for the SDN applications are,

- SDN via Device APIs
- SDN via Controller APIs
- SDN via Policy APIs
- **SDN via Device APIs:** The SDN give the intelligent and predictable pattern by the centralized software which concentrate on the devices by the control points.

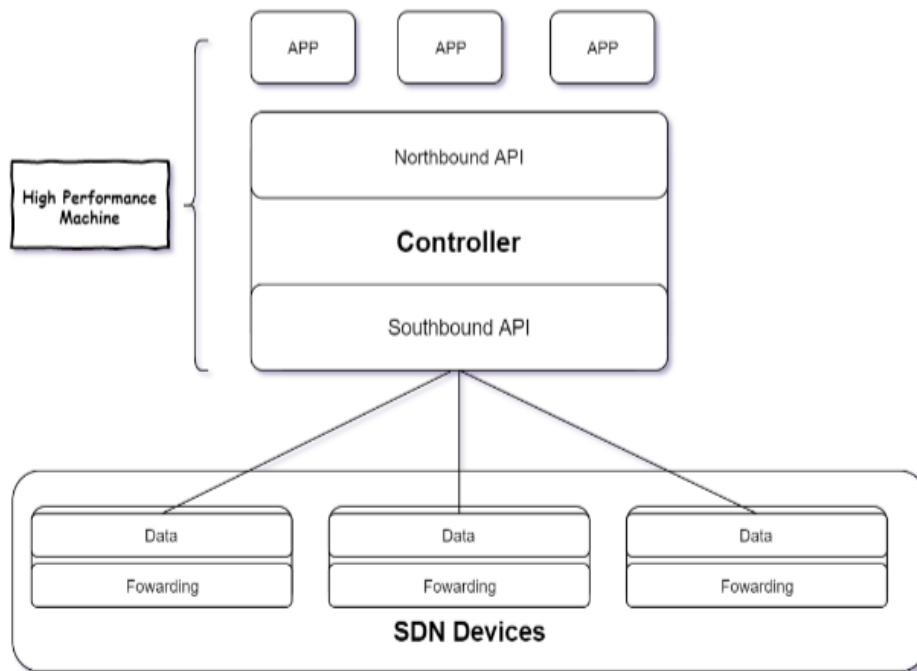


Figure 4: Device APIs

The Figure 4 transparent the Device APIs which use the centralized applications. Interface to the routing system (I2RS) provides an interface between routing protocols and the Routing Information Base (RIB).The protocols were divided into four key drivers,

- The necessity of an intermediate interface for atomic process.
 - To maintain the access of routing information and configuration protocols.
 - To provide the network maintenance and to subscribe filtered event notifications from the routing table.
 - To provide flexibility and configure the standard data-models to be used by network applications.
- **SDN via Controller APIs:** For the construction of SDN applications it acts as best platform. These SDN via Controller APIs act as public source and open for the application developers.

To maintain the functions the availability of protocols are highly engaged. It acts as level of abstraction in between the device and the application. The major role is to perform the collection of dynamic topology data in the virtual network by the controller. In other cases, the message communication which focuses on the applications and the device. The best protocol NETCONF is used for the configuration.

- **SDN via Policy APIs:** The API is act as topper layer in the controller level. The structure of the network act as informative rather than imperative.
 - (i)Imperative: Get the input for how to do a job.
 - (ii)Informative: Get the actual data for what to be done.

2. Benefits:

- No up gradation in switches.
- Adopt the dynamic changes and ease of use.
- Assurance to the SDN Solution in the devices.
- High open in nature.

Limitations:

- When the absence of controller the programmer make an interaction directly to the switches.
- The controller never gives an abstract view to the programmer.
- The top level applications must be synchronized at the distributed level.

3. **SDN via Hypervisor-Based Overlay Networks:** The edge network make connection with the virtual network which collect the information of the physical that in turn connect to overlays. The high traffic of data signal focused on the top of the layer. The hypervisors inject traffic in bi directional format. The Tunnel endpoint administrates the overall control connection.

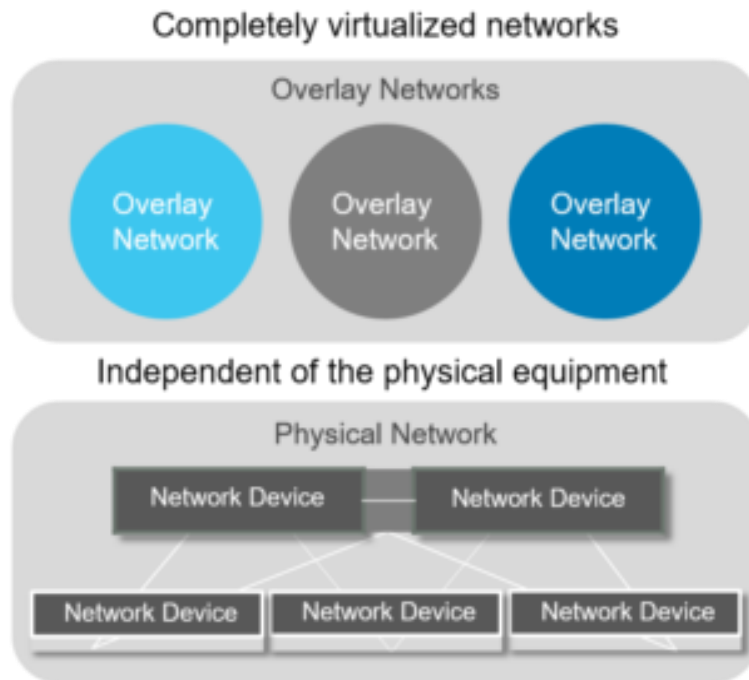


Figure 5: Hyper based Relay networks

By the data collected by the controller the hypervisor adopt the encapsulated data packet which given to the receivers TP. The decapsulation process engaged in TP and send this to the receiver host. This whole process is informed as MAC-in-IP tunneling.

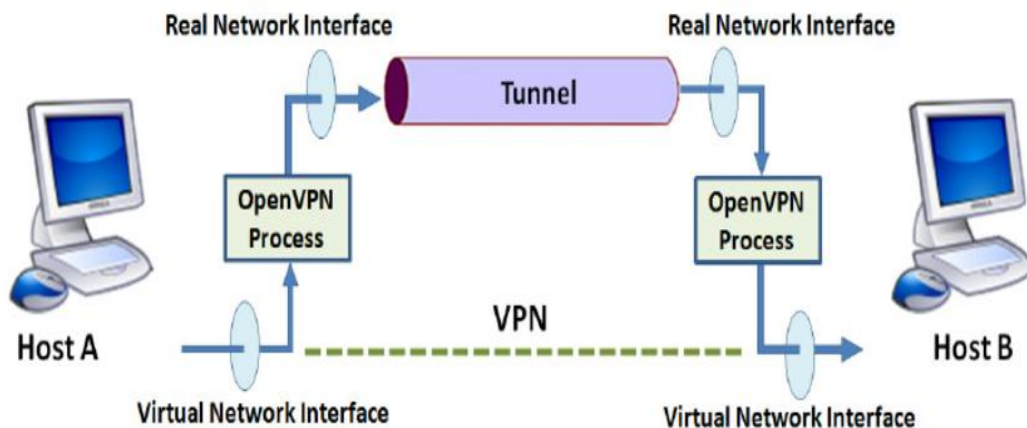


Figure 6: Tunneling

Fig1.6 transparent the overall work nature of TP. The flexibility is adopted by the hypervisor by the virtual switch. The transmission mode involved in this network is dynamic topology point-to-point link configuration.

In overall chapter summary, SDN via Hypervisor-Based Overlay Networks is highly adopted to the virtual background with already engaged with defined storage

software. Given preference to address the MAC explosion in repository centre and server storage environments. Next priority is to address the limitations of the VLAN for the data handling. The least priority given to the Third, it addresses the emergence of the availability, since it is demonstrated in the software environment with the dynamic feature with the adaptation of the network infrastructure.

VII. CONCLUSION

This chapter entitled fundamental concepts which related to the SDN solution. Given neat clarity for the understand of the various approaches which adopted in SDN and open SDN. To the demand of virtual connection use the open flow to make use of the tunnels. It is not reasonable to think of these overlay networks as stepping stones toward a more complete SDN solution which includes SDN and Openflow for addressing both the virtual as well as the physical needs of the network.

REFERENCES

- [1] Shenker S. The future of networking, and the past of protocols. Open networking summit. oct 2019.
- [2] Kerner Brocade Doubles Down on Open SDN. Enterprise Networking Planet. May 23 2016.
- [3] Open flow management and configuration protocol 2015
- [4] Wang Floodlight documentation .project floodlight Dec 2013.
- [5] Nox. Retrieved from: <http://www.norepo.org>
- [6] Open daylight technical overview .<http://www.opendaylight.org>
- [7] Ruby: a programmer best friend: <http://www.ruby-lang.org>
- [8] Learn rest: <http://www.restapitutorial.com>
- [9] Trema: Openflow framework in ruby: <http://trema.github.io/trema>