Enabling Bandwidth and Low Latency in Data Centres Using Open Flow

Rajanish Dadi

Submitted as part of the requirements for the degree
of MSc in Cloud Computing
at the School of Computing,
National College of Ireland
Dublin, Ireland.

September 2013

Supervisor Mr. Vikas Sahni

Abstract

The demand of Network Virtualization and Cloud computing is rapidly increasing day by day. The Software Defined Networking with the Openflow protocol mainly used in the network topology. In traditional networking system, due to the lack of bandwidth we cant access the services properly. However, the bandwidth and latency of the nodes are major problems. The efficiency and performance of the nodes are not utilized properly in the traditional system.

Till now, there is very small quantitative work is done on the measurement of the latency and bandwidth using the Openflow. So, this topic is selected for my thesis work. Here we will calculate the bandwidth and latency using Openflow. This will also help us to enable the dynamic path between the nodes. The experiment will tell about how to utilise the bandwidth in the data centres.

The experiment has been done by implementing the fat tree topology using the Openflow. In the first part, the allocation of Host (nodes) and open v switches in the topology by using the Openflow protocol. And in the second part the measuring of the bandwidth and latency with the help of the Iperf and Mininet tools. Analysis of the results tells that, with Openflow we can utilise the bandwidth more efficiently. And also it is observed that the dynamically allocation of nodes can be created by using Openflow. These results tell about the implementation of Openflow in the data centres can minimise the bandwidth and latency between the nodes.

Keywords: Network Virtualization, Software Defined Networking, Openflow Protocol, Bandwidth, Latency.

Acknowledgements

I would like to thank my supervisor Mr. Vikas Sahni sincerely for his expert advice, guidance, direction, and feedback throughout the project. Your support and patience, encouraged me to work more than my potentiality and to finish project successfully. I was really blessed to get a supervisor like you.

I thank our Cloud Computing Programme Director Dr. Horacio Gonzalez Velez, for his enormous support throughout the year.

I would like to thank to all my batch mates and well-wishers.I also thank to my friends Sandeep Pathivada and Dinesh Medipally for helping me in my thesis.

Last but not least, I thank to my mother, Lakshmi Balla, my father Hanumanth Rao Dadi and my brothers Ranjith Dadi, Revanth Dadi for their love and everlasting support.

Declaration

I confirm that the work contained in this MSc project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.

Signed ..... Date .....

Rajanish Dadi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview:

The Network Virtualization is has been from many years. The concept of the Openflow made the network Virtualization more efficiently use for the service providers. The implementation of the traditional topology in the data centres are not well utilised. There are some drawbacks like low utilisation of nodes, high bandwidth consumption. As day by day, the networking technology has to replace in the proper manner for the Quality of Service (QOS). The bandwidth consumption is major problem in traditional data centres. For each application we need to allocate a certain bandwidth to the nodes. As load increases then the performance will leads to decreases. Here we will outline the experiment to test the bandwidth and latency of the nodes using the Openflow. In the first part the configuring of the Open v-Switch in the Virtual Machines is takes place. In next part the Openflow controller manages and controls the nodes.

The implementation of the Openflow by using the tunnelling protocol is explained [42]. Here they used only tunnelling protocol. From there work we observed that the utilisation of the nodes for dynamic path allocation is taken in account. And also from the published work on the Multimedia trafficking of the nodes using the Openflow is explained [45]. There mainly focus on the migration of the Multimedia packets to other VM by utilising the Bandwidth. From the above experiments we extended our work to calculate the latency and bandwidth by using different benchmarking tools. Implementation of Openflow in the LTE network system for Evolved Packet Core is mainly processed by the Open v-switch and is explained in the research paper [21]. The main motto of our experiment is to calculate both bandwidth and Latency by utilising the Open v-switch efficiently for Quantitative results.

1.2 Problem:

The Openflow protocol and Mininet tool is used mainly for the research purpose. As till know a small quantitative work is reviewed and published on the bandwidth and Latency measurements using Openflow. This experiment is mostly important for the users who want to know the performance of the Openflow by using the Mininet tool. Here we will collaborate with the combination of various hosts and switches. The Openflow controller will control the nodes. And it will explain in various chapters.

1.3 Challenge:

The major challenge is to evaluate and analyse the outputs of the experiment by using the Openflow. The implementation is takes place by using the different hardware and various tools. The main key factors of this experiment are the VMs CPU performance, Network connection between the nodes and the load factor on the processor while execution of test. Generally, the process of implementation of test takes place without the load factor on the nodes.

For measuring of bandwidth and latency in various systems by using different tools is really very difficult. So, by considering this statement we conduct the same process of experiment for many times. We are not depending upon the single test results. For accurate results we need to perform many tests. And average the values of many outputs and finalise the accurate values.

1.4 Objectives:

The experiment mainly concentrates on the measuring of the latency and bandwidth of the nodes using the Openflow Protocol. For this we need to create the virtual open switch, Hosts and switches on the topology by using the Mininet tool. The objective of the experiment is starting the test by building a virtual network in a VM i.e. by using the Openflow protocol in the Mininet tool.Design the topology for the number of hosts and switches.

The controller has to interconnect with the switches and Hosts through Open V switch. Then perform the Ping test between the hosts and note down the latency values between the hosts. And also conduct the bandwidth test in the hosts by using the Iperf tool.

Finally, evaluate the results for better output. The objective is mainly based on the time factor and network availability between the nodes.

1.5 Scope of the work:

Mainly the scope of the work is concentrate on the controller performance. And how the controller is correlated with the hosts and switches for free flow of network. The packets are rerouted to the hosts and switches through the controller. Here we use the Ubuntu 13.04 operating system for the execution of the experiment. And also the test mainly depends on the hardware components like number of Processors, Memory, Hard disk and Network connection between the nodes. And also some of the tools like the VMware workstation, Mininet tool and Iperf tool for the execution of the experiment.

1.6 Brief Methodology:

For execution of the experiment we derived the Openflow switch called as Open v-switch. And in the first part the open v-switch is tied up with the open flow controller. Here the controller acts as the centralised system to control all the nodes and switches. The topology of the network is tells about the depth and the number of host and switches that can directly connected to the upper layer switches. The Switches are connected with the hosts. And the each number of hosts i.e. depends upon the fan-out configuration of the Open v-switch.

These hosts are interconnected with the switches. Each switch will carry the packets information to the controller through a connection between the controller and switches. The measure of the bandwidth and latency in the open v switch mainly depends upon the controller performance and hardware components like processors, Network measurements. Firstly, Install the Mininet tool in the Linux Operating system for topology purpose. The topology will contain the controller, Open v switch, hosts and switches.

The step by step execution of the flow of packets from host to other host will be measured. Here time has been taken as reference for calculation of Ping test between the hosts. The total execution time for a single host to the multiple is calculated. And also the total cumulative bandwidth is measured for all the hosts. Each host ping the adjacent host for certain time. These calculations are measured and recorded for various times of process. And finally evaluate the results for accurate values.

1.7 Potential Contribution:

This experiment can give us an overview about the calculation of latency and bandwidth of the open flow switch. The controller part is the major part in the contribution. As we use the general open source controller for our experiment. The design architecture of the controllers is different with other controllers. The values of the outputs will vary if we change the controller type but the actual scenario will not change. Therefore, the results of the experiment tell about the open flow switch characteristics for minimising the bandwidth and latency in the data centres.

1.8 Dissertation work:

The thesis document mainly covers about the brief overview of the experiment in coming chapters. From the chapter 2 it covers about the research background of the experiment. The introduction of the Software Defined Networking and Openflow controller is explained. And also about the types of controllers and merits of the SDN with Openflow protocol.

From Chapter 3 it mainly covers about the different works done by the various people in this experiment. From the Chapter 4 it tells about the design architecture, tools and also about the hardware and software system requirements.

The Chapter 5 tells about the implementation of the experiment and the process of the experiment. The execution steps like procedure and process of the Openflow switch are explained. From those steps we can measure the outputs values and finalise the accurate values. Finally the Chapter 6 tells about the evaluation of results and it finalises the conclusion of the experiment.

Chapter 2

Background

2.1 Introduction:

The Research Background tells about the background structure of the dissertation. It gives an overview of the Network Virtualization, Software Defined Networking (SDN), Open Flow protocol (OFP) and Open vSwitch. And we will see the merits and demerits of SDN using Open flow protocol. And, also we will look how network virtualization is implemented in the Cloud Data Centres by using SDN.

2.2 Software Defined Networking (SDN) and Openflow Protocol (OF):

SDN is also named as Programmable Network architecture where it decouples the Control plane and data plane using Openflow protocol. In traditional network architecture both control plane and Data plane are coupled together. The Data centres are generally connected with the tree like topology. Here the end hosts are interconnected with the top of the rack switches in the upper layer. And it looks like a leave of the tree. The core switch is form of the root and is interconnected with the leaves like end switches. In the general tree topology all the switches has the parent switch except the core switch. There is no interconnection between the switches. In general tree topology there is lack of the inter rack bandwidth and fault tolerance. In traditional data centres all servers are linked with the IP address only. Due to the VLAN configuration and physical location the IP's are assigned. Therefore if we want to move servers to other location we need to use the same IP subnet.

In order to minimise the bandwidth and fault tolerance the Fat Tree topology is introduced in SDN-Open flow Protocol. In fat tree topology all the switches are interconnected with the other switches in the network. There is no end leafs of the tree all will acts as roots of tree. Each switch acts as a core switch. Here from the below figure we represents the Hosts with the squares and the Switches with the circles. We can resolve this issue using the SDN open flow by allocating the IP's to the servers whether they are in IP subnet or not.

In SDN there are two protocols one is open flow and other is ForCES(Forwarding and Control Element Separation).Openflow is used due to the high capability of adding the new forwarding plane whereas ForCES didn't high capability for forwarding plane.ForCES has dynamic model and it makes the protocol stronger. The main difference is Openflow protocol is open source model whereas ForCES is not an Open source Model. Due to these reasons ForCES lack in the market.

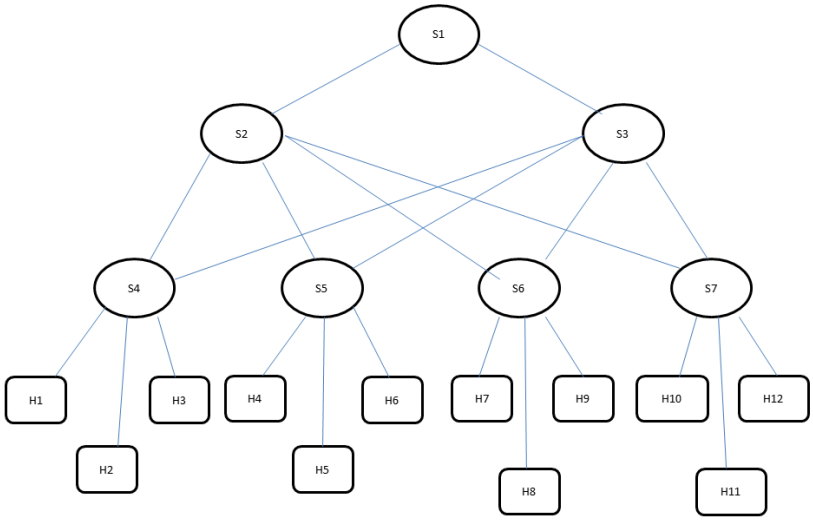

Figure 2.1: General fat Tree Topology

Software Defined Networking and Openflow allows operators to control the network using software running on a network operating system within an external controller and also provides the maximum flexibility for the operator to control a network. Using Open API's(Application Programmable Interface) SDN Openflow provides access to network services. These services included the security, routing, access control and bandwidth management. Due to the simple centralized architecture it is easily manageable. We can see the difference between the general fat tree topology without open flow and the fat tree topology with open flow in the following figures.

2.2.1 Why we use the SDN?

The major benefits in the centralised controller are Easy maintenance and up gradations, Flexible nature for expansion of network, reduce the maintenance cost of the network. Lastly, it is applicable for both large and small networks.

SDN can fit in infrastructure as a service (Iaas) or Platform as a service (Paas). The open flow is standard protocol for creating virtual network. Using SDN, we achieve scalability, flexibility and performance. Here traditional network switches are replaced with virtual LANs (VLANs).

Using SDN, we can reduce the work load of management of network. SDN is a network control frame work that helps operators in all aspects. In SDN, two bound Interfaces and they are south bound Interfaces and north bound Interfaces. South bound refers the layer between programmable switches and north bound refers to external and network policies. Open flow SDN comes under the southbound.

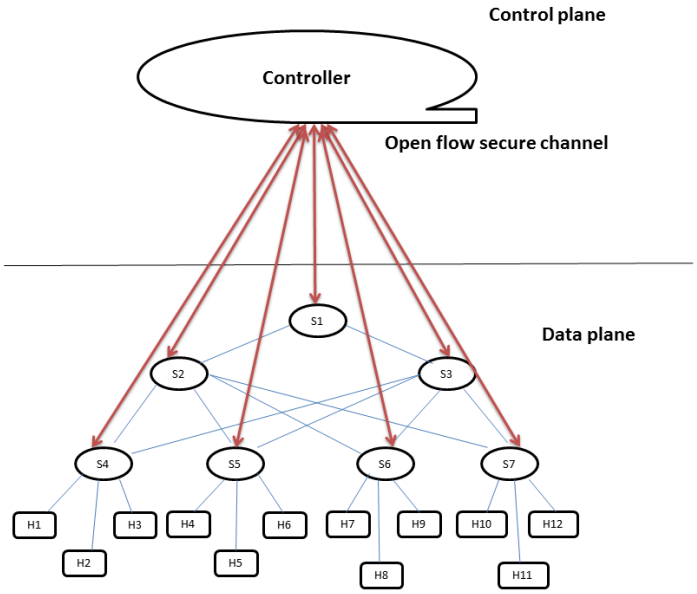

Figure 2.2: Fat Tree Topology with Openflow Controller

2.3 Open Flow Protocol:

The open networking foundation (ONF) establishes the open flow protocol. The Open flow is the standard designed for SDN. It is mainly defined between the data plane and control plane. Packet forwarding and the network policy is translated by network

controller.

The open flow mainly separates the control plane from the data plane. The open flow switches uses open flow protocol to connect the network controller. In SDN, packet forwarding is completely taken over to the controller. The Openflow mainly allows the header fields in the Layer 2, Layer 3 and Layer 4.

In layer 2 the protocols such as the VLAN, MAC address and Spanning Tree Protocol (STP) and in the Layer 3 the protocols such as Interior Gateway Protocol (IGP) and Exterior Gateway Protocol (EGP) will takes place remotely.

There are two modes such as active mode and the passive mode. In active mode, depend upon the flow table the controller forwards the traffic. And in the passive mode if any packet is not matched with the flow table then it reroutes the packet from other route.

The authentication of VLAN domains are happened in the network. Using the network policies and firewalls the configuration of VLANs will takes place. The Open flow mainly consists of the flow table and it performs the packet lookup and forwards through secure channel by an external controller. The flow table have the headers data.

2.4 Open Virtual Switch (Open vSwitch):

The open vSwitch is designed based on the Apache 2.0. The establishment of the open vSwitch is to enable the Cloud computing implements and Open flow protocol. The main features are the IPsec tunnelling, port bonding and VM trafficking policing. In virtual switch, it is mainly classified in two paths, one fast path and second slow path. In fast path processing of packets, encapsulation in VLAN packet, packet forwarding and trafficking is happen. In slow path, the control and configuration of open virtual switch. The open virtual switch manages the elements in the kernel and maintains the flow table to execute the packets.

Open vSwitch is a part of the Linux kernel. Mainly the Open vSwitch divides into the kernel space and the User space. All data forwarding will takes place in the kernel space and it is called as data path. The delivery of the packets to the data path and data path generates the flow key. The flow key contains all the packet information i.e. sender and receiver information. The same flow key of the packets will process in the kernel space. For configuring of Open vSwitch the network device (netdev) layer is used. For communication between the data path and control layer the data path interface (dpif) is used.

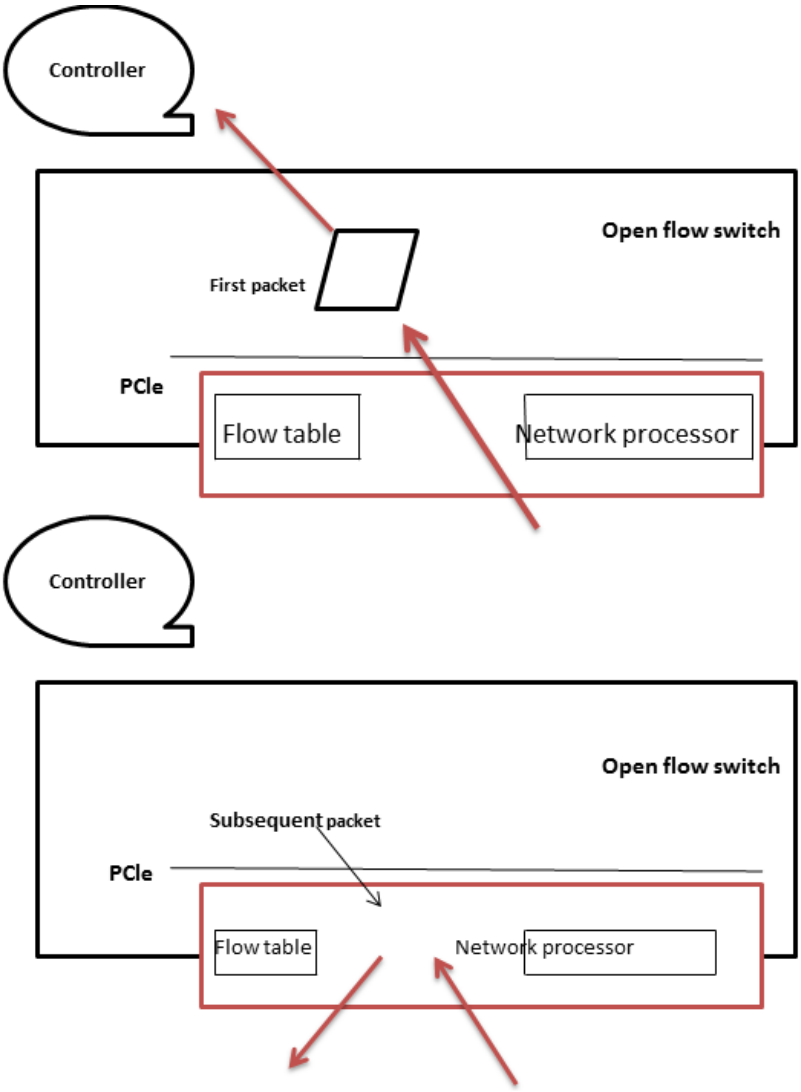

Figure 2.3: Open Flow Switching with a Network Processor based NIC

The virtual switch is interconnected with RAM and CPU in physical machine. The rate of transmission increases the load factor of virtual switch varies. Using virtual switching at the NICs, we can increase the host CPU utilization that can concentrate on directly to VMs. The better processing of packet switching when compare with host CPU. Also we can increase the data centre security with the NICs virtual switching.Open V- switch consists of two types module virtual supervisor module (VSM) and other virtual Ethernet module (VEM).VSM generally performs management functions and VEM performs switching.

In a network of Openflow enabled switches the path of the packets can be determined by Openflow controller. The open flow enables switches are two types one is

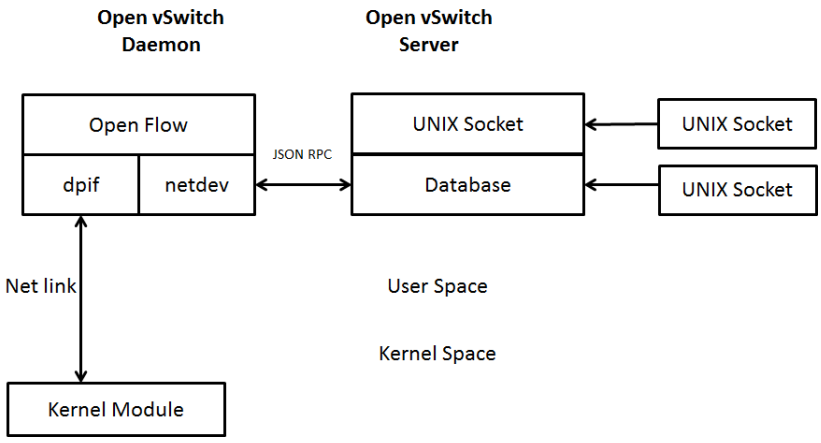

Figure 2.4: Open vSwitch Protocol Stack

Openflow-only and Openflow-hybrid. In Openflow-only, they only support Openflow operations whereas Openflow-hybrid switches allows both Openflow operation and Ethernet switching.

In Openflow there are two modes of switching takes place. One is TCAMs (Ternary Content Addressable Memory) and other is Open flow switch which is software based. The main task of Open flow switch is to maintain the Flow Table and Securing the channel between controller and nodes.

The Openflow switch maintains the flow table of the packets. If any new flow entry is found then its forward to the controller. Then controller will route the packet through shortest path. In flow entries there are two crucial time outs. Hard time out indicates the maximum life of flow entries, regardless of activity. Idle time out indicates the time when flow entries should be removed due to lack of activity. It is time interval of flow entry with which open flow switch has not received the packet of a particular flow of that entry. UPC (unified control plane) is the open flow based unified control plane which is used to control optical switching networks.

The open flow nodes update the flow table by adding or deleting the requests using FIB (Forwarding Information Base) in hardware using TCAMs. The flow table update the requests using header of packets. If any request is happened, the nodes firstly check flow table (Packet IN message). If no match is found in flow table, then it forward request to the controller in secured way (Packet OUT message).

The secured layer between open flow switch and controller is called the Transport Layer Security channel (TLS).In open flow, the forwarding of packets is done according to the forwarding table list. The controller can update, deleting and manage the forwarding table. This flow is not assigned in the basis of IP but also in the headers in the packet.

The first packet always forward to the controller to assign a new route and it is updated in the flow table. If any node is failed then the controller assigned a new header to the packet to restore the network.

In	VLAN	Ethernet			IP			TCP	
Port	ID	SA	DA	Type	SA	DA	Proto	Src	Dst

Figure 2.5: Packet Header

The open flow packet header length is 64 bits. And they are break up as 8 bit type, 8 bit version, 16 bits of message length and 32 bit transaction id. In VPN network there are two types of layers they are layer 2 (L2 VPN) and Layer 3(L3 VPN). In layer 3 we use the Routers and in layer 2 we use switches. Both are designed for IP/MPLS+BGP core. In Layer 1 the connection is established and it is designed for TDM(Time Division Multiplexing)

Openflow is mainly used in the research environment for conducting the experiments. And Openflow needs to improve the virtualized computing for resource allocation Due to this issue it is unable to handle Openflow equipment. The main benefits of SDN Open flow can be seen mainly in the adoption of three technologies like cloud computing, virtualization and large data centres. SDN improves the network process speed. And also it reduces the network cost and improves the network architecture. Due to the centralized controller, we can tie up with the low cost routers and switches. In this way we can minimise the maintenance expenditure of the Network.

2.5 Network Virtualization:

Network Virtualization is a technology that allows the system to create the virtual network partition in Physical infrastructure. In other words, slicing of the bandwidth into the number of channels by using resources and they are independent to one another. And each source is named as the node. The Nodes are mapped by using the virtual network topology. The network virtualization allows the multiple virtual networks to share the same infrastructure.

The Network Virtualization is divided into three components called Access control, Path isolation and Virtual services. In Access control all authentications will takes place i.e. authenticate clients only can access the network. In Path isolation, the creation of logical paths in a physical network system takes place i.e. maintaining the logical traffic path. And the Virtual services are for the sharing of network services. The

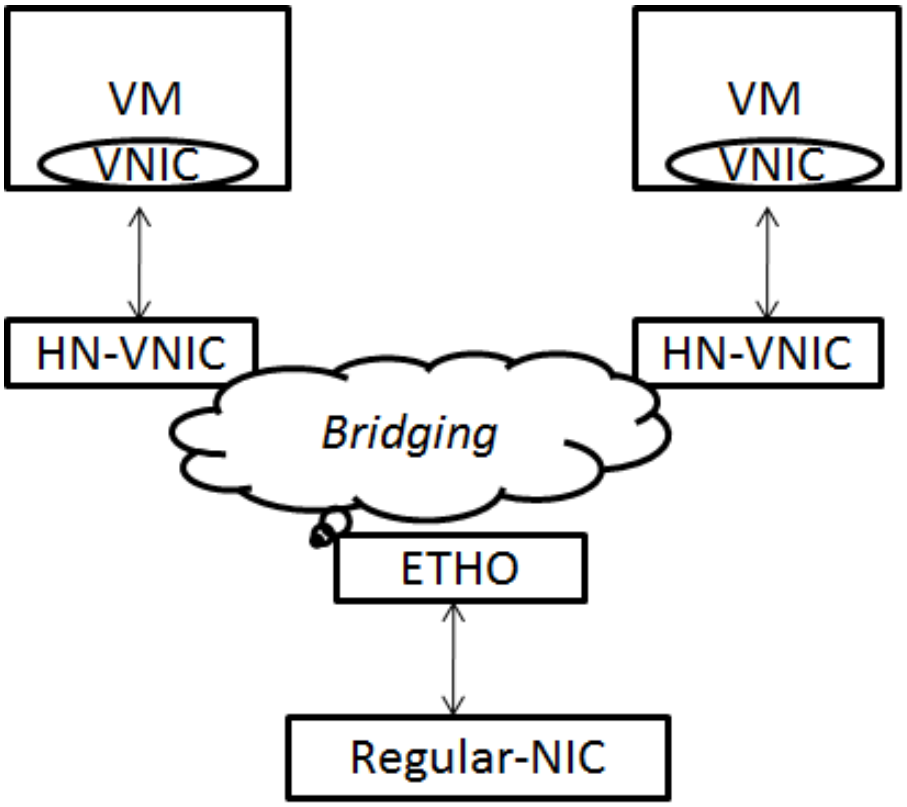

Figure 2.6: Virtual NICs bridging for Virtual Machines

technologies VLAN (Virtual Local Area Network) and VPN (Virtual Private Network) are based on virtualization.

Virtualization provides the centralized management for the users. And other advantages of virtualization are low cost and more power reduction. In virtualization utilising of the resource takes place. The main characteristics of network virtualization are deformation of resources and trafficking of resources, capability of dynamic allocation and flexible mechanism, end users can easily operate virtual infrastructure in various domains.

A network bridge is useful to interconnect two Ethernet ports. The network interface card (NIC) is connected with Network Bridge to communicate with the other hosts. Using bridges, we can transfer the packets between senders to receiver. Bridge is also useful for shaping of traffic and filter. Every Internet packet is attached with headers, which can be characterizing with parity bits (1s and 0s). In every header, we can get the destiny address. This type of transferring the packets is called layer 2 transparent packet forwarding VNIC acts as NIC for virtual machine, so it interconnects with the physical NIC card.

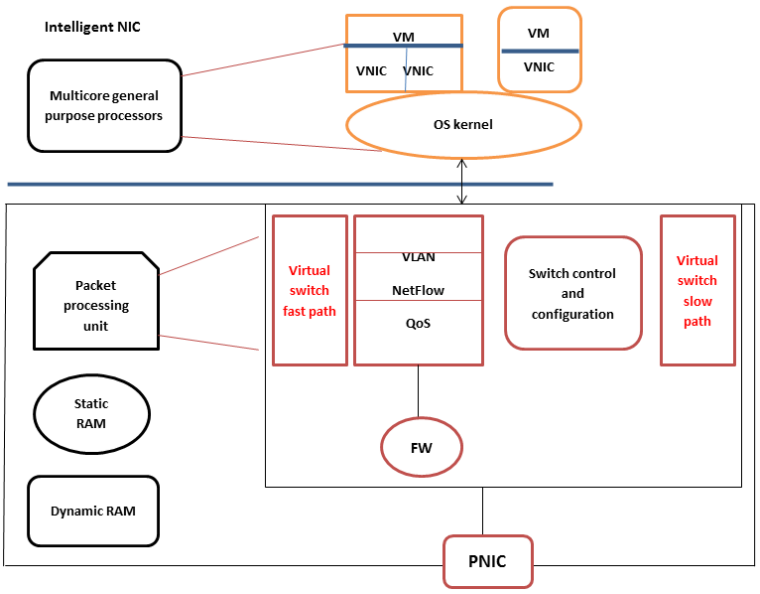

Figure 2.7: Network Bridge of VMs

A virtual machine can easily communicate with many virtual NICs (VNICs) using bridge network. Now by using network virtualization, NIC bonding can increase the reliability. In Network virtualization, NICs can control resource sharing using single root I/O virtualization (SR-IOV) [17]

In cloud data centres both physical machines and virtual machines play an important role in delivering of applications. The cloud data centres are grouped with large number of servers, firewalls, storage, clusters and nodes. So, to communicate with each other effectively, we need highly network configuration. Using virtualization, the operators can minimise the operational cost.

Limitations of using network virtualization.

1. Firstly, allocating of IP's to the all VNICs, VMs and other switches in the network.
2. Secondly, the delivering of VMs packets is based on the headers destination. It is difficult to deliver the packet to exact VMs in the VLAN network.
3. And lastly, there is increase of workloads on the CPUs due to the large usage of bandwidth.

2.6 Conclusion:

In the conclusion it is to conclude that, a complete over view of the Software Defined Networking, Openflow Protocol and Network Virtualization including the drawbacks is explained in the background. Difference between traditional network architecture and SDN Openflow architecture is mentioned above. The difference types of controllers, OSPF (Open shortest Path First) protocol and MPLS(Multi-Protocol Label Switching) will be explained in the literature review chapter.

Chapter 3

Literature Review

3.1 Introduction:

Introduction of Literature review tells about the clear structure from the above background. It mainly concentrates on the reviewing of various works and experiments are taken in account. The following sections in the literature review will be as follows section 1 tells about the different types of controllers, section 2 tells about the difference between Openflow protocols to the MPLS protocol and section 3 tells about the OSPF protocol for latency measurement and section 4 tells about the Mininet tool.

3.2 Openflow Protocol:

The Thesis paper by the [11]tells about the deploying of the LTE (Long Term Evaluation) mobile networks using SDN-Openflow. The experiment tells about the extension of the open v-Switch in the Mobile networks using the GTP protocol is will increase the availability and flexibility of the network. From the paper, it helps to calculate the UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) throughput by using the SDN in the LTE system. This experiment mainly implement in the transparent phase of the evolution. And the LTE system will be more benefit able with the combination of SDN and Cloud computing.

Firstly the research method starts with the contribution of the GPRS Tunnelling Protocol user plane (GTP-U) and the GPRS Tunnelling Protocol control plane (GTP-C) in the data centres. In Data centres gateway, the implementation is depends on the virtual switch (open v-switch). The GTP protocol is the main protocol in the EPC (Evolved

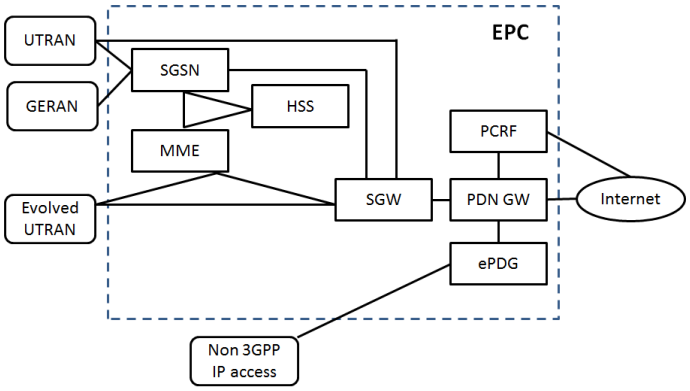

Figure 3.1: Evolved Packet Core of Openflow

Packet Core). [4]The main function of the GTP is the Mobility management, tunnel management and user data interface. The first task of the GTP-U is the tunnelling and encapsulation of the user's IP packets.

And the other paper by the [21]tells about the Route aggregation in SDN. In this paper the author concentrated on the existing network architecture to the SDN architecture. And he found some limitations in the existing architecture are the vendor dependence, high network complexity, inability to easily scale the network and inconsistent network policies. These all limitations are overcome by using the SDN architecture.

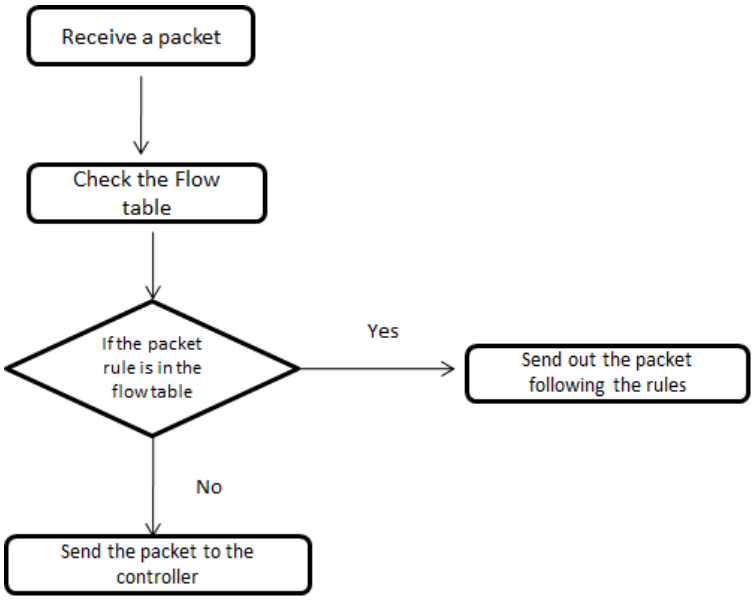

Figure 3.2: Flow chart detailing packet flow through an open flow switch

Generally the network is characterised into Production network and Research network. In routing perspective they are two types of routing called fast packet forwarding and

high level routing decision. In fast packet forwarding the routing is done by the data path whereas the high level routing is done in the control path. [1]The OSPF (Open Shortest Path First) protocol is implemented in the controller for the shortest path between the nodes. From this experiment we can learn that the open flow switching technology in the layer 2 and layer 3 is more flexible and more accurate than the legacy network. So, the open flow is replacement of the IP routing and Ethernet switching.

The research paper proposed by the Stanford University, California tells about the packet and Circuit network convergence with Openflow system. Here they proposed the Unified Openflow enabled packet and circuit network architecture [3]. And also they explain the benefits of the packet and circuit switching. The main concept of the Openflow Unified architecture having three systems like network operating system, network applications that run on top of the network operating system and Openflow protocol that makes the access of the control and data plane.[10]

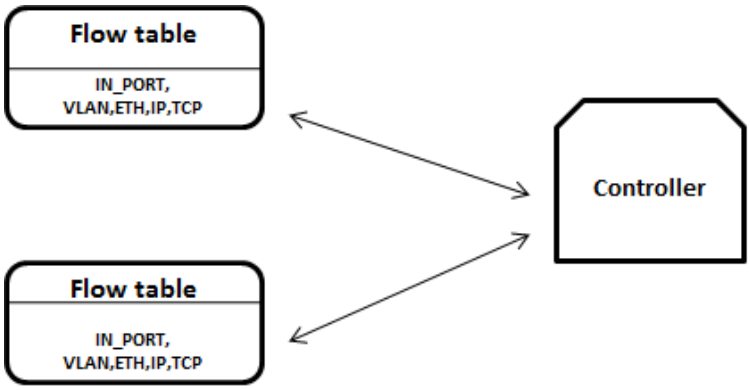

Figure 3.3: Openflow Architecture

Using Openflow we can minimise the cost and improves energy efficiency of the networks [23]. And also by using the Openflow we can increase the fast failure recovery .The research paper on the Enabling fast failure recovery in openflow networks[22]. Here they conduct the experiment between the different nodes using openflow. And they observed the fast restoration is able to switch to other nodes within 12ms interval only. Open flow is an open platform i.e. the operator can manage the controller as per there design [15]. By using flow manager in open flow controller we can manages the routes for specific load balancing

3.3 Openflow Controllers

:

Typically in the conventional network architecture both control and data planes are controlled by switches or routers. But in the Openflow the control and data plane are controlled by controller. The controller is the crucial element in openflow. There are different types of controllers are in action using openflow protocol.

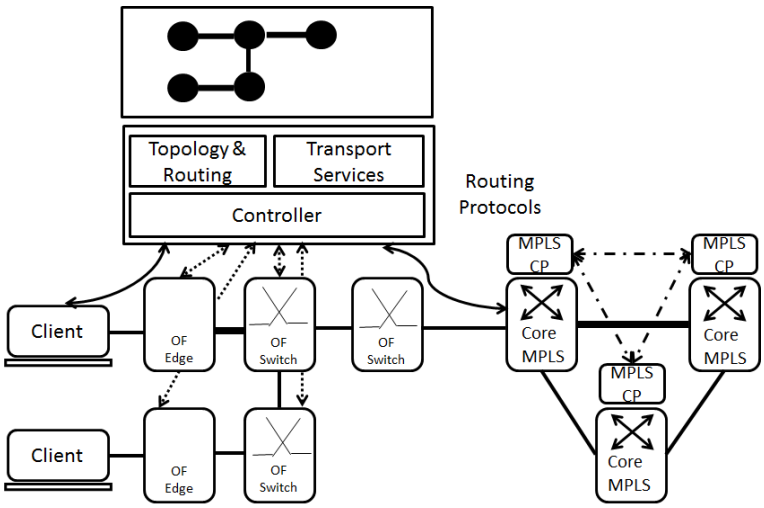

Figure 3.4: Openflow Network Topology

Depend upon the programmable languages, the Java based controllers are the Maestro, Jaxon, Floodlight and Beacon. The controller Trema is based on Ruby and C language. And the controller NOX is based on the C++ and Phyton[18]. The evaluation of different controllers and the switch is been emulated by Mininet tool (and we go through this Mininet tool in coming section).And the Phyton based controllers are the POX, Ryu. The other controllers based on the C language are MUL, Helios, Ovs-controller and Flowvisor[15]. Here the author view is to evaluate the performance of the different controllers.

The performance of the openflow controllers are depending upon the network design of the controllers centralised architecture and also the flow based commands[20]. The aim of this dissertation is to measure the Latency and bandwidth of nodes by using openflow network. And it mainly depends upon the controller behaviour i.e. throughput and latency mode. Depend upon the study on the openflow controllers the suitable and simple open source controller is NOX controller.

3.4 NOX Controller

:

NOX is the first openflow controller and it allows developers to program their software program with C++ or python, using a set of application programming interfaces (APIs) to interact with open flow-capable switches.

The NOX controller can operate in the two modes called proactive and reactive modes. In proactive mode all the new packets can process to the controller. And reactive modes the controller will save the packets information and sends to respective nodes.

The Research paper on the openflow based unified control plane using NOX controller tells about the optical networks using the openflow in different switching environment. By using NOX controller he tested the dynamic cross layer path[12] and also the restoration.In NOX controller the L2 (Layer 2) learning switch creates a mapping in between MAC and physical ports of switch.

3.5 GMPLS (Generalised Multi-Protocol Label Switching) vs. OF/SDN

:

[7]In traditional system the GMPLS uses the MPLS control plane for the UCP Architecture (Unified control plane). But in the Openflow we use the common switch API for distributed network system. The thesis paper by the [5]on the Openflow networks tells about the new architecture i.e. QOS (Quality of Service) architecture. By using openflow they introduce the concept of dynamically optimizing QOS routes. Author concludes that the Constrained Shortest path (CSP)[16] is taken in consideration for QOS of nodes. And also he mentioned the difference between the OSPF and CSP protocols for shortest path routing.

The white paper on the PCE (Path Computation Element) using the GMPLS network tells about the limitations of the GMPLS over the SDN- openflow [40].And also the architecture of SDN-OF and GMPLS systems are explained. [14]Mainly the limitations of the GMPLS are the Multi- layer path computation and Path computation blocking.

The RSVP (Resource Reservation Protocol)[6] is responsible for the both data and control plane in the GMPLS. And in the SDN-OF the Openflow protocol will takes the control of the system.

The Research paper by the[2] implemented the combination of both GMPLS and Openflow network connectivity in the optical transport system. The performance and feasibility of hybrid GMPLS- OF are tested with the GMPLS system. With this experiment

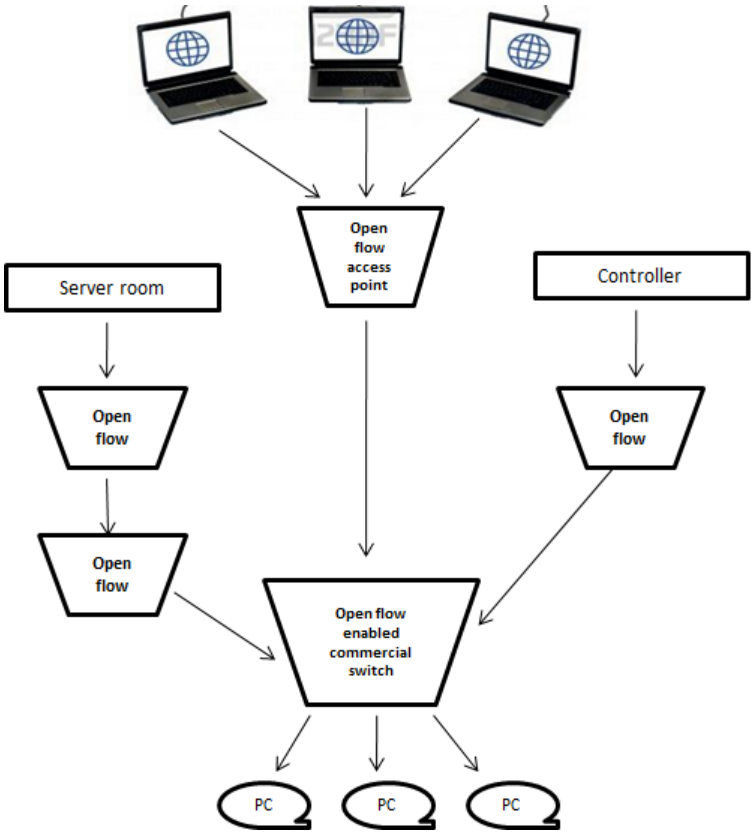

Figure 3.5: Openflow Enabled Switch in Real world

he concludes that the Openflow based SDN can provides the extensible frame work for the nodes in a network. [19]And also the other research paper on the Sharing of hybrid networks tells about the proposed networks sharing and easily manageable by the users for any applications and Services by using SDN-OF.

The conference paper on the Formal modelling and verification of SDN-Openflow tells about the logically describes as formula i.e. the dead lock Specifications (spec) is measured by the Monitoring process (Mp) and System model (Msys) i.e. Spec = Msys/Mp [9].The NCP (Network Copy) system demonstrates the replication that enables the scalable service in real time using minimal overloads by using the Openflow controllers[13].

3.6 OSPF Protocol(Open Shortest Path First

:

The OSPF is called as link state protocol. It was developed by IETF (Internet Engineering Task force). [1]The OSPF uses the Dijkstra algorithm for the shortest path.

The algorithm mainly works on the calculation of the nodes CPU intensive. The main formula for calculation of n number of nodes is the n log n. The OSPF is a protocol which helps the open flow controller to communicate with other routers in shortest path.

The OSPF sends the link information to the adjacent nodes by link state advertisement (LSAs). In the LSAs the IP address of each node is noted in the flow table.[21] In OSPF the selection of route is depend upon the cost metric. The each interface has the particular cost. So, the OSPF calculate the shortest path depend upon the source and destination. The bandwidth of the nodes will take in consideration while routing of the packets. Maintain the substantial topology and minimize the bandwidth.

In the process of packet transmission parallel the Address Resolution Protocol (ARP) will allocate the store to source MAC address of destiny address. To stop the ARP traffic between server and client, the end host won't send the ARP request with broadcast address. Using routing mode we can build the short path between end nodes. The shortest paths always track by the routing mechanism.

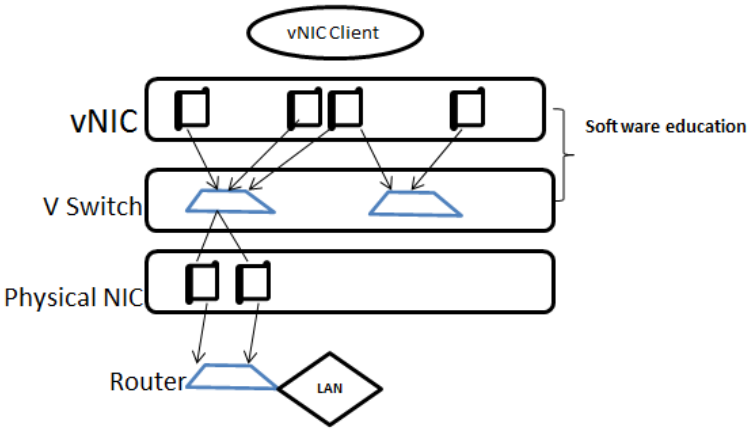

Figure 3.6: NIC Virtualization Architecture

3.7 Mininet Tool

: Mininet tool is a network emulator and it helps to create a realistic virtual network in a real kernel. Mininet is written in Python language. And it allows the all python based scripts to interface in it. Using Mininet tool by the author Satyanarayana in his research on the Software Defined Network Defence he successfully completed the experiment [20]. And his main motto is to test the entire network consists of the hosts, Openflow and controllers in a kernel. He uses the NOX controller in Mininet tool

because of both functions with the Python scripts.

It helps to interact and customize the SDN prototype with the network[18] .It can easily share and deploy the open flow protocol in real hardware for smooth path of network. It creates the virtual switches and application code on physical or virtual machine using the commands. And it collects the all the switches, routers and links in a single Linux kernel. It makes the single system to look like complete network by using the lightweight virtualization.

Mininet creates the bridge network connection between the hosts. Mininet tool mainly helps in the research and development domain[8]. The Mininet VM contains the Open flow binaries and they interact with the Linux kernel configuration to support networks. The advantages of Mininet are we can create the topologies like single switch and data centre. We can easily customize the line rate packet forwarding using open flow protocol. We can run and create the experiments by writing simple python scripts. By using Mininet tool we will measure the latency between the nodes i.e. low latency, Number of Hops covered Status

3.8 Conclusion:

Here all the various controllers in Openflow are mentioned detailed. And from the other thesis papers we can take some specifications like how to measure the latency and bandwidth. And finally we select the appropriate tools like Mininet and NOX controllers for design and implementation of experiment.

Chapter 4

Design

4.1 Introduction:

The design tells about the requirements of the experiment. The software and hardware system requirements are helpful to setup the test. The tools discussed from the above topic are needed to be design for the execution of experiment. Here the tools will discuss about the testing of the both bandwidth and Latency measurements in the Openflow protocol. The tools place a major role for design and execution of the results.

4.2 Hardware Requirements:

The research design experiment is mainly depends upon the decoupling of the data and control planes. In Design it tells about the architecture of the Mininet tool for calculating the bandwidth and latency of the Data centre Networks.

4.2.1 Physical hardware:

The Hardware configurations details are explained below for the design of the experiment are as follows. These entire Hardware configurations are existed in my server (laptop). The implementing of the experiment is done by considering these hardware configurations.

1. Processor: Intel core i5-3317U CPU@ 1.70GHz
2. NIC card: Qualcomm Atheros AR8162/8166/8168 PCI-E Fast Ethernet Controller (NDIS 6.30)

3. Core speed: 2394 MHz
4. RAM: 6-8 GB
5. System Type : 64 bit Operating system, X64- based processor
6. Hard Drive : 500GB Serial ATA (5400 RPM) with 32GB MSATA Card
7. Memory : 6144MB (1600Mhz)
8. Ubuntu 13.04 Linux operating System (64bit)
9. VM ware work station (9.0) or VM ware Player
10. Mininet tool kit
11. Iperf tool

4.2.2 Virtual hardware:

The Virtual Machine (VM) is prepared for the launch of the Ubuntu 13.04 on it. For implementing the VM the hardware are as follows.

1. Disk Space : 20 GB
2. Disk Provision: Thin
3. Memory: 2GB
4. vCPUs: 2
5. NIC: 1

4.3 Tools:

4.3.1 VMware Workstation

It helps to create the multiple VMs (Virtual Machines) in a same physical hardware. We can install and execute the Operating system in each VM i.e. one VM workstation can easily operates many number of Operating systems in a single physical machine. For the each VM we need to allocate specifications like Memory, Processors, Hard disk space and Network adapter. We can down load the VMware workstation 9.0 as an ISO file from the www.vmware.com

4.3.2 Mininet Tool

With Mininet, we can simulate multi host network and multi switch. It creates the virtual open flow network and controller, switches, hosts and links in a real or virtual machine. The Mininet tool will work in different operating system such as Mac OS, windows and Linux. Mostly for Research part Linux is preferable. Using Mininet, we can create the small data centre consists of hosts and open flow switches. By implementing this experiment, we can achieve the output.

Mininet mainly allows the hosts, switches and controllers and it creates the huge network simulation in a single PC. The Mininet VM can be downloaded from the http://mininet.org/ and it need to set-up on the top of VMware

4.3.3 Iperf tool:

Iperf is a performance measuring tool for the TCP (Transport Control Protocol) and UDP (User Datagram Protocol) band width. Mainly Iperf is used for the bandwidth and datagram loss. It mainly uses to calculate the network flow between the two nodes. And it can be downloaded from the iperf.fr

4.4 Design architecture:

Mainly the architecture is implemented on the physical hardware. Firstly deploy the VMware workstation on the physical hardware. And after create a new VM (virtual Machine) on the VMware workstation. Download the Ubuntu 13.04 and install on the new VM. Here the Ubuntu acts as Guest Operating System. On the top of the VM the guest operating system will run (without any Hypervisor).

After the completion of Ubuntu VM then download the Mininet in the Ubuntu. The Mininet tool is installed in the Ubuntu 13.04. The Guest operating system didn't have any Hypervisor. The Mininet tool helps to create the network topology in the Virtual system. The topology mainly consists of host, switches and Open v switch. From the Mininet tool the Openflow protocol is generated. The Controller in the Openflow Protocol takes the control of the hosts and switches by means of flow tables. The latency is measured by using the Ping test between the hosts and the bandwidth is measured by using the Iperf tool.

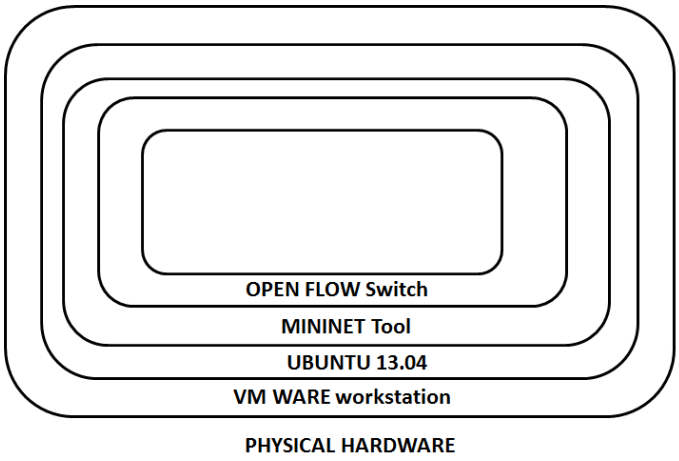

Figure 4.1: Design Architecture

4.5 Conclusion:

In conclusion it mainly tells about the basic design for the experiment. And also it tells about the hardware and software requirements. The benchmark tools and software's installations and the implementations will be explained in the next chapter for the accurate outputs.

Chapter 5

Implementation

5.1 Introduction:

In this chapter it mainly tells about the implementation of the experiment with the designed benchmark tools. Here we will use the Mininet, VM workstation, and Iperf. The implementation will be divided into various streams. The Implementation takes part with the Installations of VMware and it follows with the Mininet VM setup. And also we will see the setup of some other tools like Iperf command functions. And all this experiment will take place in the Ubuntu Platform. The implementation of the experiment is for the proper outputs. The process of experiment is repeated many times for accurate values.

5.2 Installing the VMware Workstation:

For the installation of VMware workstation the following steps need to be followed.

1. After download the VM ware workstation in form of ISO file.
2. Run the ISO file for setup
3. It shows Typical and custom type. Select custom type and click next.
4. Install the setup in the destination folder and click next.
5. Then it performs the requested operations.
6. After the requested operations then click next

7. Then it will ask for the Licensing Key. Enter the Licensing key..
8. After licensing key it will ask for the Licensing agreement.Accept the licensing agreement.
9. Then click OK and it creates the VMware workstation on specified path.

Now the setup of VMware workstation is completed in physical machine. After that, create the new Virtual Machine (VM) in the VM ware workstation.

5.3 Installing the Ubuntu VM in the VMware workstation:

1. For Guest Operating system we need to download the Ubuntu 64 bit 13.04 iso image.
2. After the set-up of the VMware workstation.
3. Now click the File and New virtual Machine
4. After that select the Custom type and click next.
5. For Guest Operating System installations browse the disc image (iso file) from the source and click next.
6. Create the User name and Password for personalize Linux and click next
7. After that name the Virtual machine and click next
8. Allocate the Memory disk size as per recommended and select the store virtual disk as a single file, then click next
9. Then it shows the customization of hardware. Check if everything is as per recommended and select the power on this virtual machine after creation.
10. Select Finish and it will create the new Ubuntu VM in the VMware workstation.

5.4 Installing the Mininet tool in the Ubuntu:

1. Download the VM disk image file of 2GB (.vmdk).
2. Open the vmdk file in the VMware workstation and power on the button to start the VM.

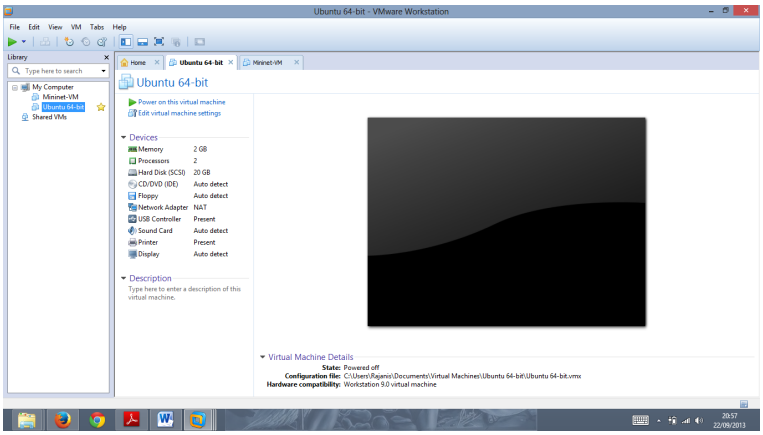

Figure 5.1: Ubuntu VM in workstation

3. In the boot prompt, we need to set up the user name and password as open flow.
4. The VM set the access between the host and guest using NAT connection
5. Need to check the IP configuration of Ethernet port 1 by using command

```
1 ifconfig eth1
```

```
1  sudodhclient eth1
```

6. Create the two different IPs one is for the host network adapter of the physical machine. And other IP has to assign to the host network adapter in the VM.
7. Generally the Mininet is starts up with the general configuration i.e. one switch (s1) and two hosts (h2 and h3).
8. Check the ping test in the general configuration by using command

```
1  h2 ping h3
```

9. If the ping test works properly then open the terminal in the Ubuntu for adding the extra switch and host configurations.
10. Create the network by using the command. Here we need to select the number of hosts and switches depend upon the fanout. And depth tells about the number of switches are interconnected with the hosts.

```
1 ~$ sudo mn --switch ovsk --controller ovsc --topo tree,depth=2,fanout=4
```

11. As per our test the depth 2 and the fanout 4 (i.e. 16 hosts and 5 switches). After the hosts and switches are created. Then measure the ping status between the hosts.

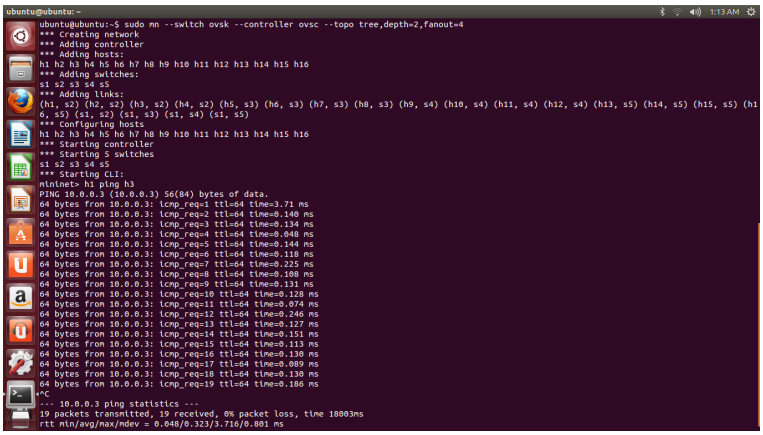

Figure 5.2: Openflow Switches and Hosts configuration

12. We can create as many hosts and switches depend upon the memory capacity of open vswitch. For every host the memory is dynamically allocated.

13. To check the interconnectivity between the hosts and switches just add the test ping

```
~$ sudo mn --switch ovsk --controller ovsc --topo tree,depth=2,fanout=4 test ping
```

14. This interconnection tells about the latency measurements between the each adjacent hosts and switches.

15. If the controller will not in active state and shows error like controller is running. We need to use the command to stop the controller.

```
~$ sudo service openvswitch-controller stop
```

16. By using dpctl command we can view the switch port and flow status. The dpctl is data path control.

17. It helps to add, remove, and modify rules in the flow table.

18. If we need to see the status of the data path control of the controller. For this we need to run the ping command between the hosts. And the dpctl command is

```
~$ sudo ovs-dpctl show
```

19. Then we can view the number of the switch ports in the open vswitch which are interconnected with Ethernet ports.

20. If we want to see the packets flows of the switches then the command is

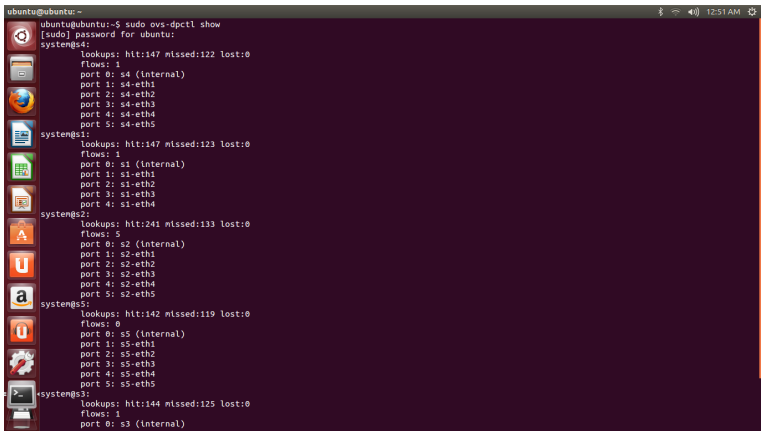

Figure 5.3: Dpctl show

```
~$ sudo ovs-dpctl dump-flows S2
```

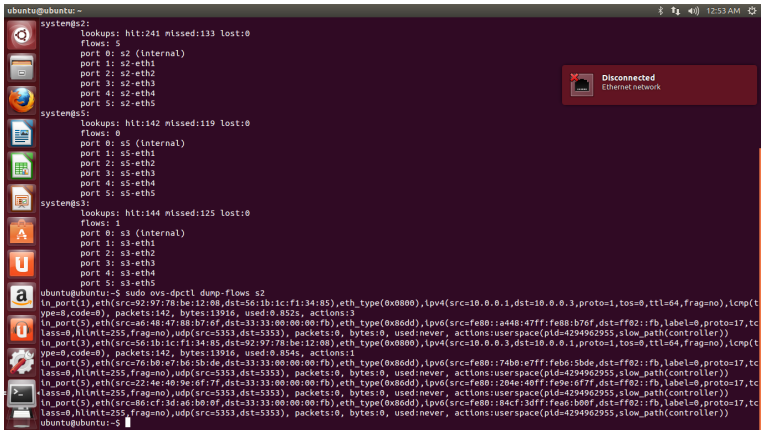

Figure 5.4: Dpctl dump flows

21. After the dpctl then we need to verify the open vswitch Openflow control of the switch and the command is

```
~$ sudo ovs-ofctl show S2
```

22. The data flows of the openvswitch can view by command

```
~$ sudo ovs-ofctl dump-flows S2
```

23. The number of packets mainly consists of the time factor with memory status of the flowtable is presented.

24. By using the Openflow we can minimise the dynamic path also. And the allocation of dynamic path of the hosts are as follows by using the command

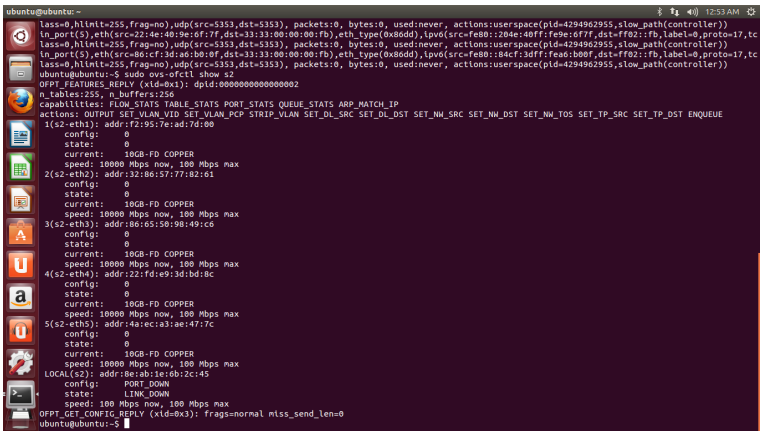

Figure 5.5: ofctl show

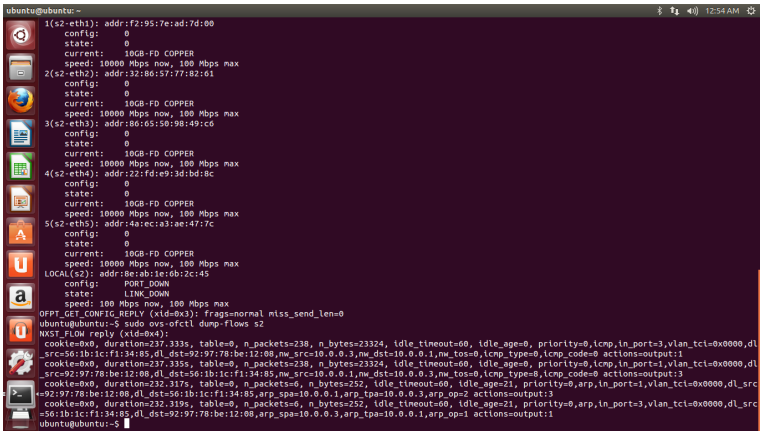

Figure 5.6: ofctl dump flows

```
~$ sudo ovs-ofctl add <s1> <flow>
```

25. Here the flow represents the path of the packets thats need to deliver to the end node. The dynamic path allocation is mainly depends on the flow tables of the switches.

26. For the graphical representation of the hosts status we need to install the consoles command.

```
/Desktop sudo-/consoles.pyw>
```

27. The consoles tell about the performance and ping test automatically represents in bar graph.

28. The Iperf tool is integrated in the consoles. This helps to check the performance of the hosts.

Repeat the test between the combination of different hosts and switches. And note down the time of execution values. Make sure that all hosts have to communicate with each other. For every time of the test calculate the average values of the hosts. For the latency and Bandwidth measurements compare the cumulative results of the hosts with the individual hosts.

5.5 Conclusion:

The Implementation of the designed planned is executed by using the benchmarking tools. Here we conduct the experiment many times and note down the results. The standard deviation of the results is to calculate and needs to evaluate those values for Evaluation of process. The evaluation of the values will takes place in the next chapter.

Chapter 6

Evaluation

6.1 Introduction

The outputs and values are taken in account from the above chapter. And we will evaluate the results and represent in the graphical form. The latency and bandwidth consumption of the hosts using Openflow is clearly explained in this chapter. From each test we note down the results and average those values for the appropriate results. Here we represent with the bar graphs.

6.2 Results

6.2.1 Bandwidth

The bandwidth is measured from the help of the Iperf tool. the calculations and values which are noted in the above chapter is taken into account. From the below figure we can see the bandwidth values of the hosts.Here each host acts are in active state i.e. the utilization of all hosts.we can see the data rate of the hosts varying from the host to host we observe that the maximum bandwidth of the 62 Mbytes of hosth5

By considering the three to four experiments and average the bandwidth results and we plotted the graphs below for host h1 to h16

From the graph we can make a conclude that the utilization of the bandwidth is more while using Openflow switch. The graphs tells about the all hosts are in active mode then the utilization of bandwidth is more than 4 Gbps. It will help to increase the API in efficient manner.

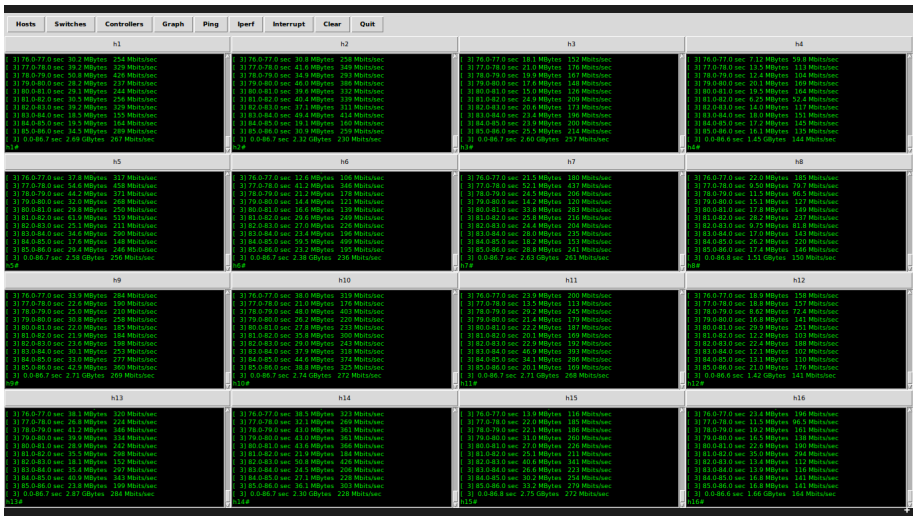

Figure 6.1: Iperf test results

6.2.2 Latency

The Latency is measured with the help of the ping command.The calculations and values which are noted in the above chapter is taken into account. From the below figure we can see the Latency values while performing the Ping test of the hosts.Here each host acts are in active state i.e. all hosts ping to the adjacent hosts for the interconnectivity. The results are taken in account and plot the graphs.

By considering the three to four experiments and average the latency values and we plotted the graphs below for host h1 to h16

From the graph we can make a conclude that the hosts are in active state and they will fall back if there is no flow of packets are not pass from it. For every flow table the certain time is noted for erasing of flow values. The graphs tells about the all hosts are in active mode and hosts will always in active mood.

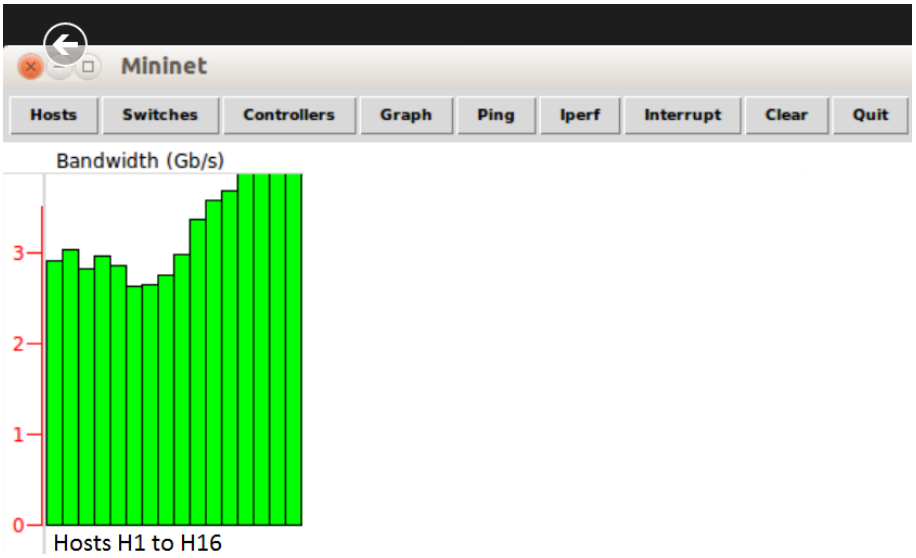

Figure 6.2: Bandwidth measure graph

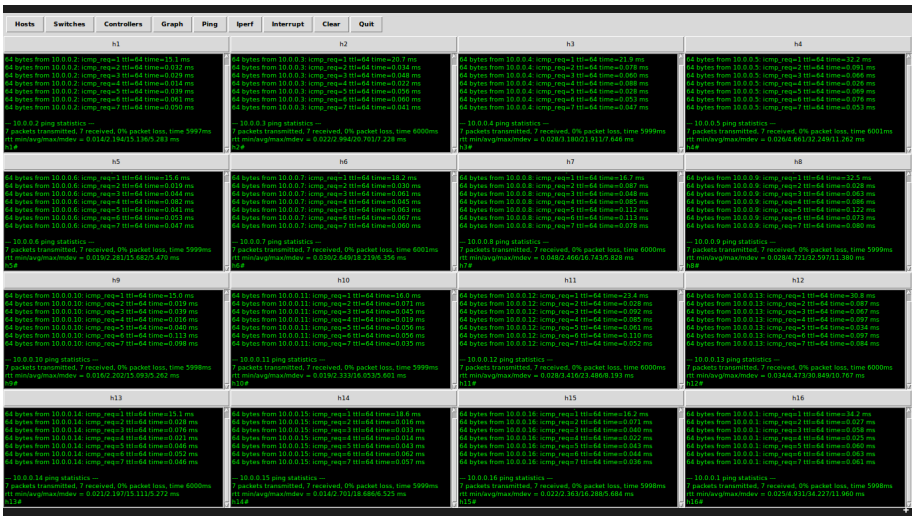

Figure 6.3: Ping test results

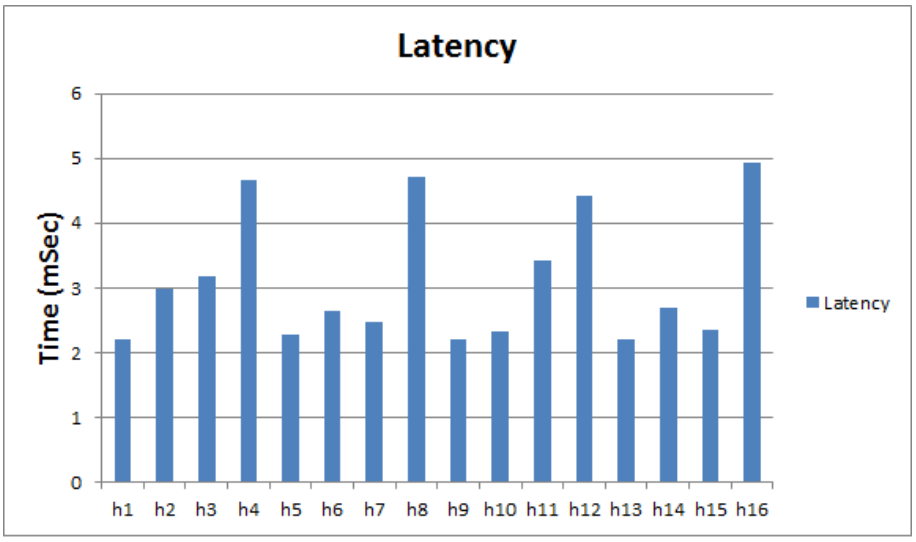

Figure 6.4: Latency measure graph

Chapter 7

Conclusion

7.1 Overview:

Therefore, the main aim of this research is to calculate the latency and bandwidth of the active hosts in the data centres using the Openflow. Here the latency is measured between the hosts by using the ping test . This concept of open flow has to implement in data centre to minimise the time and increase the efficiency. By using the SDN over flow protocol in network Virtualization we can improve the network efficiently .The use of SDN over flow is obtaining faster restoration. The overall control system is taken apart from the controller end and the nodes just acts as pipes to send the media.

Due to of centralize system, if any controller is shut down the total nodes are effected. Redundancy source has to be created. The network virtualization has been adopted both telecommunication and the computing environment. Thus, we can suggest the researchers to conduct more experiments on open flow network virtualization for efficient trafficking and secured network for future generation.

We use the Mininet tool for evaluate the both latency and bandwidth of the topology. The mininet is setup on the top of the VM and the execution of the steps is to follow from above. The tools place an important role for measuring the values. The controller acts as centralised network for all the hosts. and switches are help to pass the packets from the host to host. After evaluating the results we can conclude that the latency and bandwidth is to taken in consideration for building of new network in between the host. We also observe that the dynamic path is automatically established between the host.The Openflow protocol is very efficient technology and utilization of the bandwidth is more.

7.2 Further Work:

As there is lot of advantages of using the Openflow Protocol in the networking.But there will be some drawbacks like it is complete centralised controller for the hosts and switches. There is no redundant source for the controller¿ if any fail on the controller effect the fail on the whole network. And also, the every packet is to follow up to the controller flow table.Why not to put a small controllers in the hosts for small task.By implementing this system we can minimise th load on the controller

Bibliography

[1] Sugam Agarwal, Murali Kodialam, and TV Lakshman. Traffic engineering in software defined networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 2211–2219. IEEE, 2013.

[2] M Channegowda, R Nejabati, M Rashidi Fard, S Peng, N Amaya, G Zervas, D Simeonidou, R Vilalta, R Casellas, R Martínez, et al. Experimental demonstration of an openflow based software-defined optical network employing packet, fixed and flexible dwdm grid technologies on an international multi-domain testbed. *Optics express*, 21(5):5487–5498, 2013.

[3] Saurav Das, Guru Parulkar, Nick McKeown, Preeti Singh, Daniel Getachew, and Lyndon Ong. Packet and circuit network convergence with openflow. In *Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC)*, pages 1–3. IEEE, 2010.

[4] CJS Decusatis, Aparico Carranza, and Casimer M Decusatis. Communication within clouds: open standards and proprietary protocols for data center networking. *Communications Magazine, IEEE*, 50(9):26–33, 2012.

[5] Hilmi Enes Egilmez. *Adaptive Video Streaming over OpenFlow Networks with Quality of Service*. PhD thesis, Koç University, 2012.

[6] Fernando Farias, Igor Carvalho, Eduardo Cerqueira, Antônio Abelém, Christian E Rothenberg, and Michael Stanton. Legacyflow: Bringing openflow to legacy network environments.

[7] Evangelos Haleplidis, Spyros Denazis, Odysseas Koufopavlou, Jamal Hadi Salim, and Joel Halpern. Software-defined networking: Experimenting with the control to forwarding plane interface. In *Software Defined Networking (EWSDN), 2012 European Workshop on*, pages 91–96. IEEE, 2012.

[8] Nikhil Handigol, Srinivasan Seetharaman, Mario Flajslik, Nick McKeown, and Ramesh Johari. Plug-n-serve: Load-balancing web traffic using openflow. *ACM SIGCOMM Demo*, 2009.

[9] Miyoung Kang, Eun-Young Kang, Dae-Yon Hwang, Beom-Jin Kim, Ki-Hyuk Nam, Myung-Ki Shin, and Jin-Young Choi. Formal modeling and verification of sdn-openflow. In *Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on*, pages 481–482. IEEE, 2013.

[10] Thomas A Limoncelli. Openflow: a radical new idea in networking. *Queue*, 10(6):40, 2012.

[11] Binghan Liu. *Software Defined Networking and Tunneling for Mobile Networks*. PhD thesis, KTH, 2013.

[12] Lei Liu, Dongxu Zhang, Takehiro Tsuritani, Ricard Vilalta, Ramon Casellas, Linfeng Hong, Itsuro Morita, Hongxiang Guo, Jian Wu, R Martinez, et al. Field trial of an openflow-based unified control plane for multi-layer multi-granularity optical switching networks. 2012.

[13] Vijay Mann, Kalapriya Kannan, Anilkumar Vishnoi, and Aakash S Iyer. Ncp: Service replication in data centers through software defined networking. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 561–567. IEEE, 2013.

[14] Dave Mcdysan. Software defined networking opportunities for transport. *Communications Magazine, IEEE*, 51(3):28–31, 2013.

[15] Marc Mendonça, Bruno Nunes Astuto, Xuan Nam Nguyen, Katia Obraczka, Thierry Turletti, et al. A survey of software-defined networking: Past, present, and future of programmable networks. 2013.

[16] Raul Muñoz, Ramon Casellas, and Ricardo Martínez. Pce-what is it, how does it work and what are its limitations? In *Optical Fiber Communication Conference*. Optical Society of America, 2013.

[17] Bram Naudts, Mario Kind, Fritz-Joachim Westphal, Sofie Verbrugge, Didier Colle, and Mario Pickavet. Techno-economic analysis of software defined networking as architecture for the virtualization of a mobile network. In *Software Defined Networking (EWSDN), 2012 European Workshop on*, pages 67–72. IEEE, 2012.

[18] Guillermo Romero de Tejada Muntaner. *Evaluation of OpenFlow Controllers*. PhD thesis, KTH, 2012.

[19] Mateus Augusto Silva Santos, Bruno Trevizan De Oliveira, Cintia Borges Margi, Bruno Nunes Astuto, Thierry Turletti, Katia Obraczka, et al. Software-defined networking based capacity sharing in hybrid networks. 2013.

[20] Sumanth M Sathyanarayana. *Software defined network defense*. PhD thesis, University of Pennsylvania, 2011.

[21] Syed Amir Shahzad. Route aggregation in software-defined networks. 2013.

[22] Sachin Sharma, Dimitri Staessens, Didier Colle, Mario Pickavet, and Piet Demeester. Enabling fast failure recovery in openflow networks. In *Design of Reliable Communication Networks (DRCN), 2011 8th International Workshop on the*, pages 164–171. IEEE, 2011.

[23] Dimitri Staessens, Sachin Sharma, Didier Colle, Mario Pickavet, and Piet Demeester. Software defined networking: Meeting carrier grade requirements. In *Local & Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on*, pages 1–6. IEEE, 2011.