IP ADDRESSING, TRANSITION AND SECURITY IN 5G NETWORKS

Evans Kiptoo Bartocho

This thesis is submitted in partial fulfilment of the academic requirements

for the Degree of

Master of Engineering in Telecommunications

in the Faculty of Engineering and The Built Environment

University of Cape Town

April 2018

As the candidate's supervisor, I have approved this dissertation for submission.

Name: Dr. Joyce Mwangama

Signed:_____

Date: _____

Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

This work is submitted for the Degree of Master of Engineering specialising in Telecommunications at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

Name: Evans Bartocho Kiptoo

Signed by candidate

Date: 9th April 2018

Abstract

The number of devices on the Internet is always increasing and there is need for reliable IP addressing. 5G network will be built on two main technologies; SDN and NFV which will make it elastic and agile compared to its predecessors. Elasticity will ensure that additional devices can always be added to the network. IPv4 addresses are already depleted and cannot support the expansion of the Internet to ensure the realization of future networks. IPv6 addressing has been proposed to support 5G networking because of the sufficient number of addresses that the protocol provides. However, IPv4 addressing will still be used concurrently with IPv6 addressing in networks until they become fully IPv6 based.

The structure of IPv4 header is different from IPv6 header hence the two protocols are incompatible. There is need for seamless intercommunication between devices running IPv4 and IPv6 in future networks. Three technologies namely; Dual Stack, Tunneling and Translation have been proposed to ensure that there is smooth transition from IPv4 to IPv6 protocol. This dissertation demonstrates Tunneling of IPv6 over IPv4. Also, this research work reviews network security threats of past networks that are likely to be experienced in 5G networks. To counter them, reliable IP security strategies used in current networks are proposed for use in next generation networks.

This dissertation evaluates and analyzes IPv4, IPv6 network and Tunneling models in an SDN network environment. The performance of an IPv4 only network is compared to the IPv6 only network. Also, devices addressed with both protocols are connected. The results obtained illustrate that IPv4 and IPv6 devices can effectively communicate in a 5G network environment. In addition, a tunnel is used to run IPv6 protocol over an IPv4 network. The devices on both ends of the tunnel could communicate with each other effectively.

Acknowledgement

I thank my supervisor Dr. Joyce Mwangama for her immense support and constructive criticism towards completion of this research. The experience of working with Dr. Mwangama has always been delightful and enlightening. I feel privileged to have worked with her.

I also thank my parents, brother and sisters for their encouragement and financial support in my education.

Finally, I thank the Almighty God for giving me the strength and sanity of mind to keep on working hard towards my goal.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ACL	Access Control List
AH-ESP	Authentication Headers Encapsulating Security Payloads
API	Application Programming Interface
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
DNS	Domain Name System
EPC	Evolved Packet Core
ETSI	European Telecommunication Institute
GRE	Generic Routing Encapsulation
GNS3	Graphical Network Simulator-3
ICANN	Internet Corporation Assigned Numbers
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPsec	Internet Protocol Security
IoT	Internet of Things
ITU	International Telecommunication Union
MANO	Management and Orchestration
MTD	Moving Target Defense
M2M	Machine to Machine
NAT	Network Address Translation
ODL	OpenDaylight
OF	OpenFlow
PoC	Proof of Concept
QEMU	Quick Emulator Virtual Machine

RADIUS	Remote Dial-in User Service
REST	Representational State Transfer
SDN	Software Defined Networking
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TSL	Transport Layer Security
UDP	User Datagram Protocol
VM	Virtual Machine
VNF	Virtualised Network Function
VTN	Virtual Tenant Network

CHAPTER 1

1 INTRODUCTION

1.1. Introduction

The Fifth-generation of networks (5G) will be driven by Software Defined Networking (SDN) and Network Functions Virtualization (NFV) [1]. SDN ensures easy programmability and debugging of networks on the fly whereas NFV virtualizes network functions so that they run on industry standard servers for scalability and agility [1].

The computing, storage and networking devices in 5G networks will implement the Internet Protocol (IP) for efficient communication. Both the physical and virtual devices in SDN/NFV-based networks will require IP addressing for identification. Mobile devices will connect to the network randomly and therefore will implement dynamic IP addressing. 5G networks will support many applications and services requiring communication with and through numerous devices. Pico-cells will be used to facilitate connectivity of numerous devices in densely populated areas. Most of these devices will be mobile devices and will be issued dynamic IP addresses using Dynamic Host Configuration Protocol (DHCP) [2].

DHCP is an important protocol that facilitates issuance of IP addresses dynamically to network devices. In past and current networks, DHCP servers function to lease IP addresses automatically to devices for a period. Network devices held on to their issued IP addresses until their lease time expired. In networks with shortage of IP addresses, lease times depended on the importance of devices on the network. Important devices were prioritized when it came to IP address allocation. In the 5G use-cases e.g Extreme real-time communications [1], applications will require 99.999 percent reliability and the latency of the network should also be extremely low to meet Quality of Service (QoS) requirements. DHCP can be leveraged to ensure efficient management of IP addresses [2]. The DHCP functionality most probably would be implemented on the SDN controller in a 5G network [3]. The network controller has complete visibility of the network.

Internet Protocol version four (IPv4) addresses have already been depleted and Internet Protocol version six (IPv6) addressing provides for more than 50 billion addresses available for use [4]. Technologies such as Dual stack, Translation and Tunneling will be useful in the transition towards a fully IPv6 based network. To achieve great progress in the transition towards IPv6 only networks, these three technologies should be used simultaneously. This is because each of them has its own setbacks and it would make sense to use them to complement each other.

IP addressing will be relevant in 5G network slices. Network resource slicing in 5G will ensure maximum utilization of resources i.e computing, storage, networking. In each of the slices multi-tenancy will be used to divide resources amongst the different network operators [5]. These operators will have machines or devices running in their networks. These devices will have to be issued with IP addresses. DHCP will allocate IP address ranges to different subnets representing tenants in network slices in future networks.

5G networking is not only about human subscribers but about providing connectivity in general to all devices. Therefore, 5G has several use-cases that are unique compared to its predecessors. Each of these use-cases will be implemented with device or machine connections. An example is the Internet of Things (IoT) which will have sensor networks [1]. For sensors to communicate with actuators anywhere in the network to perform specific tasks, IP address identification will be key. Also, for Machine to Machine (M2M) communication, IP addresses will have to be issued to all devices in the network. Smart cities will be a reality in the future. The various smart city applications e.g. Smart Energy, Traffic monitoring,Tracking & Tracing, Smart cars will have sensors connected to a centralized IT system and then to actuators [1]. All these devices and machines will require IP addresses for communication and identification. The functions of these devices will in one way or another be connected to the economy and administration of a smart city.

Another important use-case of 5G is Extreme real-time communications i.e tactile Internet [1]. Tactile Internet will ensure a balance of delay, efficiency, reliability and scalability in 5G networks. Tactile Internet is mobile in proximity; therefore, network islands are mobile and move with mobile devices. Devices dynamically join and leave the network islands. These mobile network devices will require dynamic IP addressing to maintain constant connectivity to the network islands. Dynamic IP addressing will be necessary in ensuring sufficient security in 5G networks by making network topologies unpredictable through changing of IP addresses periodically. One concept that will secure future networks using IP is the Moving Target Defense (MTD). In MTD, the topology of the network is changed from time to time to mislead intruders [6].

Dynamic IP addressing on its own cannot guarantee reliable 5G networks. Therefore, it must be interfaced well with technologies that drive 5G such as SDN and NFV [1]. These technologies will make the implementation of DHCP easy and reliable. With proper planning and management through SDN and NFV, mobile devices will be able to connect and disconnect to/from 5G networks swiftly without interruptions. This will also ensure in time debugging and

logging of faults in the network as it will be easy to distinguish devices that have static IP addresses from the ones that have dynamic IP addresses.

1.2. Problem Statement

In 5G networks billions of IP addresses will be used for identification and communication of devices [1]. There is need for an efficient IP addressing solution that will be used in these networks. 5G networks will have many devices that must communicate well with each other to meet the requirements by International Telecommunication Union (ITU), Institute of Electrical and Electronic Engineers (IEEE) and other standardization bodies.

5G technologies; SDN and NFV [1], have their proposed architectures that will be implemented in future networks. The use of these technologies makes the network design dynamic. Therefore, IP must be implemented effectively to fit in well with the dynamic nature of the network.

The IP addressing model used in the next generation networks must ensure interoperability between devices supporting IPv4 and IPv6 protocol [7]. Dual stack is a technology that involves running both IPv4 and IPv6 addresses on devices simultaneously. Implementing it as the only IPv6 transition technology is costly as all devices that run only one protocol will have to be replaced by ones that support both IPv4 and IPv6 protocols. Tunneling and Translation should therefore be used as alternatives to Dual stack as we approach IPv6 only based networks. Tunneling involves running IPv4 over an IPv6 network or vice versa whereas Translation is using an IPv6 address as an IPv4 address or vice versa. These three technologies must be complementary to each other to optimize network performance.

In 5G networks there will be billions of wireless devices as mentioned in [1]. These devices will require dynamic IP addressing as they connect and disconnect to/from the network periodically. A good IP addressing policy must be used to avoid confusion and ensure network stability. IP address pools must be managed centrally from the SDN controller in future networks.

Security is very crucial in a network environment as stated in [6]. A good IP addressing policy would ensure security in 5G networks. This is particularly relevant when it comes to implementing multi-tenancy in 5G. Various tenants or network operators will share network slices in 5G networks as explained in [6]. The use of Access Control Lists (ACLs) and IP filtering on firewalls will provide security to tenant networks. Through separation of each tenant efficient IP addressing will ensure confidentiality; one tenant cannot access the information of another. The devices hosted in the different tenant networks in a 5G slice will be allocated IP addresses

from different pools.

1.3. Aim and Objectives

The aim of this research work is to model an IP addressing and transition solution for a 5G network. The proposed solution will include both IPv4 and IPv6 addressing in next generation networks. The internetworking of IPv4 and IPv6 addressed devices is proposed for SDN-NFV based networks. IP Transition technologies; dual stack, tunneling and translation are reviewed for use in future networks. Their role in the path towards IPv6-only future networks is discussed. Out of these three technologies, tunneling is implemented in a proposed network model. Network security is discussed in general and the use of IP to secure future networks is also proposed.

The following are the objectives that will be attained by this research work;

- To model IPv4 and IPv6 networks on an SDN based network environment
- To review the IP transition technologies that are proposed towards the realization of IPv6 networks
- To model IP tunneling by connecting two IPv6 networks via an IPv4 network using a Generic Routing Encapsulation (GRE) Tunnel in an SDN network environment
- To review common threats to network security related to IP and their solutions
- To suggest some of the IP security solutions used in past networks that can be used in 5G networks
- To validate the communication of IPv4 and IPv6 network devices in an SDN network environment
- To validate IP Tunneling between two IPv6 networks by sending test data streams across and using ICMP messages

1.4. Scope

This research work gives a model of IPv4 and IPv6 network in an SDN environment. This model shows the working of IPv4 and IPv6 addressing in an SDN network. The IP transition technologies; Dual Stack, Tunneling and Translation are discussed. Out of these, tunneling is the chosen method for IPv4 to IPv6 transition. A GRE tunnel is created in an SDN network environment to link two IPv6 networks over an IPv4 network. This demonstrates how IPv4 can be run over 1Pv6 or vice versa in future networks.

This work also focuses on the technologies that build 5G networks; SDN and NFV. These technologies must interoperate well with the Internet Protocol (IP) for efficient networks. The other focus of this work is to discuss the network security attacks common in past networks that are likely

to be experienced in the next generation of networks. These attacks involve the alteration of Internet Protocol in the network. The network security strategies that have been used before in past networks that involve the use of IP to secure networks have also been mentioned.

1.5. Dissertation Outline

In this Chapter, an introduction to IP addressing, transition and security in 5G SDN- based networks is made. The applications of 5G networks and the role of the Internet Protocol in ensuring that devices communicate well in network environments are given. IPv4 and IPv6 have been introduced as used in networks. IP transition has been introduced and the importance of transitioning to IPv6 in future networks is given. The role of DHCP in ensuring efficient IP addressing in networks is also mentioned.

Chapter 2 gives a review of IPv4, IPv6, SDN, NFV, IP Transition and security strategies as used in networks. IPv4 and IPv6 have been discussed as used in networks and their datagram structures given. The two technologies that will build 5G networks; SDN and NFV have been reviewed in detail. IP Transition technologies; Dual Stack, Tunneling and Translation have been discussed. Finally, the threats to network security have been reviewed and network security strategies have been discussed in Chapter 2.

Chapter 3 presents a framework that is proposed for IP addressing, transition and security in 5G networks. The use of IPv6 in SDN networks is explained and the planes of the SDN-IPv6 architecture are discussed in brief. The IP transition architecture is proposed for communication between IPv4 and IPv6 networks. The Tunneling architecture that shows how two IPv6 networks are connected over IPv4 using a GRE Tunnel is defined in this Chapter. This Chapter further explains the application of DHCP in SDN using the SDN-DHCP architecture. Finally, ACLs, AAA, IPsec and DHCPv6 shielding are proposed in this Chapter as security solutions that can be used in the next generation of networks.

Chapter 4 presents the proof of concept implementation for the IP addressing, transition and security solution. This chapter introduces the tools and technologies used for the IP addressing and transition solution. The setup of how the tools and technologies are used to implement the PoC is presented. Finally, the proposed security solutions for next generation networks is presented in this Chapter.

Chapter 5 evaluates and analyses the implemented IPv4, IPv6 only network and the two IPv6 networks connected via a GRE tunnel. The connectivity of the IPv4 and IPv6 network is checked using ICMP requests and Input-Output traffic analysis is done using Wireshark. The

Flows in the network are also evaluated using Wireshark. TCP and UDP throughput in the network is also checked using Iperf and the results analyzed. Finally, the end to end connectivity of the created tunnel is checked using ICMP requests and analyzed using Wireshark. The TCP and UDP throughput across the GRE tunnel is also evaluated and analyzed.

Chapter 6 is the conclusion of the entire dissertation. The two protocols for IP have been discussed as used in networks. The transition from IPv4 to IPv6 is mentioned in brief. The two IP protocols have been compared against each other. The two technologies (SDN and NFV) for building 5G networks have been explained in brief. IP security and DHCP have been discussed and proposed for use in next generation networks. This dissertation explains the importance of IP addressing and transition in 5G networks. It gives an IPv4, IPv6 addressing and transition model and also proposes IP security solutions for 5G networks.

1.6. Chapter Summary

This Chapter introduces 5G networks and explains the importance of IP addressing to communication. The technologies that will build 5G networks have been mentioned in brief. The reason for undertaking this research work is given and the scope is stated. The technologies that are proposed for the transition from IPv4 to IPv6 addressing in future networks are also introduced in this Chapter. Network security is also mentioned with a focus on how IP can be used to secure networks.

CHAPTER 2

2 LITERATURE REVIEW

2.1 Introduction

This chapter discusses the related work associated with this research and gives an in-depth discussion on the relevance of the Internet Protocol (IP) in past, present and future networks. Other works related to IPv4 and IPv6 addressing are discussed. The use of IPv4 in traditional and current networks is reviewed. Then IPv6 is introduced and a discussion follows of its use in current networks and its proposed use in future networks. The relevance of IP to the Internet of Things (IoT), Smart cities and Machine to Machine communication is also discussed in this chapter. IP Transition technologies which are crucial in the migration from IPv4 to IPv6 are explained. These are Dual Stack, Tunneling and Translation. Related work to these IP transition technologies is mentioned in this chapter with a focus on tunneling. The merits and drawbacks of each of these technologies are stated and a review is given of how they have been implemented before in networks.

In this chapter, a review of Software Defined Networking (SDN) and Network Functions Virtualization (NFV) is given. These two technologies are key for the realization of 5G networks. SDN and NFV must complement each other for efficient networking in 5G [8]. The OpenFlow protocol; an SDN standard that ensures that the SDN controller effectively communicates with routers and switches (virtual or physical) in an SDN-based network is reviewed in this chapter.

In any network, security is very crucial in protecting sensitive data from being accessed by intruders. IP is very important in ensuring network security. Several technologies and strategies have been used before to ensure security in networks [6]. A review of these techniques and related work to this research is given in this chapter.

2.2 IP Addressing

The Internet Protocol (IP) address is a numerical tag assigned to devices on the Internet to identify them for purposes of communication [4]. IP addresses on the Internet are assigned to both hosts and network devices to identify and associate them with a given location. Host IP addresses are used to identify individual devices on the Internet whereas network IP addresses are used to identify networks which are made up of hosts. The body responsible for assigning

IP address space globally is the Internet Corporation for Assigned Names and Numbers (ICANN). There are two types of IP addresses namely; IPv4 and IPv6.

2.2.1 IPv4

IPv4 addresses are 32-bit IP addresses [4] that were used for identification in past networks and are still being used in present networks. IPv4 only nodes implement IPv4 addressing. The earliest networks were homogeneous in nature and implemented only IPv4 addressing. With time demand for new services necessitated the innovation of new technologies that supported multimode operations which led to the need for increased capacity in networks.

The number of devices in the internet is always increasing. 3G networks on their own can use billions of addresses and 4G (which is the current network technology) uses more than that as stated in [9]. The Internet Assigned Numbers Authority (IANA) exhausted the global address space for IPv4 in February 2011 while Regional Internet Registries (RIRs) were projected to exhaust their IPv4 address space in 2014 [9]. Currently, IPv4 addresses are reused using Network Address Translation (NAT) technology because of their depletion since 2011. Therefore, it will be impossible to satisfy demand for address space with IPv4 addressing.

There are strategies that were used to solve the shortage of IPv4 address space in networks. [9] explains dynamic IPv4 addressing, where devices on the Internet are temporarily issued with IP addresses. A Dynamic Host Configuration Protocol (DHCPv4) server is responsible for issuance of IPv4 addresses to hosts. It leases IP addresses to devices on the network from a selected pool of addresses [9]. The lease times for addresses depends on the priority of usage of the addresses by the devices. Devices with higher priority are given longer lease times whereas low priority devices are given shorter times. The other technology that can solve the IPv4 address space shortage is Network Address Translation (NATv4). IP address space is categorized into two; public and private. The IP ranges 10.0.0.0- 10.255.255.255, 172.16.0.0-172.31.255.255, and 192.168.0.0-192.168.255.255 are reserved by IANA as classes A, B and C of private IPv4 addresses. NAT translates public IPv4 addresses to private. Public networks (the ones with public IP addresses) by design can access the Internet whereas the private networks cannot as stated in [4].

In modern SDN networks, there are devices that will still use IPv4 addresses. To accommodate these devices, future networks ought to be designed well. The main technologies that will build 5G networks are SDN and NFV as explained in [1] and [10]. These technologies must be compatible with IPv4 addressed devices. The format of IPv4 header is shown below in the Figure 1.

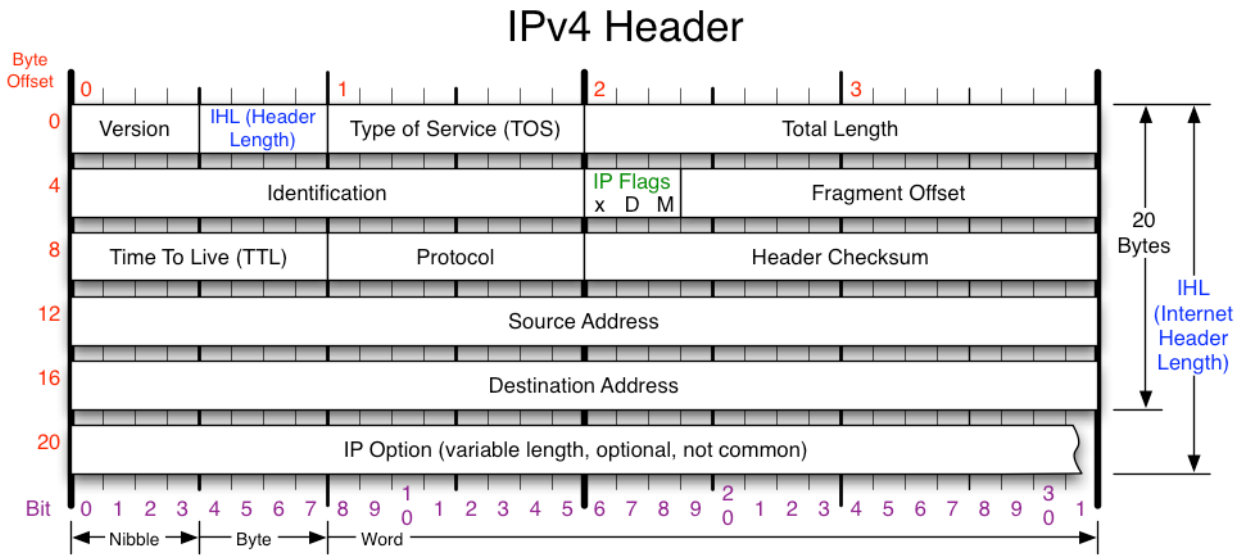

Figure 1 IPv4 Header [11]

2.2.2 IPv6

IPv6 is the latest version of the Internet Protocol. IPv6 is 128-bit in length and was designed by the Internet Engineering Task Force (IETF) to provide more address space as compared to IPv4 [4]. In the long run, it is anticipated that it will replace its predecessor completely. IPv6 provides for 3.4×10^{38} available addresses approximately as stated in [7]. IPv6 address header consists of two parts namely; 64-bit interface identifier and 64-bit prefix. The IPv6 address header is unique because it has a different architecture to that of IPv4. The delegation of address blocks has been restructured in IPv6. Also, the checksum has been removed in the IPv6 datagram to reduce the processing time [7]. It is projected that IPv6 will be the complete solution for the next generation of networks because of its efficiency and robustness. IPv6 addressing eliminates the need for NAT therefore ensures end-to-end security and applications according to [9].

In traditional networks, only IPv4 was being used in identification of network devices. In modern networks both IPv4 and IPv6 are used for addressing. In next generation networks IPv6 will be used to meet IP addressing needs because there will be billions of devices hosted [1]. To meet Quality of Service (QoS) requirements in 5G networks, devices running IPv6 must communicate effectively with the ones running IPv4 [8]. There are three technologies that have been suggested and used before for transition from IPv4 addressing to IPv6 are namely; dual stack, tunneling and translation [7].

IPv6 is incompatible with IPv4 because of the difference of its datagram structure from IPv4. K. Govindarajan et al. [12] presents a solution for the interoperability issue between IPv4 and IPv6 through compatible transaction using Dual Stack and Tunneling for various

OpenFlow protocol versions. The Dual stack mechanism is port-based, MAC-based and IP-based in OpenFlow enabled networks. The tunneling mechanism used is Virtual Extensible LAN (VXLAN) and Generic Routing Encapsulation (GRE). These IP transition technologies (Dual Stack and Tunneling) have been discussed in detail later in this dissertation.

The nature of design of IPv6 supports plug and play and autoconfiguration is in-built. IPv6 allows for fragmentation of datagrams at the source and reassembly at the destination [13]. However, there are parts of the datagram that must be left unfragmented. This is the header and other parts that are processed on the way to the destination. Therefore, nodes that transmit datagrams that need to be fragmented should not create overlapping fragments according to [14] otherwise the fragments that are found to be overlapping will be discarded. The structure of the IPv6 header is as shown in the Figure 2 below.

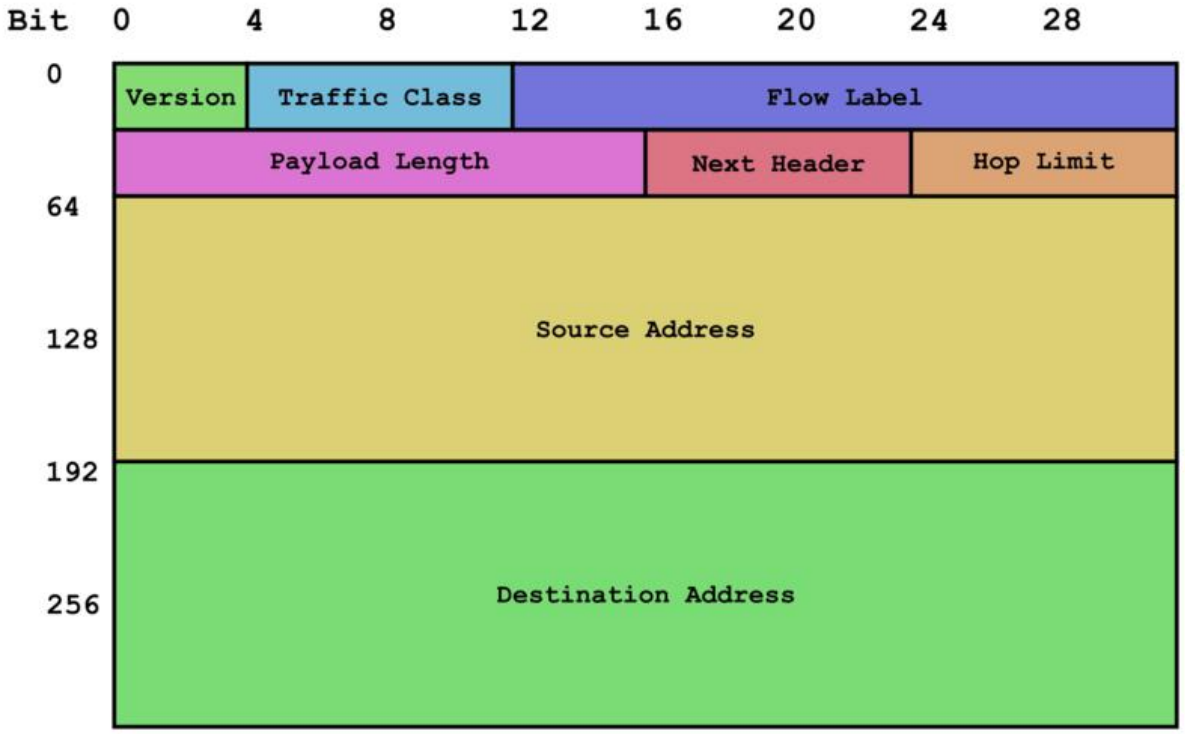

Figure 2 IPv6 header [15]

The various types of addresses associated with IPv6 are; Unicast, Anycast and Multicast. Unicast addressing is where a packet is delivered to only one specific interface (one to one). Anycast addressing is where a packet is delivered to the nearest interface (one to nearest). Multicast addressing is where a packet is delivered to many interfaces (one to many) according to [16]. There are three scopes associated with Anycast and Unicast addresses

namely; link-local, site-local and global. Multicast addresses have their scope located in their structure. The link-local addresses are locally significant, site-local addresses are only relevant to a given organization and global addresses are the ones routed on the Internet.

2.2.2.1 Internet of Things

The Internet of Things is an ecosystem of devices and people connected on the Internet. The devices have sensors which are connected to the IoT platform to share information with applications which are built to perform specific functions [31]. It is projected that there will be more than 50 billion devices in the Internet by the year 2020 [32]. IPv6 addresses will be used to identify these devices in the network because IPv4 addresses have already been depleted.

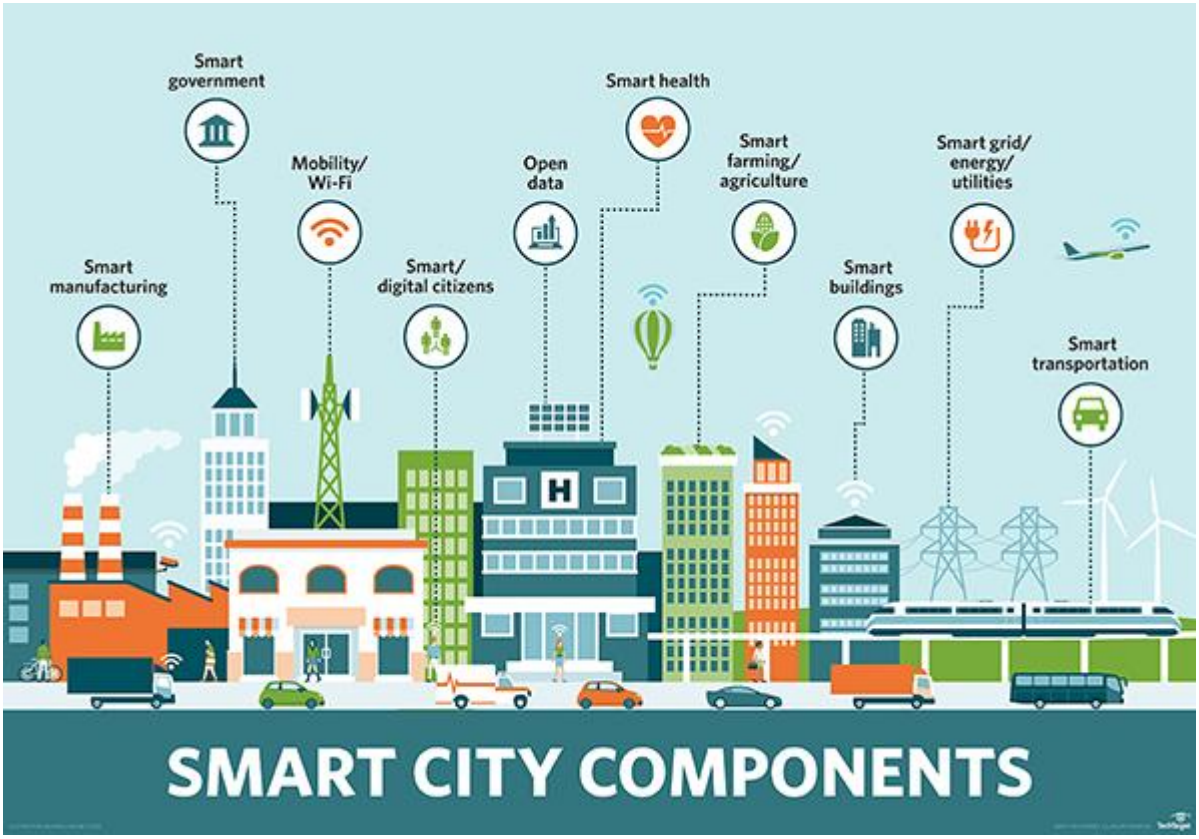

Figure 3 Smart City Applications [33]

Smart city applications namely; smart energy, smart security, smart transport, smart health and others will be a reality in 5G networks. IP addresses will be used for identification and communication of devices to support these applications [34]. Machine to machine (M2M) communication will support the realization of smart cities in the next generation networks [35]. The types of M2M communications that exist are; one to one, one to many, many to many, One to Thing and Things to Things. The Figure 6 above shows the various smart city applications.

2.3 IP Transition Technologies

IPv4 addresses will still coexist with IPv6 in the coming years. There are three technologies that have been proposed by the Internet Engineering Taskforce (IETF) as ways of implementing IPv4 and IPv6 concurrently in next generation networks. These are namely; dual stack, tunneling and translation [17]. Out of these approaches, dual stack is mostly preferred due its simplicity even though its high cost of deployment is its drawback.

2.3.1 Dual Stack

Dual stack is a technology where devices on the network have inbuilt support for both IPv4 and IPv6. The devices running dual stack normally give preference to IPv6. Devices running IPv4 only should communicate with devices running IPv6 only without hitches via devices running dual stack as explained in [12]. In dual stack, the Domain Name System (DNS) resolver must be able to identify both IPv4 and IPv6 if they are deployed individually. Another alternative to ensure constant communication in a network is to deploy only the devices running dual stack. Even though address space capacity is sufficient when IPv6 is used only, IPv4 and IPv6 will coexist for many years before networks will be able to run only IPv6 as stated in [17]. This is because some network functions only work efficiently with IPv4 addressing.

K. Govindarajan et al. [12] presents a Dual Stack solution where each device resides in an OpenFlow enabled cloud network. The devices are configured with both IPv4 and IPv6 addresses. This eliminates complexities that are possible with tunneling and translation mechanisms.

IPv4 to IPv6 communication is still work in progress even as we go towards next generation networks. The following are the main challenges associated with the implementation of dual stack;

- IPv4 Network Management tools may not be compatible with IPv6
- Dual stack requires hardware and software upgrades which might be costly. IPv4 devices that do not support IPv6 will have to be replaced with devices that support both protocols throughout the IPv4 only network [18].
- The security solutions for the two technologies are different, therefore making IPv4 security solutions work in IPv6 might be a daunting task.
- Security controls and firewall restrictions that stop unwanted IPv4 traffic might not stop all the unwanted IPv6 traffic.
- The idea of having every interface having both IPv4 and IPv6 makes dual stack complex in environments that implement either IPv4 or IPv6 purely according to [12].

2.3.2 Tunneling

Tunneling is a technology that allows private network data to be conveyed over a public network with the aim of ensuring that the transmitted data is not accessed by the nodes in the public network [19]. The data to be transmitted is encapsulated to make it look as if it belongs to the public network. The public network over which private network data is transmitted is sometimes the Internet. Regarding IP, tunneling can be used to run IPv4 over an IPv6 network or vice versa. Depending on how it will be implemented, tunneling can be a very useful technology to ensure reliable IP addressing in the transition towards IPv6 only next generation networks.

K. Govindarajan et al. [12] presents tunneling solutions using Virtual Extensible LAN (VXLAN) and GRE. The GRE tunnel created connects two IPv6 networks over an IPv4 network. This is similar to the IPv6 tunneling over IPv4 covered in this research work. VXLAN allows for the creation of a logical network for VMs that are located on different networks (creates layer 2 networks on top of layer 3) and allows for scaling of virtual networks. A. Burande et al. [20] presents Secure Shell (SSH) tunneling a method that creates a virtual tunnel over a wireless network. This tunnel connects the client and server on both ends of the tunnel. Port forwarding is used to establish the tunnel and traffic going across this tunnel gets encrypted by default.

Tunneling must be interfaced well with SDN and NFV [6] to ensure efficient IP addressing in next generation networks. If implemented well, this is the technology that will ensure secure networks. When 5G network is deployed, there will be more devices running IPv6 than IPv4 to meet IP addressing needs [1]. Therefore, it will be reasonable to run IPv4 over IPv6 networks for simplicity. Tunnels to run IPv4 will be created in IPv6 networks to route IPv4 traffic over IPv6. However, this approach to IP transition has the following demerits;

- The users of IPv4 network cannot use the services of IPv6 network; IPv4 traffic will only be forwarded across IPv6 network.
- There is no interoperability between devices; IPv6 hosts cannot communicate with IPv4 hosts without dual stack as indicated in [18].

2.3.3 Translation

Network Address Translation (NAT) is commonly used in traditional and modern networks to translate between private IPv4 addresses and public IPv4 addresses as in [12]. NAT64 provides for translation from IPv6 address to IPv4 address and vice versa. NAT64 translation occurs between IPv6 only and IPv4 only hosts and networks. It is another alternative to dual stack and tunneling regarding transition from IPv4 to IPv6 only addressing in future networks as stated in [18]. Translation is preferred to dual stack and tunneling because of the following reasons;

- It is seamless and gradual when migrating from IPv4 to IPv6

- In 5G networks, service providers can provide services transparently to IPv4 Internet users [17].

J. Lin et al. [21] presents an IPv6 translation migration approach over SDN. In this approach, IPv4 and IPv6 only networks are interconnected with OpenFlow switches and are all connected to a centralized controller. Two OpenFlow switches support IPv4/IPv6 hybrid network while a specific OpenFlow switch manages NAT64 pool and only offers IPv6 connectivity. The OpenFlow switch on the IPv6 side communicates with the controller to manage load balancing for the NAT64 pool. The OpenFlow switch on the IPv4 side is just a general network switch. OpenStack is exploited to deploy NAT64 nodes for address translation.

Translation from IPv4 to IPv6 and vice versa is different from IPv4 NAT as it was originally used in traditional networks. This is because the two protocols in discussion have differently structured headers thus incompatible which make the translation from one to the other complex.

2.4 Software Defined Networks (SDN)

2.4.1 SDN Architecture

SDN is a technology that is key to the realization of 5G networks. In SDN, the control and data/forwarding plane of network are physically separated from each other. The control plane is the part of the network that makes decisions of where data is to be routed in the network whereas the data plane is responsible for simply forwarding the data. SDN introduces network programmability which makes it possible to control and manage network functions through software. SDN is complementary to NFV even though both technologies use different methods to achieve their results [8].

The Open Networking Foundation [22] is a non-profit organization dedicated to development, standardization and commercialization of SDN [8]. The SDN Architecture consists of three layers namely; Infrastructure, Control and Application as shown in Figure 4 below. The Application layer consists of programs that communicate to the control layer using Northbound APIs [6]. These applications also use the abstracted view of the network for their internal decision making [6]. The Control layer in the SDN architecture provides for direct programmability of networking functions. The SDN controller in this layer is moved outside the internal network to ensure that it has complete network view. The controller has a similar purpose to that of a traditional operating system [8]. The Infrastructure layer consists of network devices. This is similar to the infrastructure layer of traditional network environments only that now these devices are simple forwarding elements without decision making ability [8]. Southbound APIs are used

to communicate between the Control layer (control plane) and the Infrastructure layer (data plane).

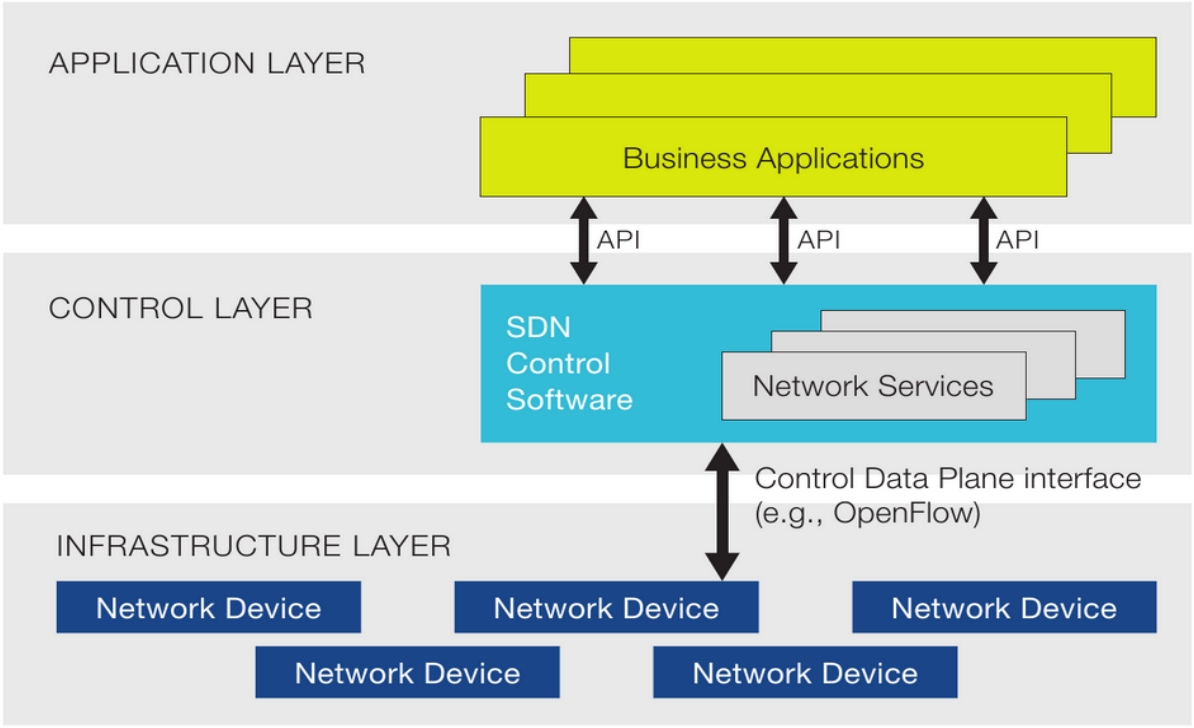

Figure 4 SDN Architecture [22]

There are several reasons why SDN is one of the preferred technologies for next generation networks. The SDN architecture is;

- Easily programmable because of the separation of the control plane from the data plane of the network.
- Agile; decoupling the control plane from the data plane ensures the flow of traffic in the network is controlled dynamically by administrators.
- Centrally managed; the intelligence of the network is centralized in SDN controllers which have a global view of the network which appears to applications as a logical switch [22].
- Configured on the fly; SDN lets network engineers and managers to make changes to networks very quickly using their own programs since the programs are not proprietary.
- Open source nature; instructions to networks are provided by the SDN controller and not several vendor-specific devices and protocols [1].

2.4.2 OpenFlow Protocol

This is a protocol that facilitates communication in an SDN network between switches and a server through issuance of instructions to ensure that packets are sent by the switches. In current networks, switches run software that is proprietary which instruct them where and when to send packets. In the OpenFlow protocol, the decision-making process to send packets is centralized because the computing intelligence is separated from the switches [23]. Therefore, the network can be programmed separately from the switches.

OpenFlow connects the switches associated with it to the OpenFlow controller [6]. The OpenFlow switches have flow tables which have entries of flows in the network. An OpenFlow switch separates the decision plane from the forwarding plane [10]. The forwarding plane resides within the switch; a separate controller makes intelligent decisions on behalf of the switch. The controller functions to add flow table entries to switches. The switch then uses these entries to forward packets. The controller affects the nature of flows in the network by inserting, modifying and removing entries of flow tables [6]. OpenFlow ensures easy, innovative and fast deployment of switching and routing protocols to SDN networks. It is used for applications in next generation IP networks and secure networks.

2.5 Network Functions Virtualization (NFV)

NFV is a standard defined by European Telecommunications Standards Institute (ETSI) that offers a new approach to the design of networks whereby network functions are deployed and managed to run separately as software on dedicated industry standard servers rather than on proprietary hardware devices [23]. NFV is complementary to SDN even though it can be implemented without requiring SDN and vice-versa [8]. As much as NFV depends on traditional server virtualization techniques, it is unique. A virtual session controller can be used in the deployment of NFV to secure the network with simplicity and at a low cost.

The ETSI NFV proposed architecture has three main domains namely; Management and Orchestration (MANO), Virtual Network Functions (VNFs) and Virtual Network Functions Infrastructure (NFVI) as shown in the Figure 4 below.

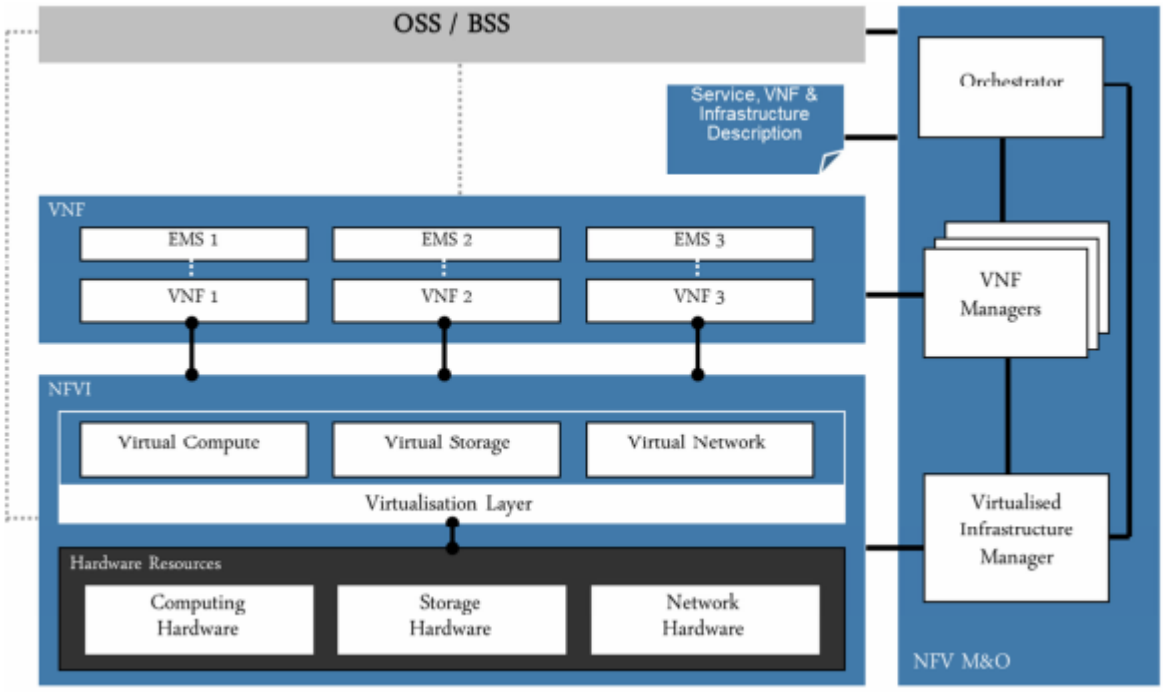

Figure 5 NFV Architecture [24]

The NFVI consists of both physical and virtualized infrastructure. The computing, networking and storage functions are abstracted from hardware appliances and are virtualized on top in the virtualization layer [6]. The Virtual Network Functions (VNFs) are images of software that run on top of the NFVI [6]. The Management and Orchestration (MANO) consists of the VNF manager, Virtual Infrastructure Manager (VIM) and the NFV orchestrator [10]. The VNF manager is responsible for the lifecycle management of Virtual Network Functions (VNFs). The VIM oversees the functionality of the networking, computing and storage resources of the network. The NFV orchestrator ensures the internetworking of in the SDN network through standardization of virtual network functions [10].

In a 5G network, virtualization of network functions will have the following benefits;

- There is reduction of expenditure and improved efficiency because of sharing of infrastructure by the various network operators. The time to market is reduced because applications are deployed faster. Cloud computing ensures flexibility in the network applications [25].
- Scalability; the network can be scaled up easily as a new hardware equipment can be connected to the network without going through lengthy procurement processes.
- Elasticity; the network can be scaled up or down depending on the requirements. Scaling down will free up resources in the network therefore reducing operational costs [25].

2.6 Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS)

DHCP is a protocol that functions to automatically issue IP addresses to devices in the network. The DHCP server assigns an IP address and default gateway to the client automatically [26]. A network device requests an IP address from the DHCP server by default therefore eliminating the need for static addressing which is tedious. Whenever there is shortage of IP addresses, the DHCP server includes lease times when issuing addresses. High priority devices are given long time leases whereas low priority devices are given short time leases [27][28].

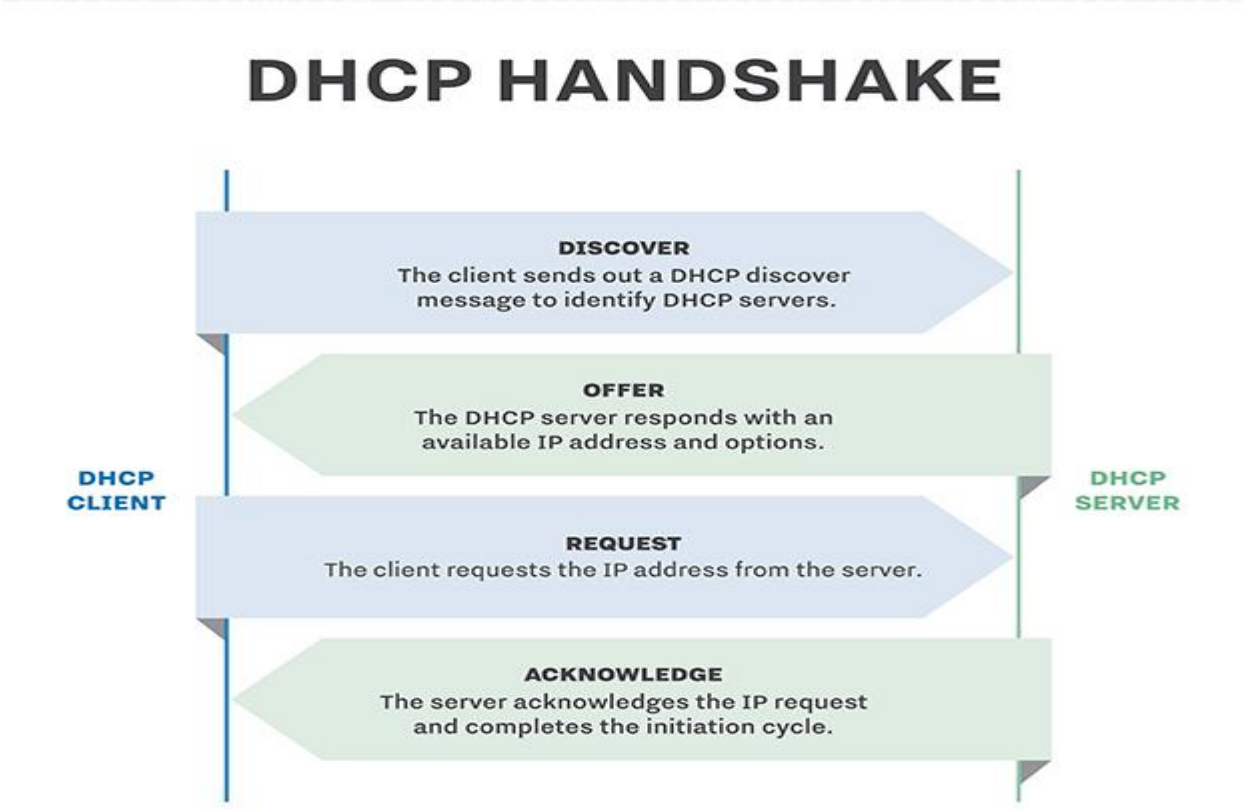

Figure 6 DHCP Handshake [48]

In traditional networks, DNS and DHCP functions were hosted on physical servers which hosted other network services [29]. This had the following demerits;

- Because there were many other network functions being hosted on the same server, it resulted in overloading thus making scalability impossible.
- If the physical server hosting different functions failed, it resulted in single point of failure in the network.
- If one software on the server is breached it led to vulnerabilities on the other services being hosted on the same appliance.
- Running different services on the same server led to inconsistent maintenance and

management practices.

- Since several different services are being hosted on the same server, making changes to the network architecture was almost impossible hence inflexibility in deployment.

The IP addresses and configurations in a 5G network virtualized environment will be always be changing [1]. To address this, DNS and DHCP functions have to be virtualized [30]. The following are the benefits of virtualizing DNS and DHCP in networks;

- Virtualizing these two functions will ensure maximum CPU utilization in future networks
- There is cost savings in the network because the number of hardware appliances to be deployed is decreased. Maintenance costs associated with equipment's will also be readily reduced.
- Elasticity; scalability and rapid deployment resulting from virtualization of these functions.

2.7 Common Threats to Network Security

There are threats to network security that have been experienced in past and present networks. The same security threats are likely to be experienced in 5G networks. The ones that relate directly to IP routing are Denial of Service and IP Spoofing [36]. These security threats and their possible solutions are reviewed below.

2.7.1 Denial of Service (DoS)

This is where an attacker denies an organization access to its resources. DoS attacks are very common and account for almost one third of network attacks. Attackers use the strategy of overloading the resource with illegitimate requests [37]. Techniques used to prevent DoS attacks are use of firewalls, Intrusion Prevention Systems, Blackholing and Sinkholing.

Firewalls can be used to filter untrusted IP traffic from the genuine one thus limiting illegitimate requests on resources [36]. Intrusion Prevention Systems are used to determine suspicious activities in enterprise networks. Blackholing is where traffic going towards a given destination IP address in the network is diverted to a 'black hole' where it is discarded for security purposes. Sinkholing involves routing traffic maliciously to a given destination IP address. This distinguishes between valid and invalid IP traffic being routed [36][37].

2.7.2 IP Spoofing

This is a type of security threat whereby the attacker hijacks the IP address of a legitimate host and changes the packet header so that another host appears to be the source of the packets [38]. The attacker can also use unique applications to reconstruct IP packets so that they may look as if their source is inside the private network. To prevent IP spoofing;

- The inbound and outbound IP traffic should be filtered.
- Encryption keys should be used for authentication between hosts in the network. IPsec should limit the risk of the network being breached via spoofing.
- Use Access Control Lists (ACLs) to deny private IP addresses on downstream interfaces.

2.8 Network Security Strategies

Network security is very important in any network. There are methods that have been used before to ensure security in traditional and modern networks. Strategies such as IP filtering and pooling have been used in past networks to prevent unauthorized access. This dissertation reviews Network Access control, use of Firewalls and Moving Target Defense (MTD) [36]. These security methods that relate to IP can be used to secure 5G networks from potential attackers.

2.8.1 Network Access Control

This is where unwanted users are denied access to the network to keep away intruders. Each user and device must be identified by the network before it connects. This can be done by creating an IP address database of devices allowed on the network in the networking device e.g router, server. This is done by using ACLs to allow or deny access to devices [36]. Devices whose IP addresses are not found in the common database can be given limited access. This is an important method that can be used to secure SDN networks.

2.8.2 Firewall

Firewalls are used to protect networks from unauthorized access. They have a predetermined set of security controls to affect the incoming and outgoing data flows. Firewalls protect the internal private network from the untrusted external network like the Internet [36]. They filter IP traffic being conveyed across two networks. There are other Firewalls that control IP traffic in and out of hosts. Firewalls are implemented either in software or in hardware. In SDN-NFV based networks, the Firewall function can be virtualized in the infrastructure layer.

2.8.3 Moving Target Defense (MTD)

Networks that have a static nature of configuration can easily be breached by attackers. Static configuration means that network software, names, IP addresses and security protocols remain the same for a long period. To address this threat, MTD strategy was introduced. MTD brings dynamism into the network to ensure unpredictability [6]. IP addresses can be changed from time to time in the network for devices that are statically addressed. Many devices in the next generation networks will be mobile devices. This means that they would automatically be

assigned dynamic IP addresses using DHCP [2]. This would secure the networks. The only thing that needs to change from time to time is the pool of IP addresses in the DHCP server.

There are three ways to make the network unpredictable in MTD. These are;

- Network Change; in this type of MTD the network topology is changed by IP hopping, using random port numbers, unclear port traffic and counterfeit information about the host and Operating System.
- Host Change; here the hosts and Operating Systems can be renamed and reconfigured. Their resources can also be changed.
- Application Change; the application environment can be changed. The type, settings, APIs and version of the application can be changed.
- The main aim of MTD is to increase the workload of an attacker to give the network defender an equal chance of securing the network.

2.9 Chapter Summary

This Chapter reviews IPv4, IPv6 and the proposed IP transition technologies in networks. The application of IP in the Internet of Things is also discussed. SDN and NFV; the main technologies for building 5G networks have also been reviewed. The common threats to network security is reviewed and their solutions involving IP filtering are given. Finally, network security approaches related to IP that can be used in 5G networks are reviewed.

CHAPTER 3

3 IP ADDRESSING, TRANSITION AND SECURITY IN 5G NETWORKS

3.1 Introduction

There is need for a suitable IP addressing approach to be used in the next generation of networks. 5G networks will incorporate two main technologies namely; Software Defined Networking (SDN) and Network Functions Virtualization (NFV) [1]. SDN will make future networks majorly software based and ensure programmability. It decouples the control function of the network from the forwarding plane [23]. NFV will facilitate virtualization of network functions for agility, flexibility, efficiency and scalability of the network [30]. IPv4 and IPv6 addresses will be used for identification and communication of devices in 5G networks. IP transition technologies mentioned in Chapter 2 will ensure that IPv4 and IPv6 devices intercommunicate effectively with each other until the network is fully IPv6 based. This is because IPv6 is incompatible with IPv4 [7]. There must be backward compatibility between 5G and its predecessors; 3G and 4G which mostly implement IPv4 addressing. Dynamic Host Configuration Protocol(DHCP) will ensure a reliable dynamic IPv6 addressing particularly to wireless mobile devices which will be many in 5G networks [2]. The billions of devices hosted by 5G networks will also make DHCP a useful protocol to be implemented.

This chapter presents an elaborate IP addressing and transition solution that can be implemented in SDN-NFV based 5G network. DHCP plays a crucial role in ensuring fast and efficient issuance of IP addresses from a specially selected pool [28]. The DHCP function can be virtualized in the infrastructure layer of the NFV Architecture. It can also be implemented in the SDN controller which has complete awareness of the network layout. IP transition through tunneling ensures that IPv6 can be run over an IPv4 network in an SDN network environment. A good IP addressing approach will ensure proper security in 5G networks. Network security is threatened when the identity of a network device is hijacked by an intruder. The identity of a device is represented by its IP whether IPv4 or IPv6. Next generation networks will be more IPv6 based because of capacity [17]. IPv6 addresses also offer more security features as compared to IPv4. This chapter also gives the architecture of an IPv6 based network in SDN.

The rest of the chapter is organized as follows. Section 3.2 represents a framework of IPv6 addressing in an SDN network environment. Section 3.3 shows how the IP transition technologies can be deployed in an SDN-NFV based network. Section 3.4 proposes how DHCP can be implemented in an SDN based network. Section 3.5 shows how network IP security can be achieved in next generation networks using IPv6 addressing. Finally, the chapter summary is presented in section 3.6.

3.2 SDN with IPv6 Addressing

Future networks should not only be ready for IPv6 but also IPv6 based now that IPv4 addresses have already been depleted. Therefore, IPv6 needs to be fully implemented in SDN. IPv6 does not only solve the address depletion problem. It also supports multicast instead of broadcast thereby saving the network bandwidth [16]. Since IPv6 header is different from IPv4 header, packet processing is more efficient because IP-level checksum is eliminated. This dissertation proposes an IPv6 addressing model for SDN-NFV based networks. IPv6 addressing solves some of the security problems associated with IPv4 but not all of them. This is because IP Security (IPsec) in IPv6 makes it more secure than its predecessor by allowing IPv6 packet authentication via extension headers. However, IPv6 addressing is flexible making it vulnerable to other problems. Other security solutions are still being developed for this protocol. These include security solutions that parallel IPv4 security mechanisms and firewall rules that will block undesired tunnels.

There are three technology inflexion points in the IT Infrastructure transformation namely; the Cloud, SDN and IPv6. These provide for Agility, Flexibility and Scalability in networks respectively. SDN should be tested with IPv6 addressing without impacting IPv4 services in the network. Flow Label in the IPv6 structure is used to provide flows for communication within the OpenFlow Protocol.

The Figure 7 below shows the SDN-IPv6 Architecture. IPv6 schemes are implemented as SDN applications in the Service Plane. These SDN applications communicate with the Controllers via the Northbound APIs. The network equipment in this architecture is compatible with SDN. The SDN controllers have complete visibility of the network. If one controller fails, a standby controller takes over the control of network functions.

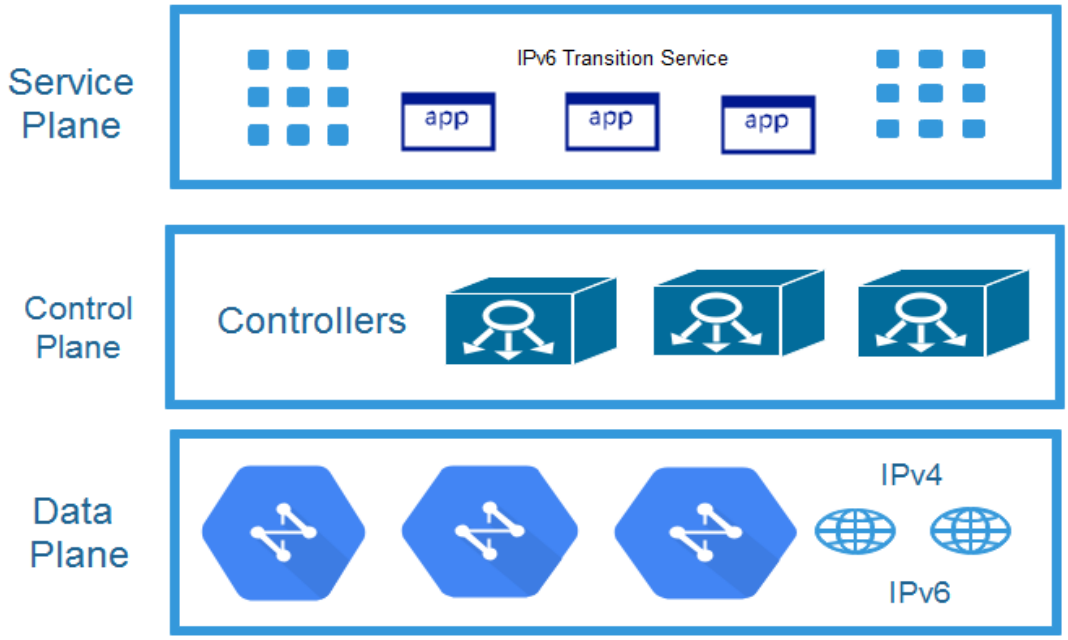

Figure 7 SDN-IPv6 Architecture

The above architecture consists of the Service, Control and Data Plane. These three planes are discussed below.

3.2.1 Data Plane

The Data Plane consists of nodes that are responsible for forwarding data in the network. Data plane traffic travels through network devices rather than to or from them [8]. It is made up of switches and other network devices that implement IP. The network infrastructure in this layer is both physical and virtual. It consists of computing, networking and storage devices which are IPv6 enabled.

3.2.2 Control Plane

The Control Plane is the part of the architecture that makes decision on where traffic is to be sent. It consists of SDN controllers that have visibility of the network and are responsible for making network flow decisions to ensure intelligent networking. This plane is the 'brains' of the network and is responsible for the management of network services. The IPv6 Transition service is managed by the SDN Controller in this plane.

3.2.3 Service Plane

The Service Plane provides network users with services and maintains the state of those services. It coordinates the proper functioning of processes and maintains the database of services running in the network. This plane relies on the capabilities of the Control Plane to effect change on the Data Plane. The plug and play ability of IPv6 addressing enables applications in the Service plane to easily communicate with the Control plane via NorthBound APIs (NBIs).

3.3 IP Transition Technologies

To attain Quality of Service (QoS) requirements in future networks, IPv4 only networks must communicate well with IPv6 only networks at least before networks become fully IPv6 based. There are three technologies discussed in Chapter 2 that will be used for IP transition namely; Dual Stack, Tunneling and Translation. Dual Stack being the simplest of the three is the ideal solution for migration to IPv6. However, it has its fair share of challenges as discussed in the previous chapter. The merits and demerits of the other two technologies have also been discussed in chapter 2. All these technologies can be deployed simultaneously in an IPv6 enabled network for optimal functioning.

3.3.1 Dual Stack

In the proposed Figure 8 below, we have OpenFlow switches connecting to IPv4 and IPv6 only networks. There is a centralized SDN controller connected to these switches. In this Architecture, the OpenFlow switches support Dual Stack for communication with both networks. The SDN controller has visibility of the network and makes decisions on where IP traffic is to be sent. It is also responsible for load balancing amongst the switches. The Dual Stack nodes have a DNS resolver library capable of processing IPv4 A record and IPv6 AAAA records. The 'A' record in IPv4 stands for address and for this protocol they are mapped for 32-bit addresses. The 'AAAA' represents an IPv6 address that maps a hostname to 128-bit IPv6 address.

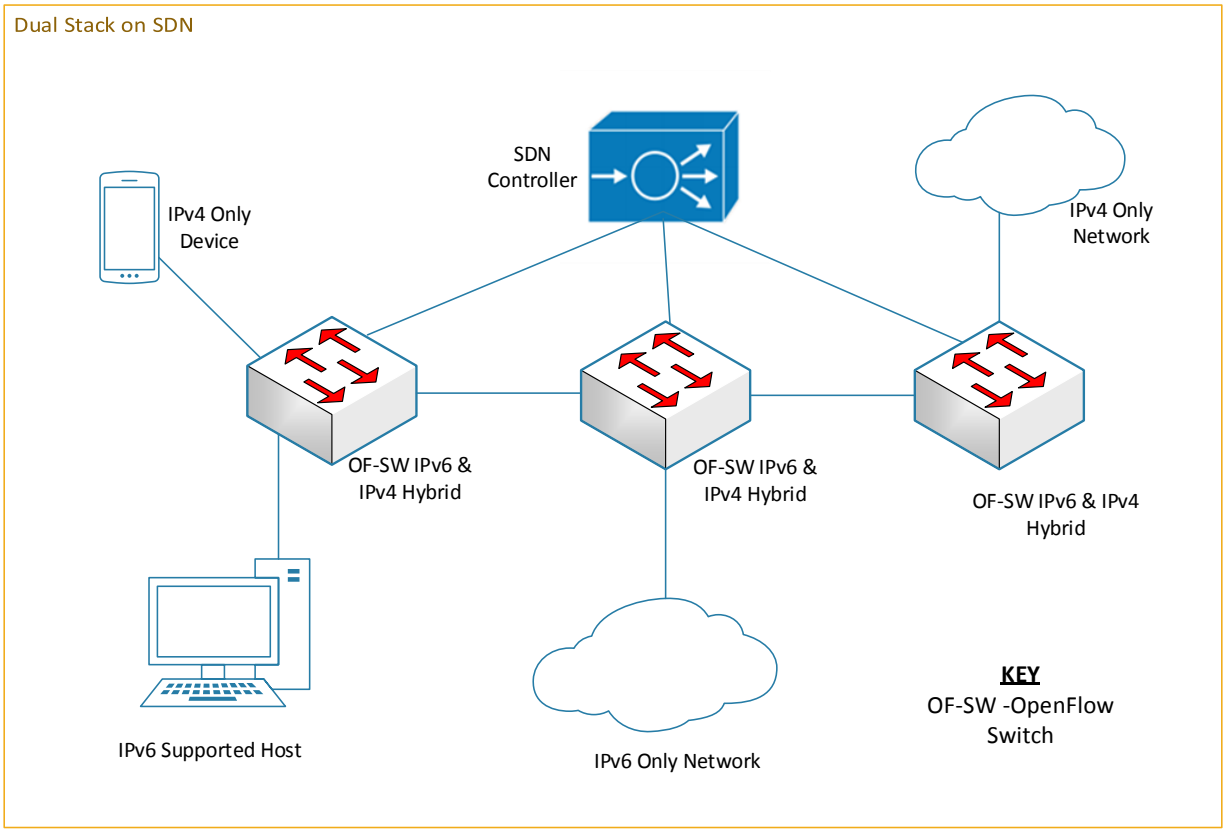

Figure 8 Network Architecture of IPv6 Migration Over SDN

3.3.2 IPv6 Tunneling

Tunneling is the technology chosen for IPv6 transition in this research work. In the Figure below, two IPv6 networks are connected over an IPv4 network via a GRE tunnel in an SDN network environment. Each of the network consists of an OpenFlow switch connected to two IPv6 hosts. The hosts on either side of the tunnel should be able to communicate effectively. The GRE tunnel across the IPv4 network also provides security to the IPv6 traffic through encapsulation of data packets. The proposed Figure 9 below shows the architecture of the GRE Tunnel.

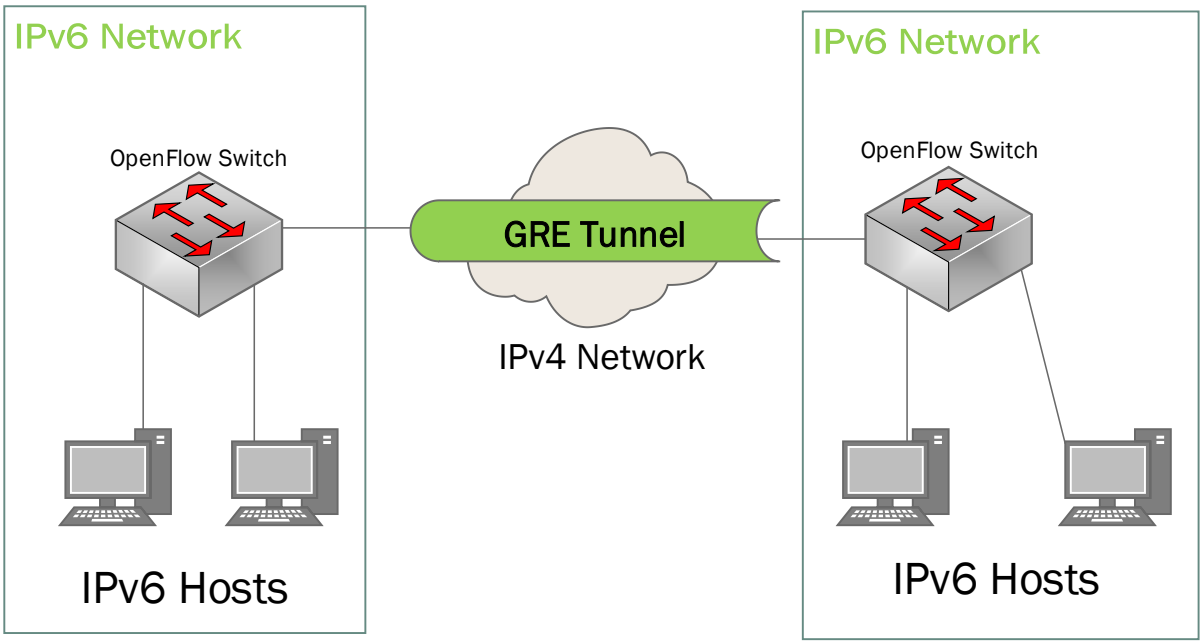

Figure 9 GRE Tunnel Architecture

3.3.3 Translation

IPv4 addresses can be translated to IPv6 or vice versa in 5G networks. Network Address Translation (NAT) is a mechanism that uses a gateway to translate between the two protocols. The translator requires a 32 bit IPv6 address space and an IPv4 address. The gateway can either be for NAT 64 or 46 i.e from IPv6 to IPv4 or vice versa. This mapping between the two protocols by the gateway can either be manual or automatic.

3.4 SDN-DHCP Architecture

The concept of DHCP is discussed in 2.6 as it has been used in networks before. The DHCP function in 5G networks is with the controller. The DHCP pool of IP addresses to be issued to network devices is managed by the controller. DHCP is one of the network functions that can be virtualized in a 5G network and managed by the controller. DHCPv6 will be used to allocate IPv6 addresses to devices in the network. Stateless Address Autoconfiguration (SLAAC) is used by IPv6 hosts in the network to generate required IP addresses.

In the proposed Figure 10 below, the SDN controller is connected to OpenFlow switches and the DHCP server. When devices connect to the network, they request for IP addresses and are allocated by the DHCP server via the controller. The DHCP server also computes the lease times for the issued IP addresses. The duration of the lease times depends on the priority of the device in the network times for the issued IP addresses. The duration of the lease times depends on the priority of the device in the network.

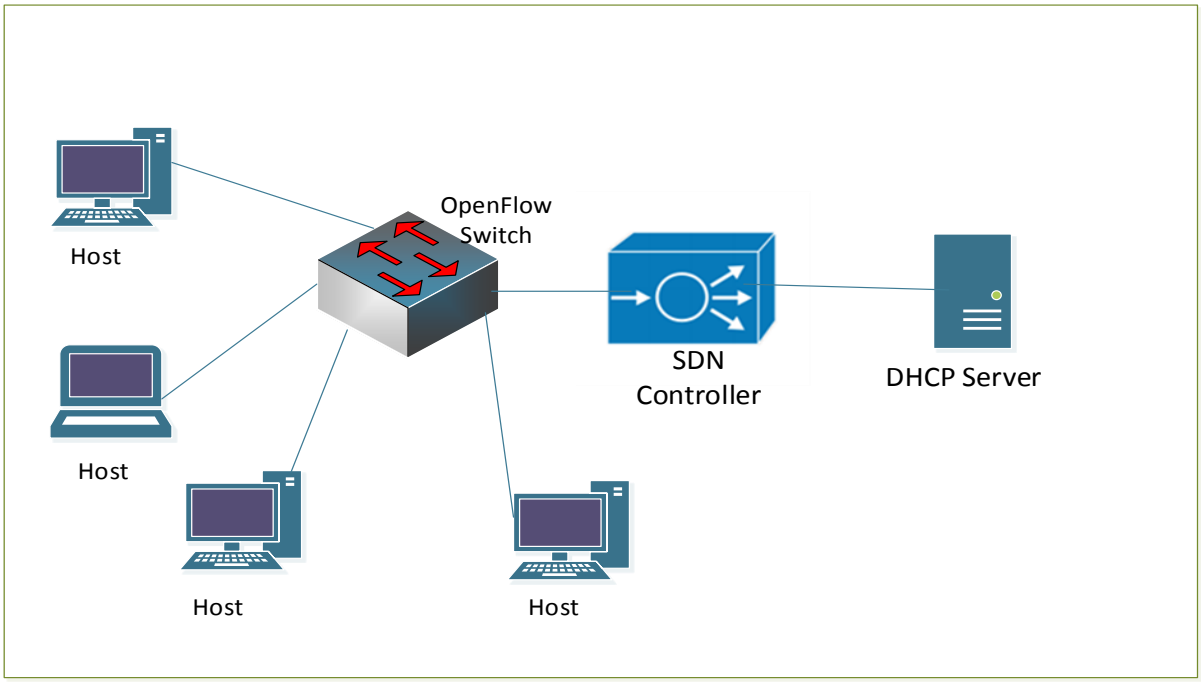

Figure 10 DHCP Server in SDN

3.5 Proposed Security in SDN

Security must be ensured when building future networks which will be SDN based. IP filtering is a method that is used to secure networks whereby devices are allowed or denied access to a network based on their IP address. This filtering can be done on different sections of an SDN network. A firewall can be used to allow or deny access of devices to the network. The firewall function is one of the Virtual Network Functions (VNFs) that can be virtualized in an SDN network. This VNF will be managed by the SDN Controller. IP filtering can also be implemented on the network controller to control access of devices. In SDN networks, OpenFlow switches have flow tables that affect how IP traffic moves. Flow tables in OpenFlow Switches can be used to secure networks. This can be done by forwarding traffic based on VLAN tags, incoming switch port, TCP port/IP addresses and MAC addresses.

IPsec and DHCPv6 shielding can be used in 5G networks to provide security to network devices. IPsec will secure traffic being forwarded on the network using tunnels. DHCPv6 shielding will prevent rogue DHCPv6 servers from causing DoS attacks on the network. These two security

strategies have been discussed below.

3.5.1 ACLs and AAA Protocol

Access Control Lists(ACLs) can be used to filter IP addresses to control SDN network flows. ACLs can be used to control traffic in Virtual Network subnets in SDN. The network Controller can be tasked to manage Access Lists. Remote Authentication Dial-In User Service (RADIUS) server can be used as a Network Access Control (NAC) method for authentication in the network. This method provides a sign on where a device requiring network access must have the right login details on the OpenFlow switch. The OpenFlow switch in SDN is the authenticating device giving information to the RADIUS server. The server in turn confirms if those login details are right by communicating back to the switch for it to either to avail or deny resources to the device. If the access capabilities of the device changes, the RADIUS server entry must be updated.

Authentication, Authorization and Accounting(AAA) is used to track IP traffic patterns and control access of devices to the SDN network. This function should be implemented as a network server. Authentication allows or denies network devices access to resources through login requirement, time restrictions and location restriction. Authentication mechanisms used are Password Authentication Protocol(PAP),Challenge-Handshake Authentication Protocol and Extensible Authentication Protocol (EAP). The network is programmed to automatically add or remove authorized or unauthorized devices based on their authentication. Authentication leads to authorization of devices on the network for various kinds of services. In Accounting, records of the SDN network activity are kept for a certain period. Network device behavior is also tracked to see if there is any malicious activity on the network. The network controller can be used to manage AAA in the OpenFlow switches in SDN.

3.5.2 Network Reconfiguration

MTD dynamically changes IP addresses to secure SDN based networks. This approach is applicable to cloud environments that support Multi-tenant networks to make them unbreachable. MTD brings unpredictability to Virtual Tenant Networks (VTNs) by changing their topology from time to time to confuse attackers. Attackers normally use software tools to scan through the network and identify potential points of weakness. Mutation leverages mapping of real and virtual IP addresses to defend networks. The unassigned IP addresses are used randomly as vIPs in the tenant networks. The SDN Controller manages the mutation process amongst VTNs centrally throughout the network. When the network starts the SDN Controller must learn the network topology and the IP address information. The SDN Controller shuffles the Virtual IP addresses of the subnets to ensure security in the network is maintained.

The Controller adds or deletes flows in the flow tables of OpenFlow switches to effect the

expected changes. DNS translates Domain Names to numerical IP addresses to identify and locate network devices and resources. The SDN Controller also responds to DNS requests through the internal DNS (iDNS) server to provide VTNs with the required name resolution. The Controller also sets the TTL value for the DNS response. After a certain time T_m, the virtual IP will be changed thus causing confusion to an attacker. The server is not exposed to attackers because it cannot be accessed by VTNs. The internal DNS comprises of several DNS servers that carry information that is specific to given tenants in the network.

The Controller is aware of the interfaces and functions in all the VTNs. It also notes the number Open Virtual Switches (OVSs) in a cloud environment. The flow tables in OpenFlow switches are populated by the SDN controller based on the desired traffic flows. When a packet arrives on an OpenFlow switch, it is examined to determine whether it is to be dropped or forwarded to a port based on flow table entries. It is the SDN Controller that instructs the switch on what action to take.

3.5.3 IP Security (IPsec)

The Internet Protocol security ensures authentication and encryption of packets in a network. In this model, two hosts or networks sending data to each other must have mutual authentication at the start of a communication session and use cryptographic keys for negotiation during the session [39]. IPsec can be deployed between a host and a host or a host and a network or a network and a network.

The Figure 11 below shows various hosts connected to an SDN network with IPsec being used for secure communication. Client 1 can communicate securely through the SDN network with subnet A or B using Tunnel 1. Similarly, Client 2 can communicate securely using Tunnel 2 with both subnets.

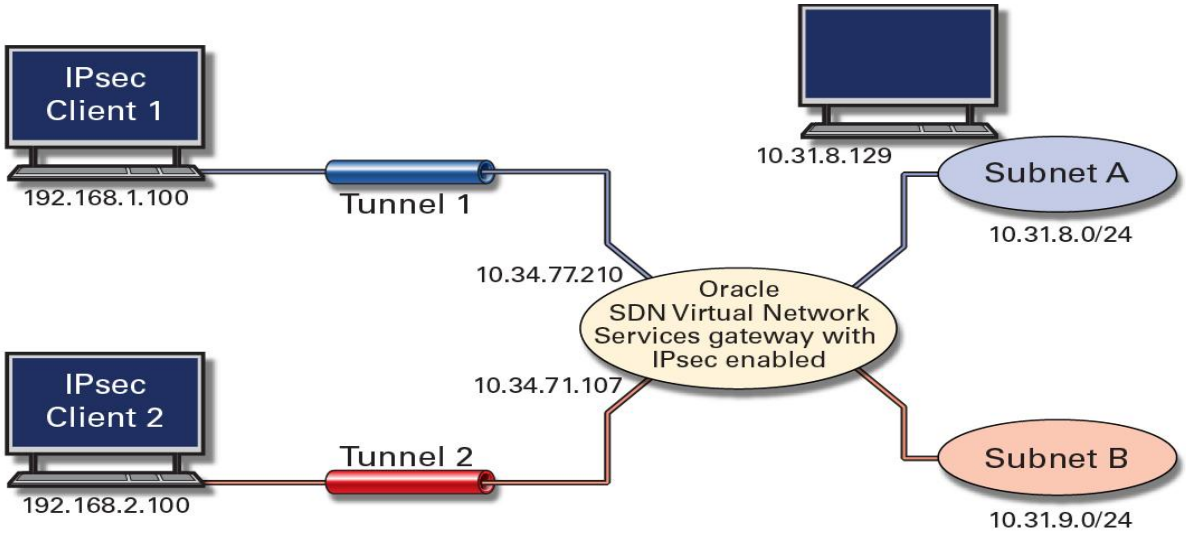

Figure 11 Application of IPsec on SDN [40]

3.5.4 DHCPv6 Shielding

DHCPv6 shielding is a way of protecting an IPv6 network against a Rogue DHCPv6 server. It protects the network from untrusted DHCPv6 traffic by allowing entry of packets only at a given port. In the Figure 12 below, the device used by the Attacker imitates the DHCPv6 server and sends wrong packets to the victim device. If not prevented this can lead to the Denial of Service to the Victim device.

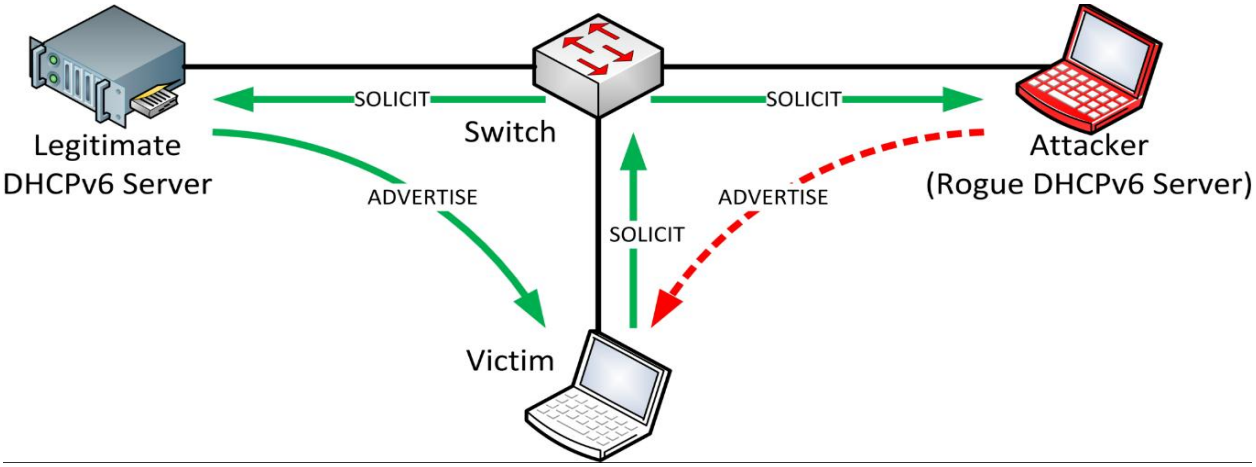

Figure 12 Legitimate DHCPv6 Server and Attacker (Rogue DHCPv6 Server) [41]

3.6 Chapter Summary

IPv6 addressing is key to the success of 5G networks since its capacity can support numerous devices because of the rapid growth of the Internet. SDN is the technology that will build and bring easy programmability to 5G networks. This technology must incorporate well IPv6 addressing to ensure reliable communication amongst devices in future networks. IPv4 addresses will still be used in networks until IPv6 is fully implemented. The three planes of SDN-IPv6 Architecture are; Data, Control and Service planes. They are discussed in brief in this chapter.

The architecture of two IP transition technologies; Dual Stack and Tunneling have been given in this chapter. The Dual stack architecture connects the IPv4 only and IPv6 only network with IPv4 and IPv6 hybrid OpenFlow switches to ensure seamless communication. A network model connecting two IPv6 networks over an IPv4 network via a tunnel is also discussed. This is to run IPv6 over an IPv4 network. Translation has also been mentioned in brief in this chapter. This Chapter also explains in brief the architecture of an SDN network with DHCP being used for dynamic IP addressing. The role of the SDN controller in dynamic addressing of future networks is discussed. There are two kinds of DHCP; version 4 and 6. DHCPv6 works similarly to DHCPv4 only that the former is for an IPv6 based network. DHCPv6 is used to issue dynamic

IPv6 addresses on an SDN based network.

This Chapter also discusses the proposed security solutions for future networks. The use of ACLs and AAA to ensure access control in SDN networks is discussed. The proposed implementation of these methods in SDN based networks is given. Network Reconfiguration can be used to enhance security further in 5G network environments. MTD can be used to bring unpredictability to networks thus make it difficult for attackers. IPsec, a technology which can be used to secure communication between two hosts or networks in SDN networks is discussed. Finally, DHCPv6 shielding which protects the network from Rogue DHCPv6 servers is also discussed.

CHAPTER 4

4 PROOF OF CONCEPT IMPLEMENTATION

4.1 Introduction

In this Chapter, the Proof of Concept (PoC) implementation for the SDN-IPv6, SDN-DHCP and IPv6 tunneling architectures is given. The Open-source project OpenDaylight is used to provide the Software Defined Networking(SDN) controller, applications and Application Programming Interfaces (APIs) for building an IP addressing and transition solution for 5G networks. The architectures presented in Chapter 3 for IPv6 addressing, Dual Stack, DHCP and IPv6 tunneling will ensure efficient communication in next generation networks. An elaborate discussion is also made on the implementation of IPv6, DHCP and IPv6 tunneling in an SDN network environment. IP security is also proposed using tunneling for secure communication. Mininet is used as a platform for building the network and showing IPv6 addressing functioning in SDN. This Chapter discusses the configuration and elements of IP addressing and transition in an SDN network.

This chapter is organized as follows; 4.2 gives the various tools and technologies that are used in the Proof of Concept (PoC) implementation including a discussion on Virtual Private Network and Virtual Tenant Network. The design of the IP addressing and transition solution is discussed in section 4.4. Finally, the Chapter summary is given in section 4.5.

4.2 Tools and Technologies

There are various tools and technologies that are used in the PoC for IP addressing and transition implementation in 5G networks. Mininet is used as a platform to build the SDN network. The OpenFlow switches are used as data forwarding devices in the network. The SDN controller from the ODL project of the Linux Foundation is used. The built network topology is viewed on OpenDaylight. These tools and technologies are discussed below.

4.2.1 Mininet

This is an SDN network emulator that is used to build large virtual networks on a single machine. On Mininet, large network topologies can be created easily and fast. This platform provides for scalability, ease of use and performance accuracy in network emulation. Custom network topologies can be built using the python API and run on Mininet to analyze them. The network built uses lightweight virtualization technology. The SDN controllers modelled can

be transferred to hardware with minimal changes for execution. Mininet can be integrated into GNS3 as one of the hosts in the simulated network to build SDN network topologies.

4.2.2 Virtual Machine Player Workstation

VM Player workstation is used as the virtualization engine to host Mininet so that SDN network topologies can be built. The bridged adapter of VM Player is used to connect Mininet to the network of the Linux host machine. In the case of the tunneling model, which is discussed later in this chapter, two instances of the virtualization engine are run simultaneously. On each of the instances networks are built with the aim of demonstrating how IPv6 tunneling works over IPv4.

4.2.3 OpenDaylight(ODL)

This is an open source SDN project that is hosted by the Linux Foundation and encourages research in SDN to come up with solutions [42]. It gives a platform to directly deploy SDN without the need of other components. In early 2014, OpenDaylight announced its first release "Hydrogen" and has had four other releases since then. The latest release of ODL is "Oxygen". OpenDaylight has the support for the OpenFlow protocol but also supports other protocols.

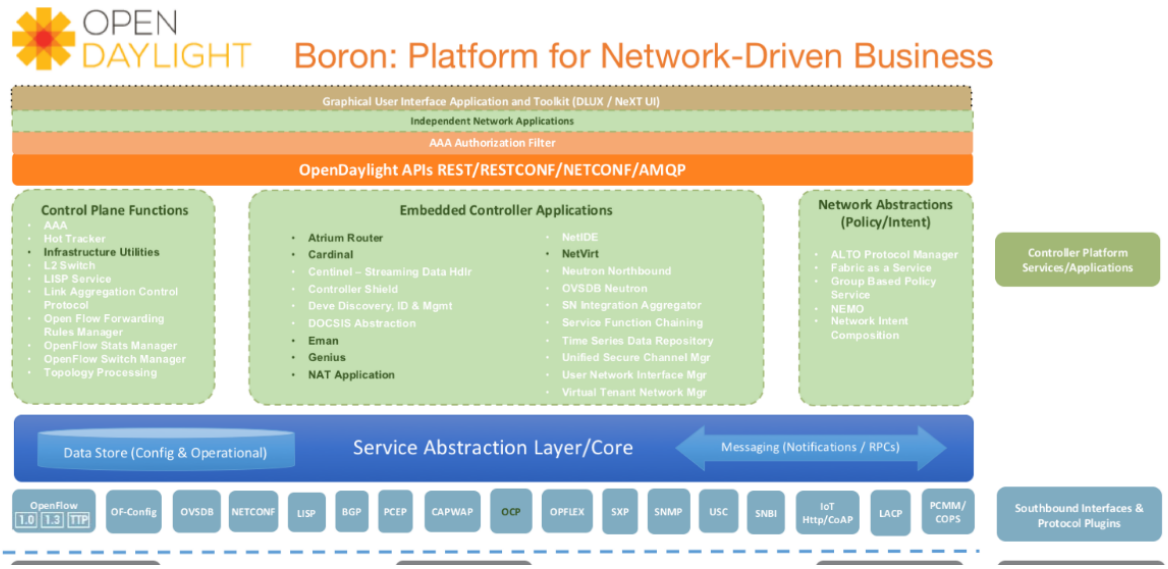

Figure 13 ODL Architecture [42]

4.2.4 ODL Controller

The OpenFlow protocol ensures the ODL controller effects changes in the network via the forwarding plane. This gives tenants the ability to adapt to the dynamic nature of the network and can manage their networks. The ODL controller is deployed in various network environments and is able support other protocols as well. Northbound APIs which support

communication with applications are exposed by the ODL controller. The applications in the network use algorithms to collect information about the behavior of the network and then use ODL controller to bring about required changes. The ODL controller is software based and resides within a Java Virtual Machine (JVM).

4.2.5 OpenFlow Switches

These are switches that support the OpenFlow protocol. Flows within these switches are determined by the OpenFlow protocol. In OpenFlow switches the part of the switch that forwards data (Data Plane) is the only one present on the switch. The Control plane is abstracted from the switch and is implemented in the form of an SDN controller in the network. The controller makes intelligent decisions on behalf of the OpenFlow switches and has visibility of the network. One of the decisions made is how flows will occur amongst the switches in the network. The switches maintain the flow tables as records to determine the next destination of packets in the network. The OpenFlow protocol facilitates the communication between switches and controllers in an SDN network.

The Open vSwitch is an OpenFlow switch that has been virtualized. Other than OpenFlow, it also supports switch management protocols. Traditional switches either operated on Layer 2 or 3 of the OSI model. That is not the case for OpenFlow switches as their forwarding decisions solely depends on IP address, Input port, mac address and VLAN ID [6].

4.2.6 Graphical Network Simulator-3(GNS3)

GNS3 is a network emulator that began mostly as a Cisco-centric tool but overtime it started being used for designing networks for other vendors. GNS3 works on Linux, Windows and Mac OS. Out of these, Linux is recommended because GNS3 supports Linux Kernel features that increase performance and stability. Internetworking Operating System (IOS) and IOS on Unix (IOU) based devices can only run on Linux. It allows for network emulation using real and virtual devices. GNS3 is used for Proof of Concept implementations and network testing.

4.2.6.1 Ubuntu Docker Container

This is a Debian based container that makes it easy to run an application on the network. The Ubuntu Docker container can be used as a remote controller in the network. The topology being built can be pointed to the IP address of this device for display on OpenDaylight. The Ubuntu Docker container can be connected to Mininet network via a switch on GNS3 network emulator. The Ubuntu Container receives an IP address via the DHCP lease.

4.2.6.2 NAT Cloud

This node ensures that the network being simulated is connected to the internet without any configuration via Network Address Translation (NAT). However, the simulated network cannot be accessed from the internet. NAT cloud is useful when downloading data from the internet. This cloud acts as a DHCP server and issues IP addresses to hosts in the GNS3 network from a given range specified.

4.2.6.3 Firefox Appliance

This is a Quick Emulator (QEMU) Virtual Machine that is a lightweight device and runs on TinyCore Linux with a preinstalled Firefox browser. It can be downloaded from the GNS3 Marketplace and used as a host in the network. It serves as a Graphical User Interface (GUI) device for viewing the built topologies on OpenDaylight. It receives its IP address from the NAT cloud through DHCP lease and connects to the internet.

4.2.7 APIs

Southbound APIs are used to ensure communication between the Infrastructure layer and the Control layer of the SDN Architecture (reviewed in section 2.4) whereas NorthBound APIs facilitate communication between the Control layer and the Application layer. A common NorthBound API is the Representation State Transfer (REST) API which uses HTTP/HTTPS for identification in communication via the IP address of the SDN Controller. OpenFlow is the most common protocol for the SouthBound APIs. Network Configuration Protocol (NetConf) uses XML language to communicate with network devices to make configuration changes.

4.2.8 Virtual Private Network (VPN)

This is a technology that ensures secure communication between devices in the network through authentication and encryption. It protects private network data over an untrusted public network. In IP addressing, data is secured by running IPv4 over an IPv6 network or vice versa. In this research work, IPv6 is run over an IPv4 network using a tunnel. IPsec is where authentication is required before communication and encryption keys are exchanged in the process of the session between two hosts or networks or a host and a network to protect secured data. As discussed in Chapter 3, a tunnel is used as a conduit for data from one host or network to another. There are various data encryption techniques that can be used namely; Advanced Encryption Standard (AES), Data Encryption Standard (DES), Authentication Headers Encapsulating Security Payloads (AH-ESP), ESP and AH.

4.2.9 Virtual Tenant Network (VTN)

This is a network that hosts various virtual tenants on an SDN controller. This is known as a multi-tenancy in networks; virtual networks are created on Open vSwitches. ODL Controller contains the following VNFs; vLink, vBridge and vTunnel [6]. The virtual tenants can be classified into subnets; therefore, network IP addresses can be used to separate them. Each of the subnets have virtual hosts that are also identified using IP addresses. The Control plane is separated from the Data plane in the VTNs. This abstraction enables users of the network to deploy their own virtual subnets without being limited by bandwidth. The VTN Manager and Coordinator are the components of the application. The VTN Manager is implemented as a plugin to the OpenDaylight Controller and provides a REST for VTN components. The VTN Coordinator is an application that gives a REST interface to the user for VTN virtualization.

4.3 Alternative Tools and Technologies

There are other tools and technologies that could be used for Proof of Concept implementation in place of the ones discussed above. EstiNET and OFnet are SDN network emulators that offer functionalities similar to Mininet. Virtual Box can be used as a virtualization engine in place of the VM Player workstation. As for the controller, Ryu and Pox controllers can be used as an alternative to the ODL controller. Windows and Mac OS operating systems can be used instead of linux for the experiment.

4.4 Design

This section discusses how the tools and technologies in 4.2 have been used in PoC implementation in this research work. The tools presented are used to setup IPv4 and IPv6 networks on Mininet to simulate SDNv4 and SDNv6 networks. An IPv6 tunnel is created to run over the computer's IPv4 network. The design of these implemented models is discussed below.

4.4.1 SDNv6 Network

IPv6 addressing has been supported since OpenFlow version 1.2 protocol even though this version does not allow for all the IPv6 features. Other important features were added in the OpenFlow version 1.3. Therefore, the Mininet releases having OpenFlow version 1.3 and beyond support building of networks with IPv6 hosts.

The Figure 14 below shows four IPv6 hosts connected to two OpenFlow switches on Mininet. IPv6 flows are allowed on the OpenFlow switch which is connected to a remote ODL Controller. The Mininet Virtual machine is used on a Linux machine to build the desired IPv6 network. The topology that is built on Mininet emulator is viewed on OpenDaylight. ODL controller has complete view of the network devices. Hosts that support IPv6 addressing are connected to the OpenFlow switches. Communication amongst the hosts in the network is

confirmed if they can be able to send ICMPv6 messages to each other.

The IPv6 network custom topology created using a python script is run on Mininet. To confirm that the hosts have picked up IPv6 addresses, their xterms(terminal emulators for hosts) are started and their IP addresses are checked. Successful communication between hosts is confirmed from their individual xterms by issuing ICMP requests.

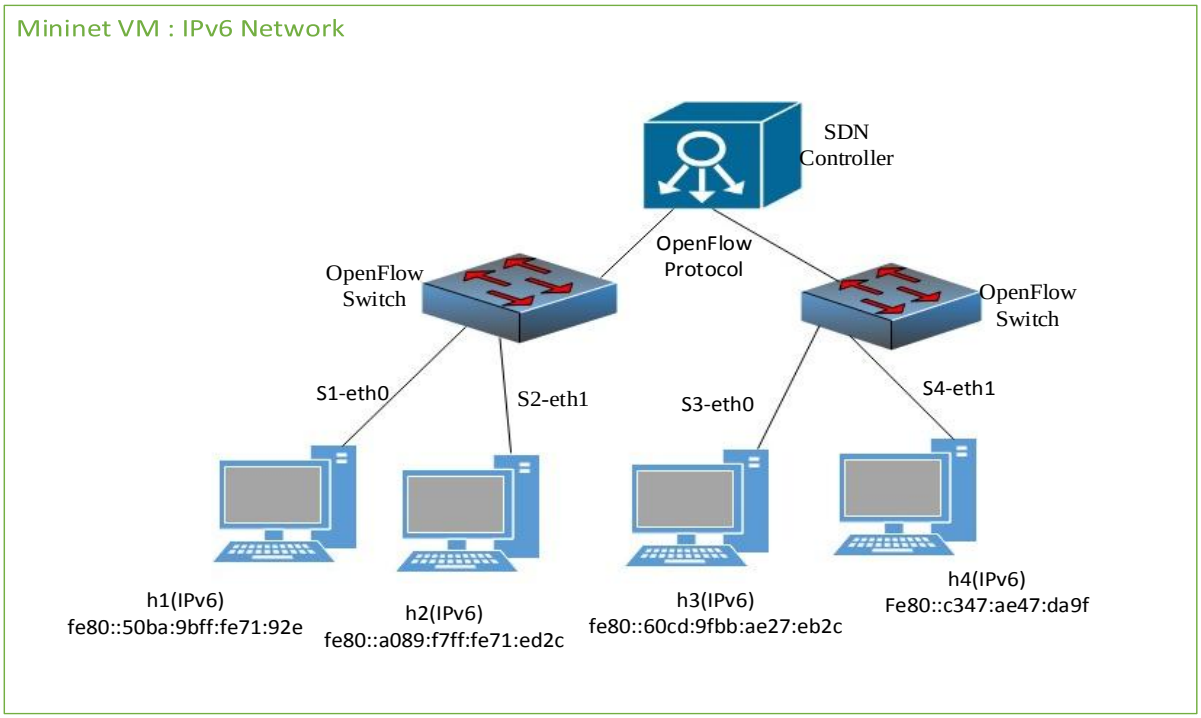

Figure 14 Topology of 2 switches and 2 virtual hosts simple network on Mininet VM

4.4.2 IPv4 Network

5G networks will still host devices that run IPv4 until IPv6 addressing becomes fully incorporated. The IPv4 network is built and connected to Mininet on a GNS3 network emulator. This network has devices that only support IPv4 addressing and can connect to the Internet. The Mininet network can be able to connect to other IPv4 hosts and to the Internet. The switch used on GNS3 is a vendor specific one and does not support the OpenFlow protocol. The Ubuntu Docker Container and the Firefox host are also connected on the network. These two hosts are able to access the Internet and the DHCP function in this IPv4 network is with the NAT Cloud. All the hosts on the GNS3 network receive dynamic IPv4 addresses from the NAT Cloud.

The Mininet host machine is used to build the SDN network on GNS3. Mininet by default issues dynamic IPv4 addresses to hosts from a predetermined range of IP addresses (10.0.0.0/8). However, the network can be configured to receive the desired dynamic IP addresses from an external DHCP server. In doing so, the IP address range can be changed from time to time. The Figure 15 below shows how the IPv4 network was setup on the GNS3 emulator.

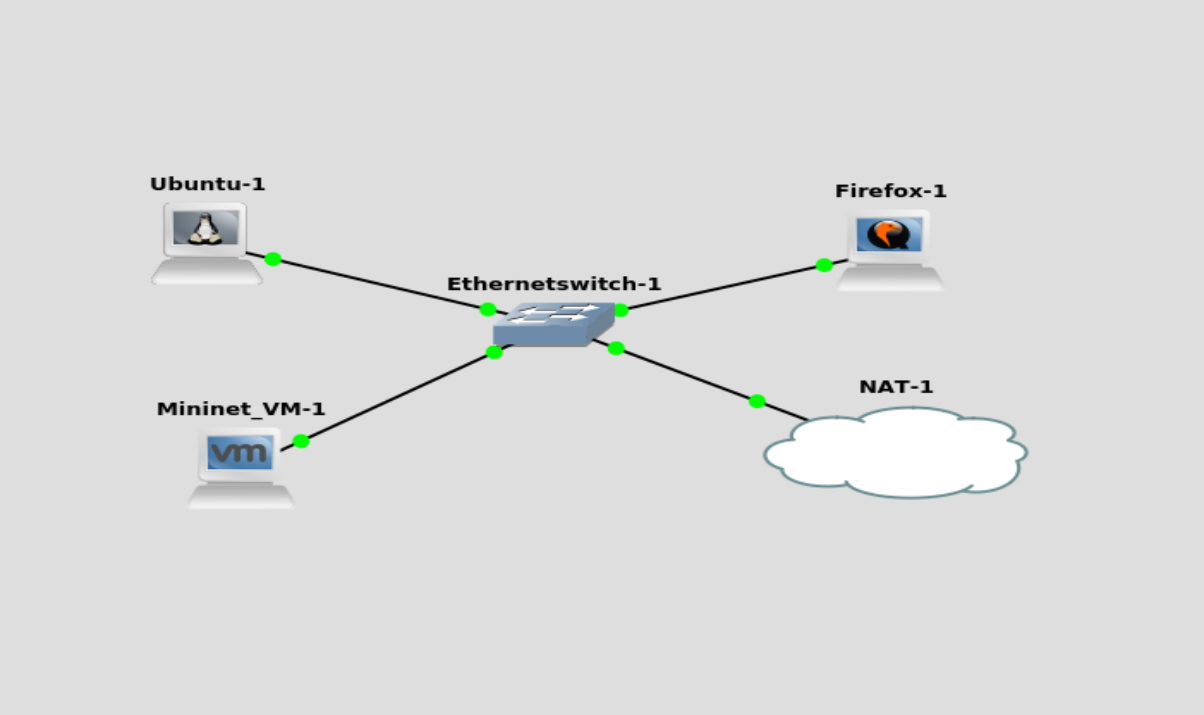

Figure 15 IPv4 Network Setup on GNS3

4.4.3 IP Transition Technologies in SDN

There are three Transition technologies; Dual Stack, Tunneling and Translation as explained in Chapter 2 that are used to ensure transition from IPv4 addressing towards IPv6 only networks. From these technologies tunneling is the one that is implemented in a network model in this research work. How each of these technologies are used in an SDN network is discussed in this dissertation.

4.4.3.1 Tunneling

Tunneling as discussed in Chapter 2 is another approach that ensures IPv4 and IPv6 are run simultaneously in a network. IPv6 traffic can be routed over an IPv4 network and vice versa by creating tunnels as shown in Chapter 3 section 3.3. Virtual Extensible Local Area Network (VXLAN) is an encapsulation protocol that will ensure creation of a logical network for virtual machines in the different created networks in SDN [12]. GRE tunneling can also be used to run IPv6 addresses over an IPv4 network. This research work leverages the use of a GRE tunnel to connect two IPv6 networks over an IPv4 computer network. This technology will support the deployment of cloud computing to support multi-tenant environments in future networks.

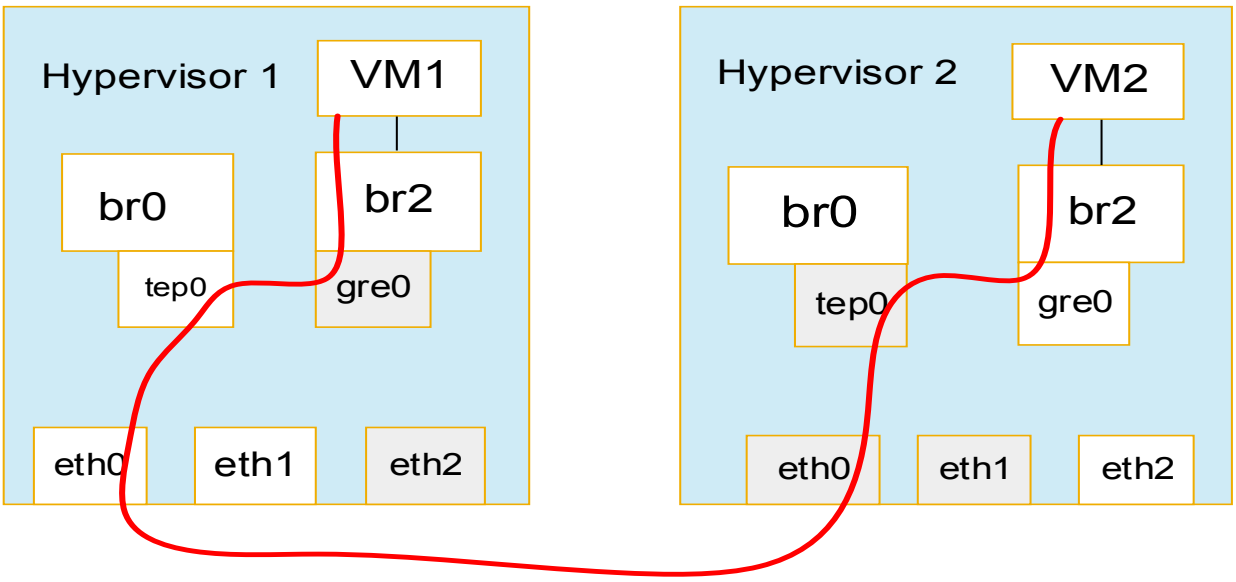

Figure 16 GRE Tunnel connecting two VMs on VM Player Workstation

Two separate IPv6 networks with two hosts each are created on two separate VMs and connected using a GRE tunnel as shown in Figure 16 above.

4.5 Chapter Summary

To validate the proposed IP addressing and transition solution for 5G networks, this Chapter presents the PoC implementation. The tools and technologies that are used to implement the proposed solution are discussed. The GNS3 network emulator is used to incorporate Mininet as a Docker container in it to build a sample network for the PoC implementation. SDN topologies are built on the Mininet Docker container which is connected to other devices via a legacy switch and ODL is used as a controller in the network. The NAT cloud performs the DHCP function and issues IPv4 addresses to the network devices and the network is connected to the Internet. IPv6 only network is built on Mininet to show communication between SDNv6 network and the IPv4 network.

This Chapter emphasizes the importance of IP addressing whether IPv4 or IPv6 in identifying and securing network devices. The PoC implementation given provides an example of a future network that will be anchored on IP.

CHAPTER FIVE

5 EXPERIMENTAL PERFORMANCE EVALUATION AND VALIDATION

5.1 Introduction

Chapter 4 explained how the various tools and technologies are used in the PoC implementation for the IP addressing and transition solution for 5G networks. To demonstrate this IP addressing and transition solution in 5G networks, two experiments are undertaken. The first is the simulation of IPv4 and IPv6 addressing models in an SDN network environment to show the working of IPv4 and IPv6 in networks. The second one is the use of a GRE tunnel to carry traffic between two IPv6 networks to show how IPv6 protocol can be run over IPv4 using tunnels.

This Chapter provides the analysis and evaluation of the IPv4 and IPv6 network models. The analysis of IPv4 and IPv6 SDN based networks connected on a GNS3 simulator is done. The communication amongst the IPv4 and IPv6 addressed network hosts is tested. The network is analyzed and evaluated based on four performance measurement indicators. These are ICMP requests, Transfer, Delay and Throughput. Tunneling is one of the proposed IP transition technologies for future networks which is discussed in this research work. Tunneling of IPv6 over an IPv4 network is modelled in an SDN network environment. Communication between the two IPv6 networks connected via a GRE tunnel is analyzed and evaluated based on four performance metrics namely; ICMP requests, Delay, Transfer and Throughput.

5.2 Evaluation Metrics

ICMP messages are used to confirm the reachability of hosts between two end points in a network by sending error messages and operational information. Ping is a tool that generates ICMP requests and awaits ICMP echo responses. ICMP requests are also used to check the delay in a network link. The amount of data transferred, and throughput can be found using Iperf tool. Transfer is an Evaluation Metric used to check the actual amount of data conveyed between two network devices. It is important to estimate how much data is sent across two network end points to know the quality of the transmission. Finally, throughput is the amount of data sent across a network link per unit time. Throughput is measured under two transport layer protocols namely; TCP and UDP. TCP is reliable, bidirectional, flow managed, acknowledged, stream oriented and connection-oriented kind of protocol. On the other hand, UDP is connection-less, has very low overhead and very high transmission speed. Retransmissions are not performed in UDP and data delivery is on best effort criteria therefore not reliable. These two protocols must be compared in any transmission to know which one is suitable for specific network links.

5.3 Evaluation Scenarios

The first evaluation scenario is where the IPv4 and IPv6 networks are evaluated on SDN. The performance of these two networks is measured based on the evaluation metrics mentioned above. The second scenario is the tunnel link characteristics measurement. Similarly, the same metrics are used to check the efficiency of the GRE Tunnel created.

5.4 IPv4 Network

To evaluate the IPv4 network on the Mininet host machine connected on GNS3 emulator, the command *'sudo mn --topo tree, depth=2, fanout=4'* is used to create a topology with 5 switches and 16 hosts. 'sudo' is a command in unix-like operating systems that allows users to run programs with security privileges of a superuser. The program being run to start the topology is Mininet signified by 'mn'. Tree topology is one of the built-in topologies on Mininet. The '--topo' in the command chooses one of the desired inbuilt topologies and 'depth' refers to two branch levels. 'fanout' in the command creates two children per node. Mininet automatically issues IPv4 addresses using DHCP from the 10.0.0.0/8 address range.

5.4.1 IPv4 Network Results

5.4.1.1 IO Graphs

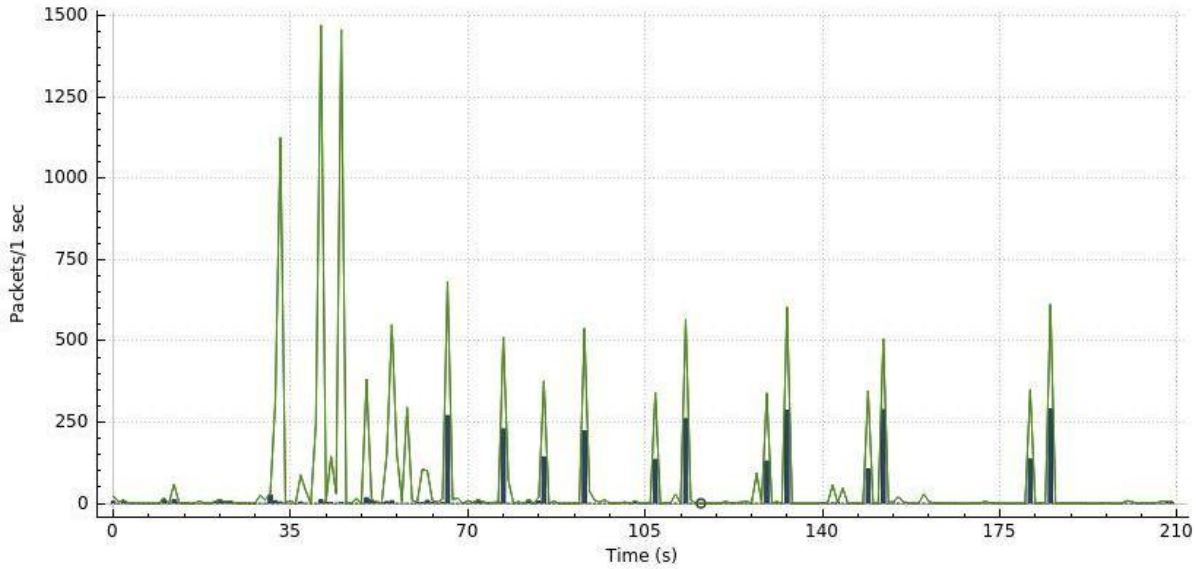

Figure 17 Traffic density in the Network before ICMP requests are issued

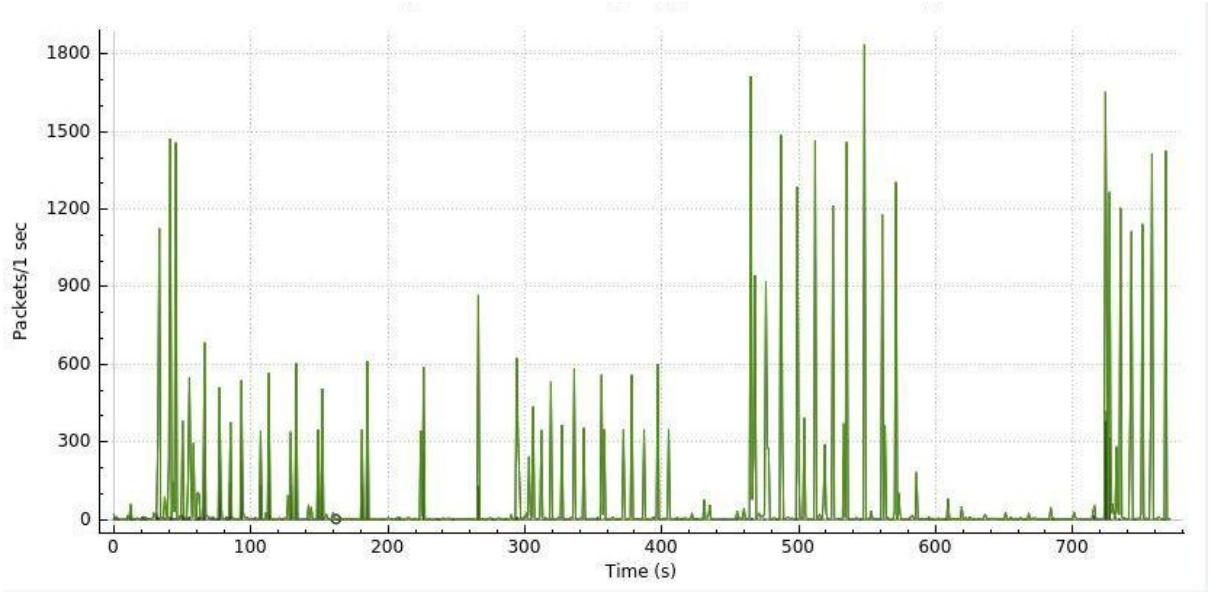

Figure 18 Traffic density in the Network after ICMP requests are issued

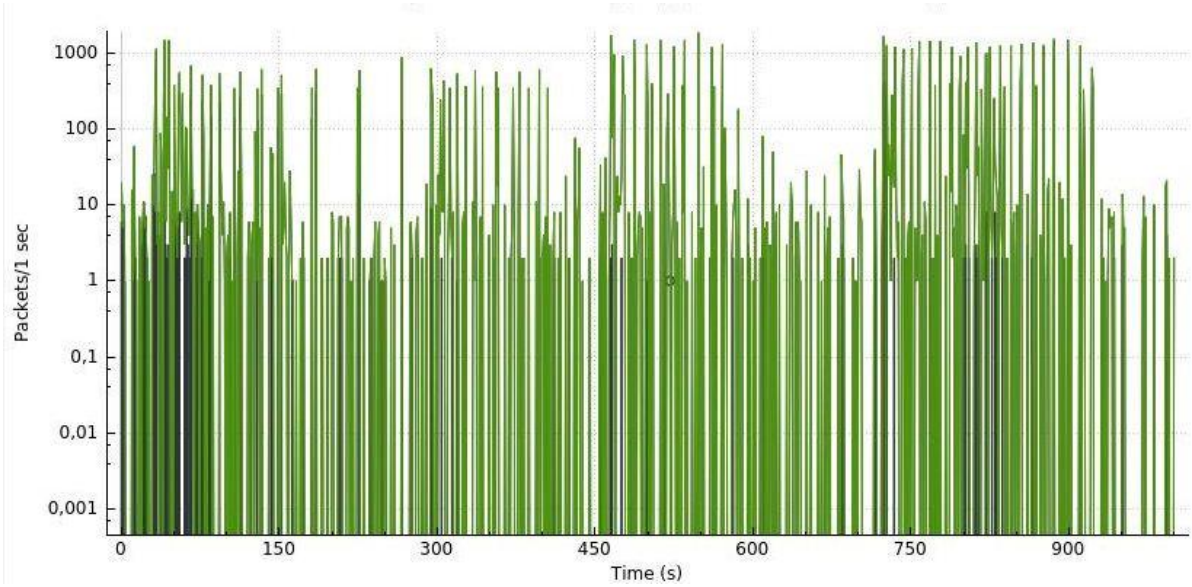

Figure 19 Traffic density viewed on the Logarithmic scale

5.4.1.2 Analysis of the Results

The graphs in Figures 17 and 18 show the traffic density in the IPv4 network before and after ICMP requests were issued respectively. Figure 17 represents the initial traffic in the network before ICMP requests are made and consists mainly of Solicitation messages, TCP messages (Ack,Seq,Win). After ICMP requests are issued between sampled hosts 7 and 15, traffic density increases in the network as represented by Figure 18 above. TCP errors can be seen in black by using the logarithmic scale as shown in Figure 19 above. These occurred because some packets are dropped during the transmission as they could not reach their destination due to the simultaneous ICMP requests between the sampled hosts and the initial traffic in the network.

5.5 IPv6 Network

To evaluate the SDNv6 network in section 4.5.1, a custom topology with 5 switches and 16 hosts was created with a Python script and run on Mininet. Manual addressing was used for IPv6 address allocation because the network hosts addressed using DHCPv6 could not be able to reach each other on the host machine. From the custom topology script, hosts received IPv6 addresses with the command *'host ifconfig host-eth0 inet6 add fc00::hostnumber/64'* as shown below for three sampled hosts.

```
h1 ifconfig h1-eth0 inet6 add fc00::1/64
h2 ifconfig h2-eth0 inet6 add fc00::2/64
h16 ifconfig h16-eth0 inet6 add fc00::16/64
```

The reachability of all hosts in the network was tested using the 'pingall' command and it was successful as shown in Figure 36 Appendix A section A.4.

5.5.1 IPv6 Network Results

5.5.1.1 IO Graphs

In the Figures 20,21,22 and 23 below, the traffic rate and TCP errors in the network is shown.

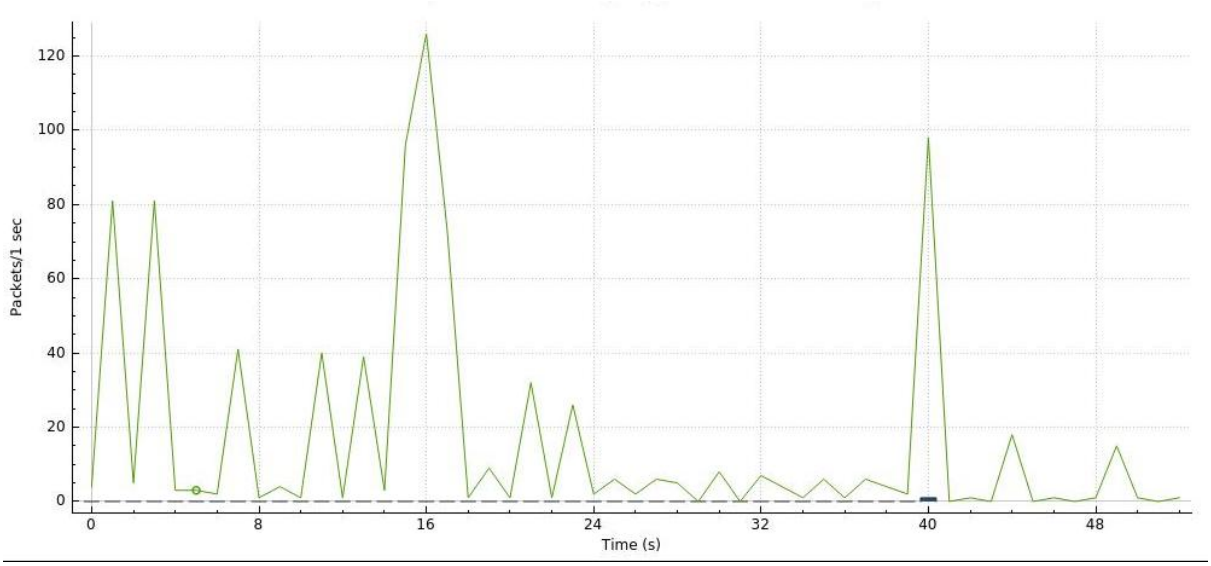

Figure 20 Traffic density in the Network before ICMP requests

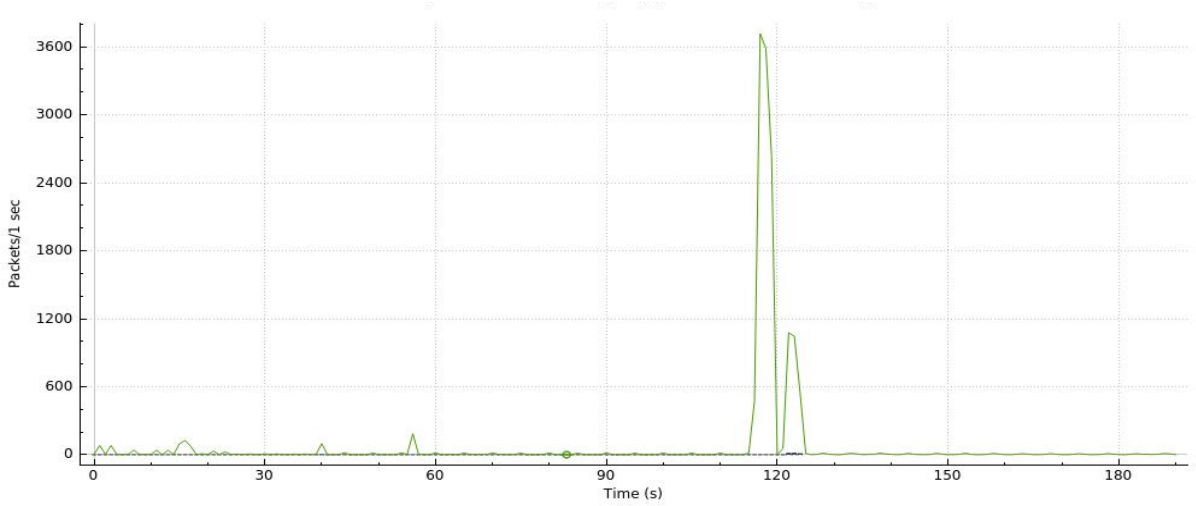

Figure 21 Traffic density in the Network after the ‘pingall’ command is issued

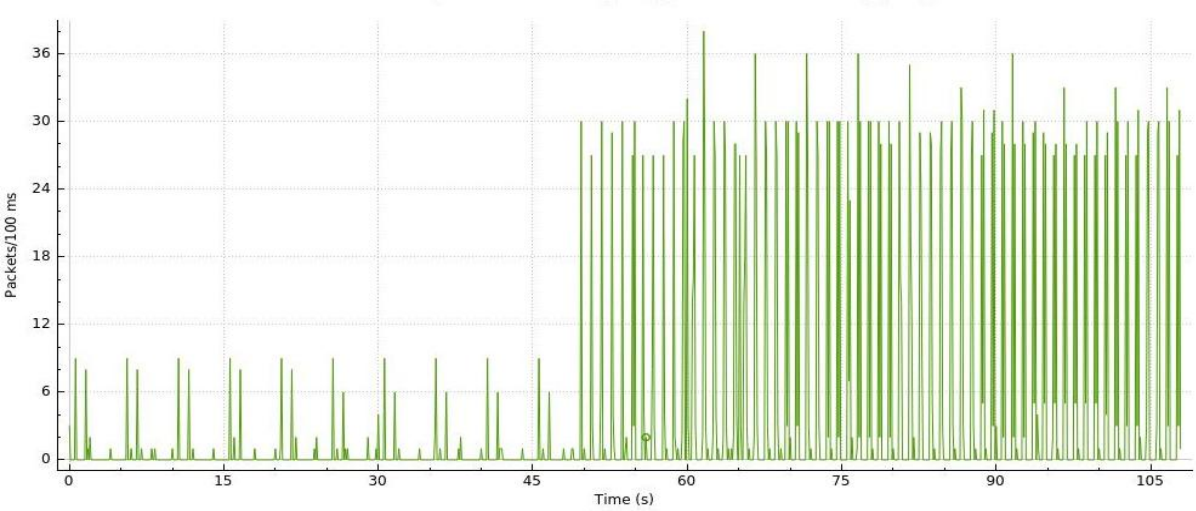

Figure 22 Increase in traffic density resulting from ICMP requests between Host 7 and 15

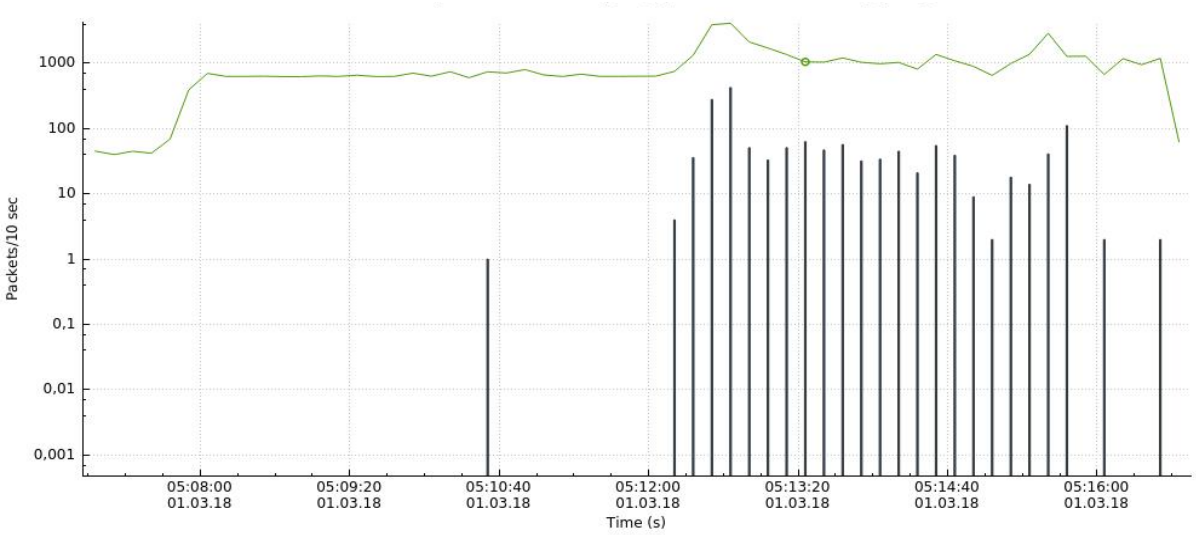

Figure 23 Representation of traffic density in the network using the Logarithmic scale

5.5.1.2 Analysis of the Results

Figure 20 above shows the IO graph for the network before ICMP requests are made. The traffic density in the network is below 125 packets/second. This traffic comprises mainly of solicitation messages, OpenFlow standard messages (OFPT) and TCP messages from the controller with IP address 127.0.0.1.

'pingall' is a command that tests connectivity amongst all devices in a Mininet network. When 'pingall' command is issued on the network to test connectivity of all the hosts, there is transmission of packets as seen in the two spikes in Figure 21 having a maximum density of 3700 and 1100 packets/second. ICMPv6 requests are also sent from Host 7 with address (*fc00::7*) to Host 15 with address (*fc00::15*) on the network and the result is shown in Figure 22. Before issuing the ICMP requests, the traffic density in the network was low but after it increased significantly as seen in the IO graph in Figure 22.

Figure 23 shows the graph of the traffic density on the network with a logarithmic scale which can show the fine details. TCP errors can be seen from time 05:10:40 onwards indicating some packets must have been retransmitted and others are dropped in the communication between Host 7 and 15.

5.5.1.3 Flow Graph

Wireshark was used to generate the Flow Graph for the traffic between the sampled hosts; host 7 and 15. The Flow Graph for ICMP requests between hosts 7(*fc00::7*) and 15(*fc00::15*) is as shown in Figure 24 below.

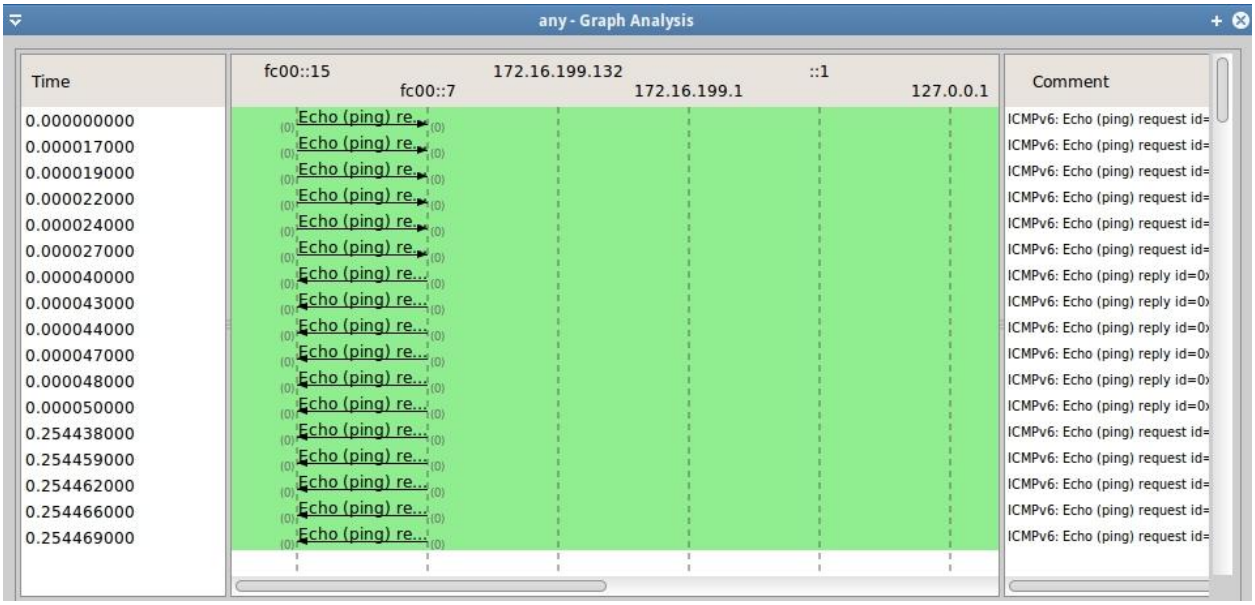

Figure 24 Flow Graph representing ICMP requests between sampled Hosts 7 and 15

As observed in the above Flow Graph traffic moves from IPv6 address *fc00::15* to *fc00::7* and back. The loopback address on the localhost machine for IPv4 is *127.0.0.1* whereas for IPv6 is *::1*. The loopback address 127.0.0.1 is also used as the network controller with port 6633.

5.5.2 Network Links Performance

The Figure 25 below shows four links; L1(s1-s2), L2(s1-s3), L3(s1-s4) and L4(s1-s5) that were created by the IPv4 and IPv6 network setup. Four OpenFlow switches s2, s3, s4 and s5 are all connected to the switch s1 in the Mininet tree topology created. Each of the switches apart from s1 are connected to four hosts as shown in the Figure 25 below.

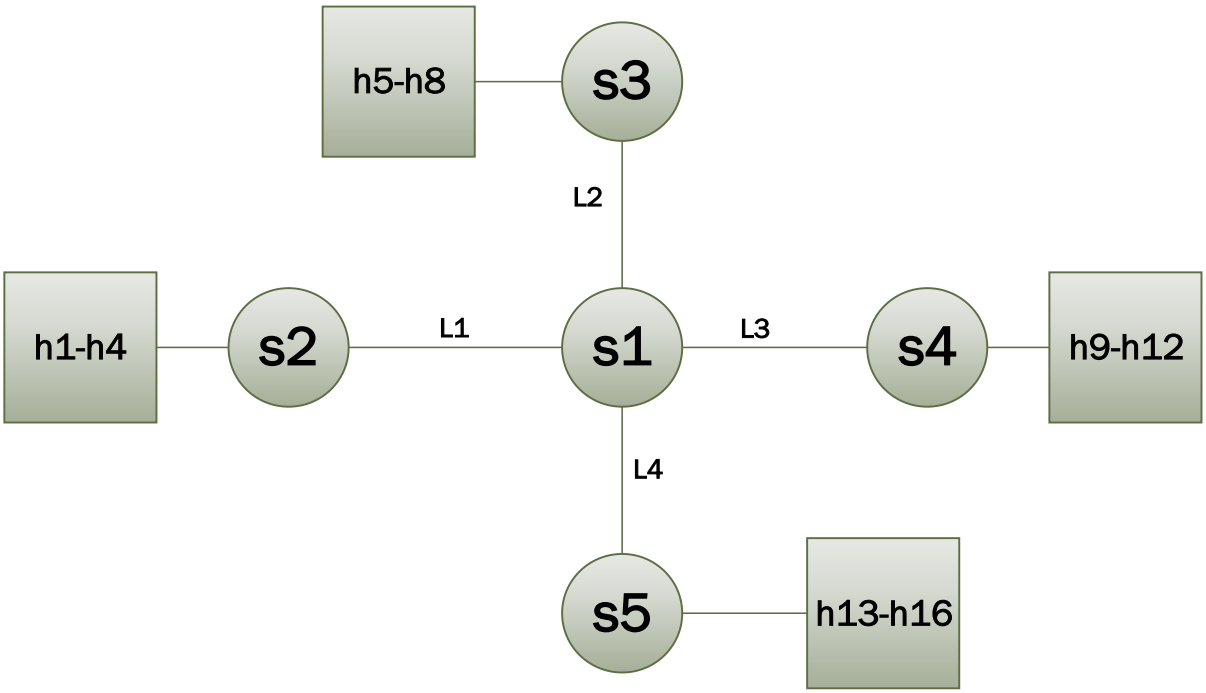

Figure 25 Network Links

5.5.2.1 IPv4 Network Links Performance

The performance metrics; TCP and UDP throughput, Latency, and Transfer for the IPv4 network setup was measured and recorded in the Tables 1 and 2 shown below. Iperf was used to send data streams across links on the network to find out TCP and UDP transmission rates whereas ICMP requests were used to determine delay in the network.

Table 1 TCP Network Links Performance

Link	Transfer(GB)	Throughput(Gbps)	Latency/RTT(ms)
L1	22.3	19.2	0.725
L2	22.1	18.9	0.693
L3	22.6	19.4	0.788
L4	22.5	19.3	0.753

Table 2 UDP Network Links Performance

Link	Transfer(MB)	Throughput(Mbps)	Latency/RTT(ms)	Jitter(ms)
L1	118	99.1	0.753	0.000
L2	118	99.1	0.821	0.001
L3	118	99.1	0.696	0.001
L4	118	99.1	0.767	0.000

5.5.2.2 Analysis of the IPv4 Network Results

The table 1 above shows that the TCP Throughput across all the links is almost the same. The TCP Throughput is impacted by latency and therefore is different for the four links. The link bandwidths are default and depends on the amount that can be handled by the computer. The Linux Machine that was used set the TCP throughput for the link at an average of 19.2 Gbps for the network links. UDP results shown in table 2 is the same for all the links. This could be because the performance of UDP is not impacted by latency therefore explains the constant rate for all the network links.

5.5.2.3 IPv6 Network Links Performance

The network performance metrics; Data transfer, Throughput and Delay/Latency (RTT) were measured. Iperf was used to check for TCP and UDP Data transfer and Throughput for the links whereas ICMP requests were used to check the Latency. The results obtained are represented in the tables 3 and 4 shown below.

Table 3 TCP Network Links Performance

Link	Transfer(GB)	Throughput(Gbps)	Latency/RTT(ms)
L1	32.8	28.1	0.051
L2	31.1	26.8	0.050
L3	30.6	26.3	0.051
L4	31.9	27.4	0.055

Table 4 UDP Network Links Performance

Link	Transfer(MB)	Throughput(Mbps)	Latency/RTT(ms)	Jitter(ms)
L1	118	99.1	0.051	0.001
L2	118	99.1	0.050	0.003
L3	118	99.1	0.051	0.004
L4	118	99.1	0.055	0.001

5.5.2.4 Analysis of the IPv6 Network Results

From the table 3 above, the Data transfer, Throughput and Latency (RTT) of all the four links have a slight variation. This is because they all originate from s1 and each link has the same default settings on Mininet. TCP Throughput is inversely proportional to the delay/RTT as expressed in the equation below.

$$Throughput = \frac{TCP\ Window\ Size}{RTT}$$

Where RTT is the Round-Trip Time in ms

From the equation, the TCP Window size can be increased or decreased to affect the traffic passing through the network thus vary the throughput. However, there is maximum size that cannot be exceeded. TCP is a reliable protocol making it have a high transfer rate as compared to UDP. Due to its good congestion control it performs better on networks with a lot of traffic. The table 4 shows UDP performance data which is lower than TCP throughput. UDP Throughput is constant for all the links because it is not affected by delay. There is also no reliability in UDP transmission therefore the low throughput experience.

5.5.3 Comparison between IPv4 and IPv6 Network Link Results

From the results obtained above in the Tables 1,2,3 and 4 of the networks of the two protocols (IPv4 and IPv6), it is seen that UDP results are the same. This could be because UDP transmits data in a best effort basis and is connection less oriented therefore the protocol type does not really affect the amount of data transmitted. However, the Latency for UDP transmission in IPv4 is higher than in IPv6. This means that data for the IPv4 transmission was sent using larger buffer sizes to achieve the same Throughput as IPv6 for all the links. On the other hand, TCP results for IPv4 and IPv6 differ. The TCP Throughput for IPv6 transmission is higher than IPv4. This could be so because IPv6 has a bigger header size than IPv4. Also, the IPv6 network has a lower latency than IPv4. This could be because IPv6 uses fewer hops as data is sent to the destination as compared to IPv4.

5.5.4 TCP and UDP Throughput between sampled hosts

In comparing TCP and UDP graphically, the IPv6 Network setup was used. The Figure 26 below represents the experimental data collected by using the Iperf network testing tool. Iperf creates TCP and UDP streams to measure the Throughput between two hosts in the network. The TCP and UDP throughput was compared to each other in the graph shown on Figure

26 below.

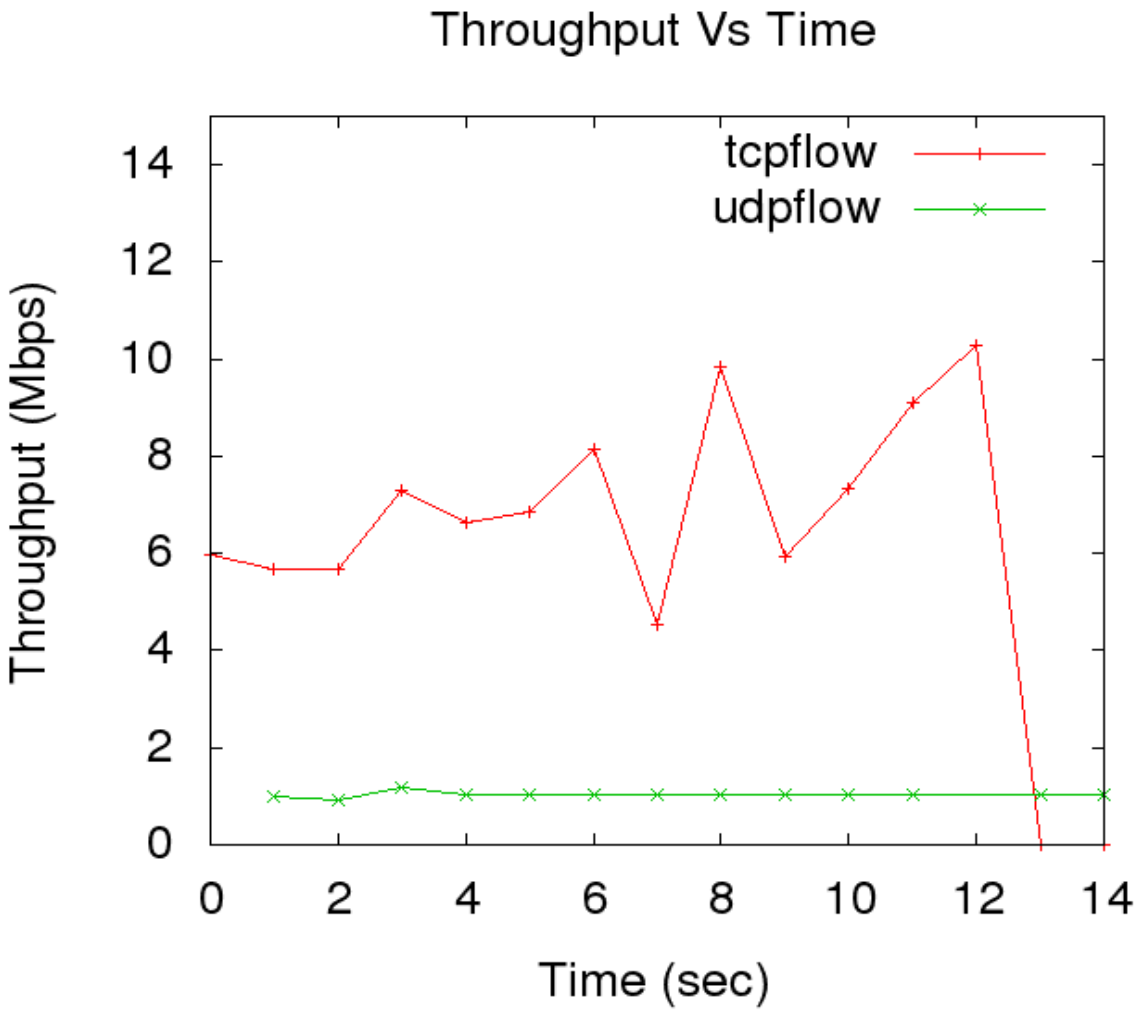

Figure 26 IPv6 Network UDP and TCP Throughput

5.5.4.1 Analysis of the Results

In the Figure 26 above TCP Throughput fluctuates between 4 to 10 Mbps from time 0 to 12 seconds then falls sharply to 0 between 12 to 13 seconds. It then maintains at 0 Mbps between 13 to 14 seconds. There is packet loss in the TCP transmission after 12 seconds elapses. The sharp fluctuation arises in TCP because it checks for errors in packets to ensure reliability. TCP uses the three-way handshake and therefore there must be acknowledgement of the received packets from the recipient end.

UDP Throughput has a slight variation between 2 to 4 seconds but mostly remains constant from 1 to 14 seconds. This is because iperf maintains a constant bit rate for UDP traffic in the network which is similar to voice traffic behavior. Also, UDP Throughput is not affected by latency in the network.

5.6 IPv6 Tunneling over IPv4

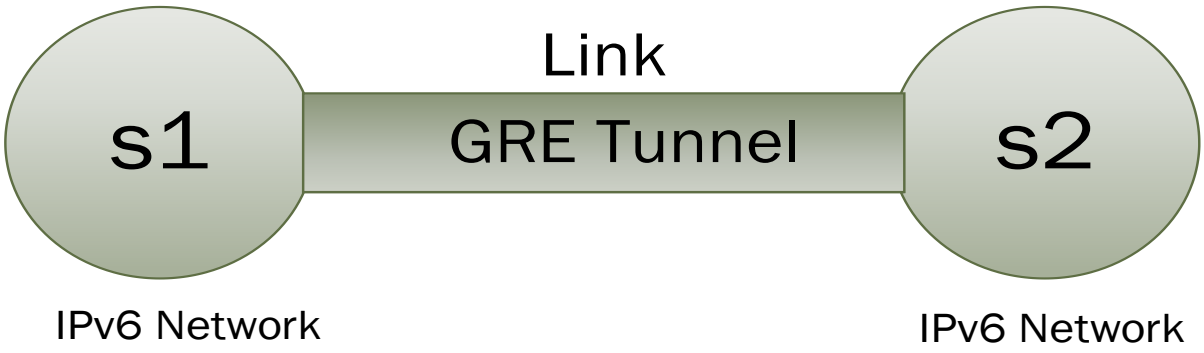

Figure 27 Tunnel Link between two IPv6 Networks

5.6.1 IO Graphs Results

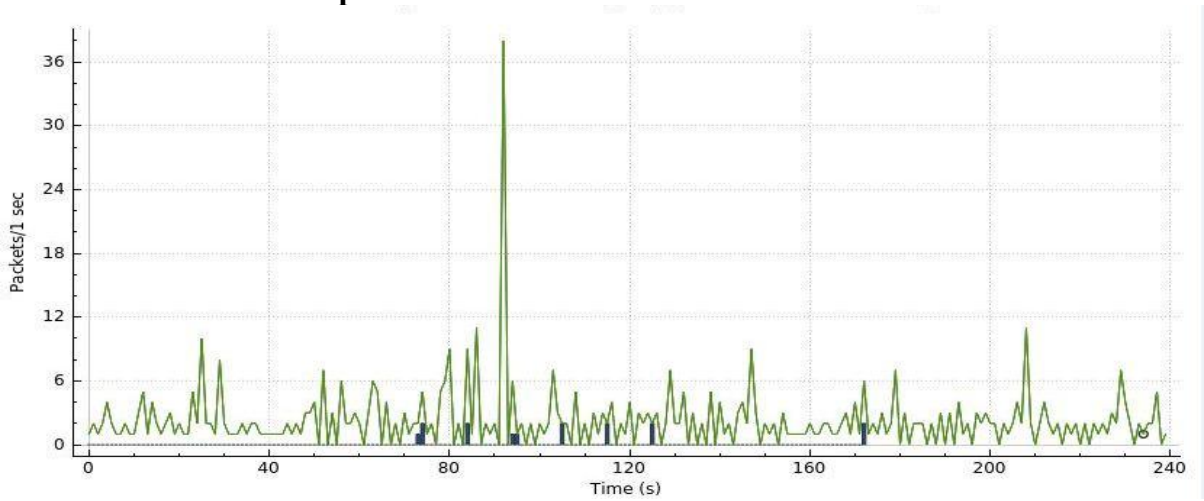

Figure 28 Traffic density between the two networks before ICMP requests

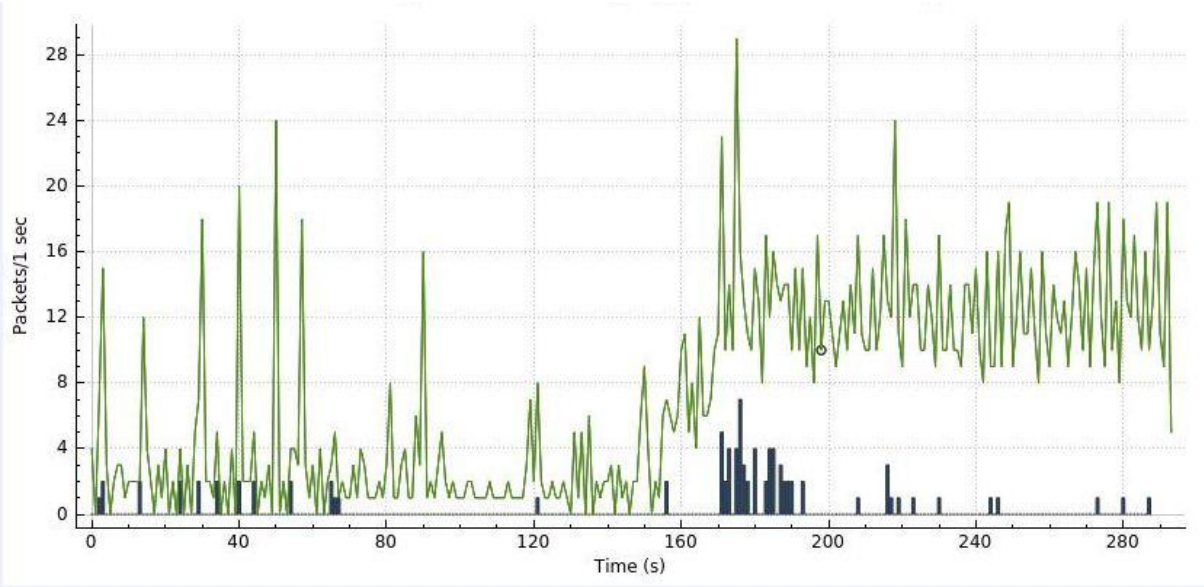

Figure 29 Increase in Traffic density due to ICMP requests across the tunnel

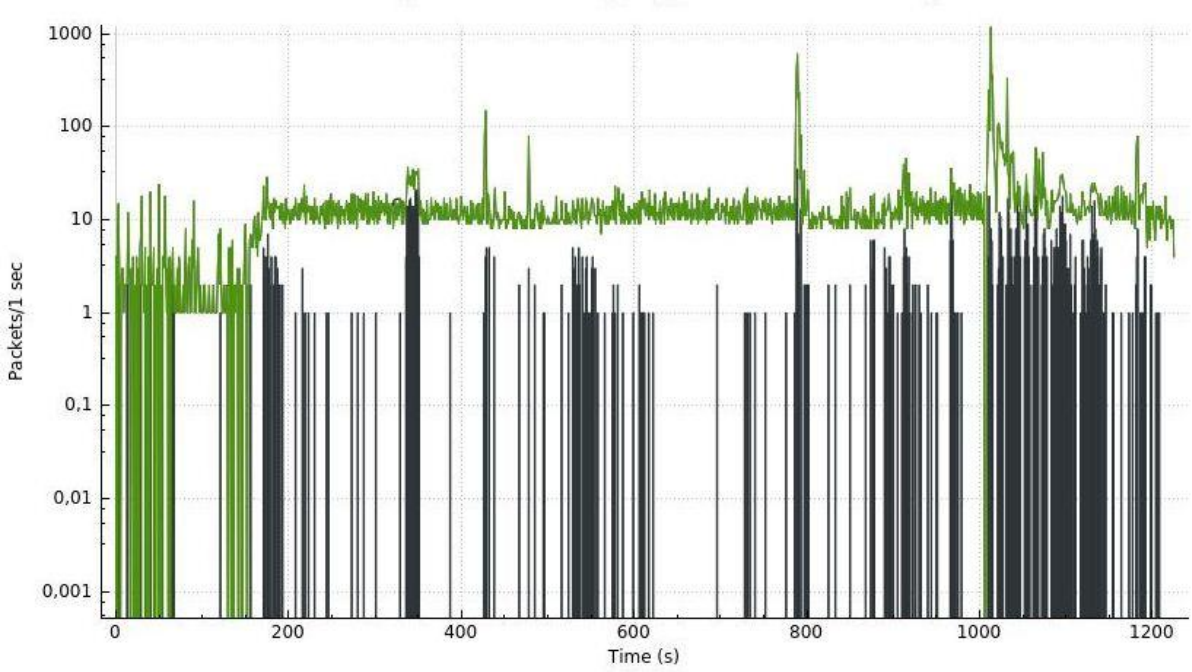

Figure 30 Traffic density viewed on the Logarithmic scale

5.6.1.1 Analysis of the Results

Figure 28 shows the traffic density across the tunnel before the ICMP requests are issued. Traffic mainly consists of TCP keep alive messages, Spanning Tree Protocol (STP) packets, OSPF hello messages, DHCPv6 solicitation messages and Address Resolution Protocol (ARP) packets. ICMP requests are made between Host 1(Network 1) and Host 3(Network 2) resulting in an increase in the traffic density in the network. From Figure 29 the traffic density increased from a maximum of 24 packets/second to 29 packets/second. Figure 30 shows the traffic density in the network represented using the logarithmic scale. The TCP errors can now be seen clearly in black. TCP errors result from retransmission and dropping of packets in the network link.

5.6.2 Flow Graphs

The Flow Graph representing the map of ICMP requests between VM 1(*137.158.131.170*) and VM 2(*137.158.131.187*) is as shown in Figure 31 below. The Flow Graph shows how traffic moves to and from the IP address *137.158.131.170* to *137.158.131.187* in the tunnel in VM Workstation Player. *255.255.255.255* is a broadcast address that sends broadcasts on the network without considering the recipient addresses.

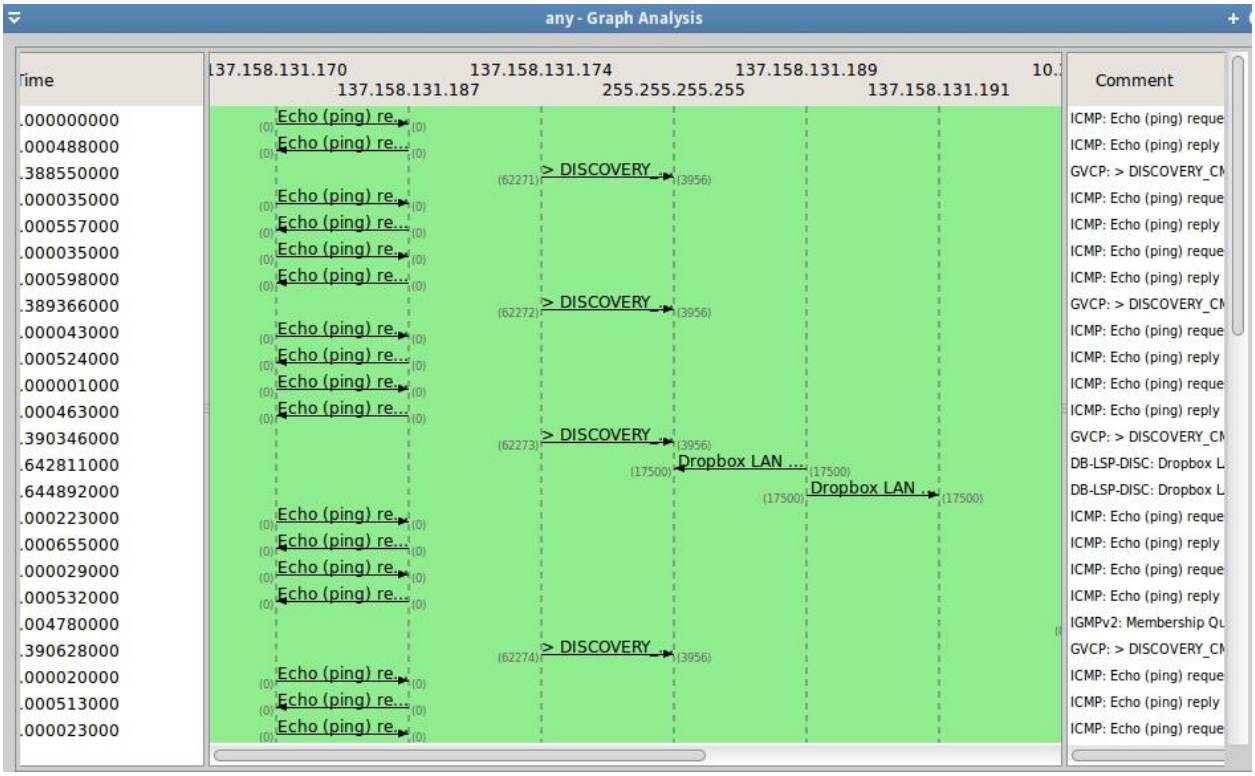

Figure 31 Flow Graph of traffic between VM1 and VM2 (IPv4 Network)

The Flow Graph for the ICMPv6 pings between Host1-VM1(*fc00::1*) and Host3-VM2(*fc00::3*) is as shown in the Figure 32 below. This confirms the end to end connectivity of the GRE tunnel created between the two VMs. The network in between is the host computers' internal network which is IPv4. The addresses *fe80::200:ff:fe00::3* and *fe80::200:ff:fe00::1* in the Flow graph 32 below are the link-local addresses for host 1 and 3 on the networks 1 and 2 respectively.

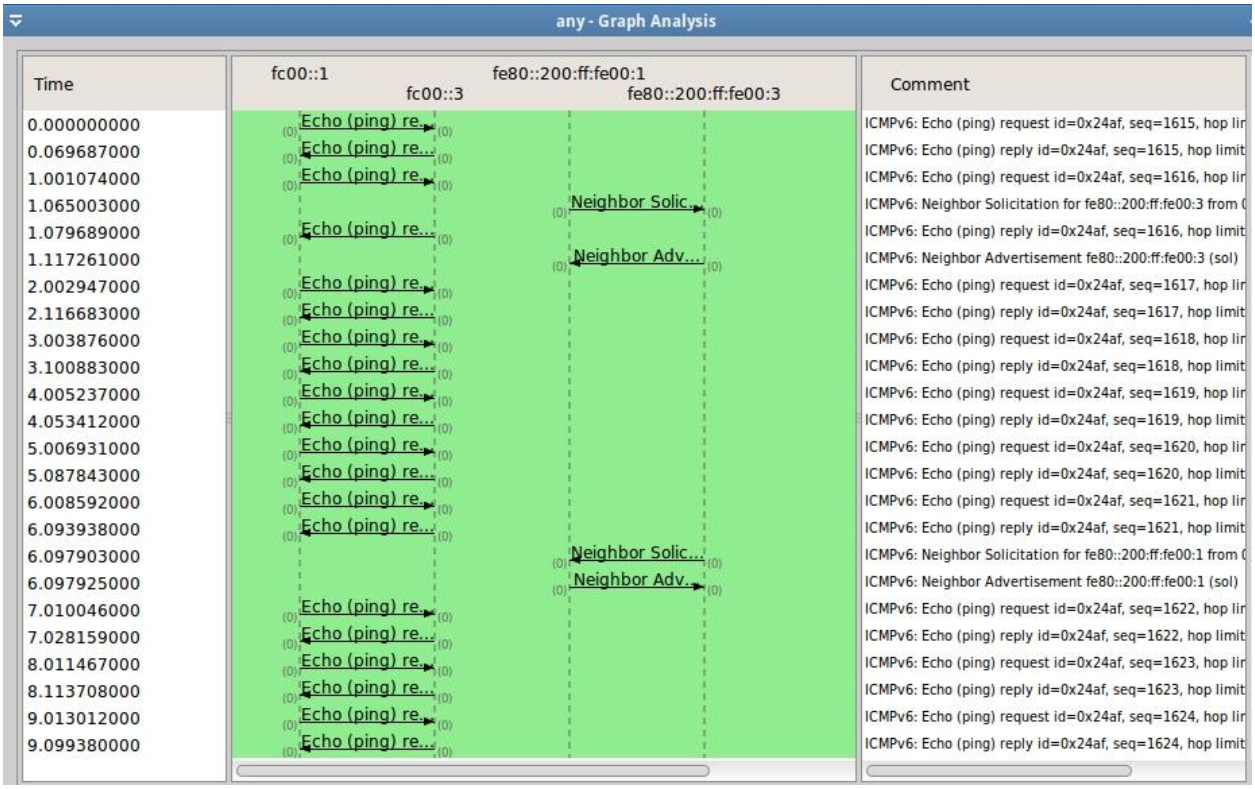

Figure 32 Flow Graph of ICMP requests between Host1-VM1 and Host3-VM2

5.6.3 Network Link Performance

TCP Throughput across the tunnel was checked using Iperf and transmission was successful only for the first second. In the remaining nine seconds there was no traffic registered therefore a low transfer and bandwidth overall as shown in the table 5 below.

Table 5 TCP Tunnel Link Performance

Link	Transfer(KB)	Throughput(Kbps)	Latency/RTT(ms)
L	66.9	54.1	0.051

The UDP throughput and data transfer across the link connecting the two networks was checked using iperf. The latency in the link was found by issuing ICMP requests from one host on Network 1 to another host on Network 2. The results obtained are as shown in the table 6 below.

Table 6 UDP Tunnel Link Performance

Link	Transfer(MB)	Throughput(Mbps)	Latency/RTT(ms)	Jitter(ms)
L	1.25	1.05	75.72	0.000

5.6.3.1 Analysis of the Results

The reason why TCP performance is poor is attributed to the Window Size being small, so traffic is there only for the first second of the transmission because of packet loss. UDP Throughput is not affected by latency and therefore a constant transfer is achieved for each second of transmission and therefore a significant value overall as shown in table 6 above. Iperf also maintains a constant bit rate for UDP.

5.6.4 TCP and UDP Throughput between sampled hosts

TCP and UDP streams were sent across the tunnel to analyze the network link between Host 1(Network 1) and Host 3(Network 2). The results were represented in a graph as shown in Figure 33 below.

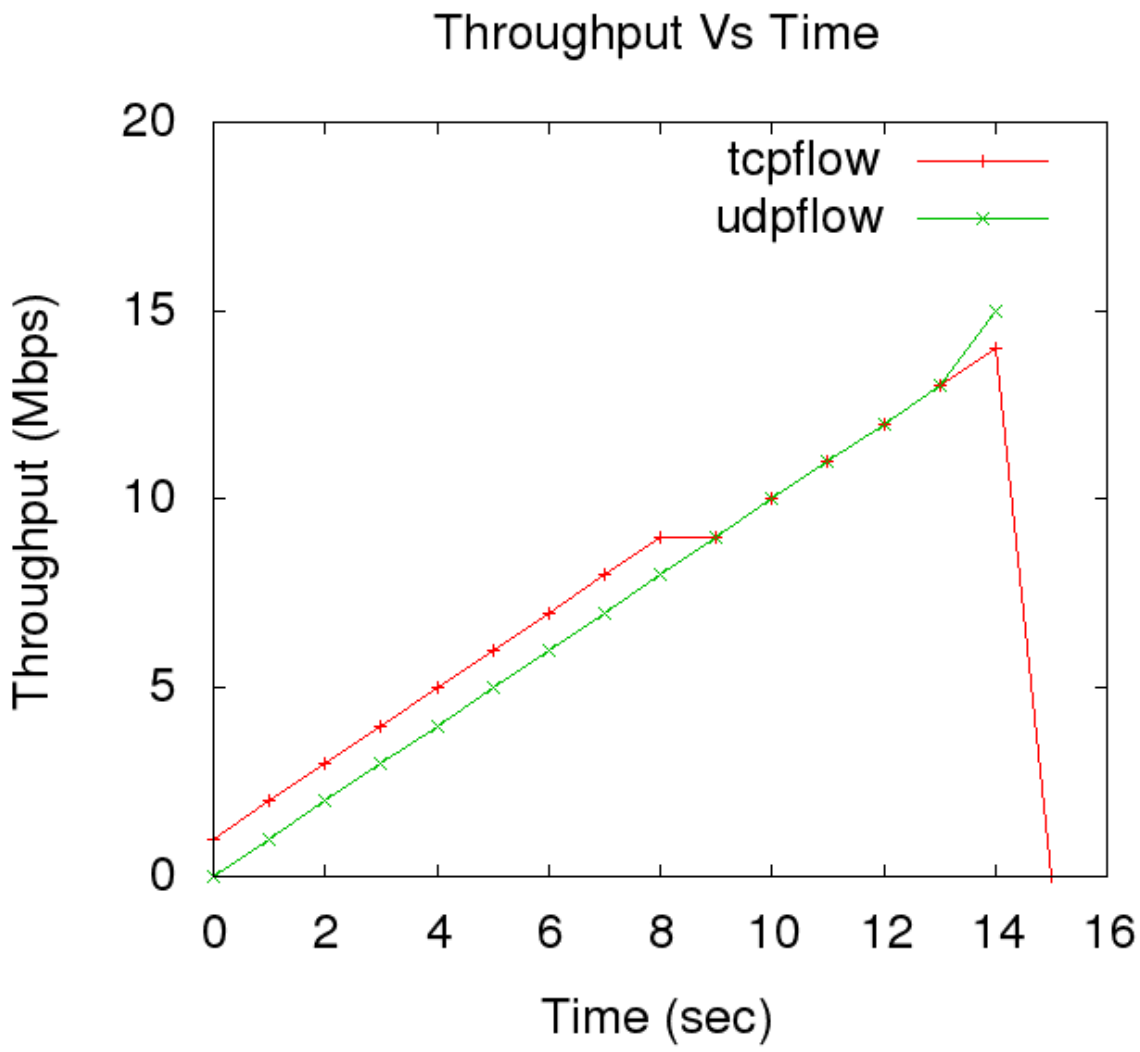

Figure 33 TCP Throughput(Mbps) vs Time(seconds)

5.6.4.1 Analysis of the Results

The Figure 33 above compares TCP and UDP throughput across the created GRE tunnel. Both TCP and UDP throughput increase with time therefore there is good connectivity. From the 14th second, UDP increases while TCP drops to zero. The reason why this happens is because the TCP stream is timed, and the last packets are dropped. TCP transmission is affected by delay. Also, the TCP Window size is small, and some packets are dropped towards the end of the transmission.

CHAPTER 6

6 CONCLUSION

6.1 Introduction

In this chapter, a summary of the content of this research work is presented in the preceding chapters. This chapter also presents the suggested future work regarding IP addressing in 5G networks especially in the transition towards IPv6 only future networks.

6.2 Discussion

The Internet Protocol is necessary for communication in all networks. IP must be utilized well in SDN-NFV based future networks for there to be seamless communication of devices. The depletion of IPv4 addresses in the last few years has necessitated the use of IPv6 addresses in networks. IPv6 protocol has enough addresses to support 5G networks. In the period before IPv6 becomes fully utilized in networks, IPv4 will be used concurrently with IPv6. IPv4 is not compatible with IPv6 therefore, there must be a way of ensuring that both are used in networks effectively without conflicting. There are three transition technologies that have been proposed to facilitate transition from IPv4 to IPv6 in future networks. These are Dual Stack, Tunneling and Translation. This research work discusses these IP transition technologies and puts a focus on tunneling.

IPv4 and IPv6 have been compared in this dissertation and how each is suited in providing addressing needs for future networks is discussed. Their datagrams are structured differently. IPv6 has additional features that make it unique and more robust. The key technologies that are used to build 5G networks are discussed in this dissertation. SDN brings easy programmability to networks by separating the Control plane from the Data plane of the network. NFV on the other hand, ensures virtualization of network functions to ensure agility and flexibility. The proposed IP transition technologies have been discussed in detail in this dissertation. They have been compared against each other and their pros and cons evaluated. The identity of devices in networks is given by IP. When this identity is altered by an attacker it can lead to packet loss in the network. Security threats related to IP have been discussed in this research work and their counter measures are given. Securing networks using IP involves filtering IP using ACLs and Firewalls. MTD strategies also are used to secure networks by dynamically altering topologies as discussed in this dissertation.

In this dissertation, the IPv4 and IPv6 networks have been modelled and their performance compared. The results obtained indicate that the TCP and UDP Throughput performance for both protocols is more or less the same. However, UDP Throughput unlike TCP has the same results

for both IPv4 and IPv6 protocols because it is not affected by delay. Tunneling is the IP transition technology demonstrated in this research work. IPv6 is run over an IPv4 network using a GRE Tunnel. The Tunnel connects two IPv6 networks over an IPv4 network. The connectivity of the tunnel is checked by issuing ICMP requests between hosts on either side and is successful. Also, TCP and UDP data streams are sent across the tunnel to check its Throughput. The tunnel is efficient in allowing UDP traffic across but for TCP there are some packets that are dropped because of a small window size.

Iperf network testing tool provides a constant bit rate for UDP traffic in the network. UDP is not affected by delay and jitter in the network. However, TCP transmission is different because it requires acknowledgement and is connection oriented. Therefore, from the results obtained in Chapter 5, UDP transmission performs better for both IPv4 and IPv6 networks and for the GRE Tunnel link created to simulate IP Tunneling.

6.3 Future Work

6.3.1 VXLAN Tunneling

In this research work, only GRE tunnelling was used to send IPv6 traffic across the IPv4 network. As part of future work VXLAN tunnels can also be used as a method to send IPv6 traffic over IPv4 in 5G networks. Its performance can then be compared with the GRE tunnel and the one that is better should be preferred.

6.3.2 Virtual Tenant Network (VTN)

Due to time constraints, the IP addressing model for VTNs explained in section 4.4 has been added to future work. This model separates VTNs using IP addresses from different subnets therefore ensuring confidentiality of information amongst tenants in the network by creating different logical network zones. This will enhance efficient business operations amongst Virtual Network Operators (VNOs).

6.3.3 Translation

Translation was not evaluated in this research work because of the complexities associated with converting IPv4 addresses to IPv6 and vice versa. IPv4 addresses are in decimal format whereas IPv6 addresses are in hexadecimal and therefore translating between the two in Mininet requires a special program. This translation can also slow down the network. Therefore, finding the efficient way to translate in 5G networks between the two protocols forms part of the future work.

6.3.4 IP Security

IP security strategies in this dissertation were proposed but their models were not

implemented. These form part of future work as some security approaches especially the ones related to IPv6 addressing are still work in progress. This is because IPv4 header is structured differently from IPv6 header therefore not all security solutions that worked for the former will work for the latter.

6.4 Chapter Summary

This Chapter gives an overview of what has been covered in this research work. This includes the proposed IP addressing strategy for 5G networks and IP transition technologies proposed with a focus on IP tunneling. Future work that can be done to improve the use of IP addresses in SDN-NFV based networks has also been suggested.

REFERENCES

[1] 5G Initiative Team and NGMN Alliance, "5G White Paper," *By NGMN Alliance 1.0*, p. 124, 2015. [Online]. Available: https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2015/NGMN_5G_White_Paper_V1_0.pdf. [Accessed: 10- Oct-2017].

[2] J. Ha and N. Park, "Unified access control for 5G convergence network with DHCP," *2016 IEEE Int. Conf. Ubiquitous Wirel. Broadband, ICUWB 2016*, pp. 3–6, 2016.

[3] V. K. Priya, "ONOS Application Tutorial - DHCP Application - Usage Information," 2016. [Online]. Available: http://kspviswa.github.io/Using-DHCP-app-ONOS.html. [Accessed: 20-Oct-2017].

[4] ICANN, "Internet protocol (ip) addresses," *Beginner's Guide*, p. 12, 2011.

[5] M. Richart, J. Baliosian, J. Serrat, and J. L. Gorricho, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Trans. Netw. Serv. Manag.*, vol. 13, no. 3, pp. 462–476, 2016.

[6] L. Retselisitsoe, "SDN Based Security Solutions for Multi-Tenacy NFV," M.Eng. dissertation, Dept. Elect. Eng., University of Cape Town, 2016.

[7] P. Wu, Y. Cui, J. Wu, J. Liu, and C. Metz, "Transition from IPv4 to IPv6: A state-of-the-art survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1407–1424, 2013.

[8] D. Niyato, H. Xie, W. Xia, Y. Wen and C. H. Foh, "A Survey on software-defined networking," *IEEE Commun. Surv. TUTORIALS*, vol. 17, no. 1, pp. 27–50, 2015.

[9] D. Rudolf, "Next Generation Internet : IPv4 Address Exhaustion, Mitigation Strategies and Implications for the U.S," 2009. [Online]. Available: https://dokumen.tips/documents/ipv4-address-exhaustion-home-ieee-usa-address-exhaustion-shares-many-of-the.html. [Accessed: 13-Oct-2017].

[10] sdx central, "SDN articles, whitepapers, videos & more," *SDN*. [Online]. Available: https://www.sdxcentral.com/sdn/. [Accessed: 13-Oct-2017].

[11] S. Verma, "IPv4," 2016. [Online]. Available: https://techtud.in/chapter/ipv4ipv6-and-tcp. [Accessed: 08- Feb-2018].

[12] K. Govindarajan, S. Setapa, K. C. Meng, and H. Ong, "Interoperability issue between IPv4 and IPv6 in OpenFlow enabled network: IPv4 and IPv6 transaction flow traffic," *Proceeding - 2014 Int. Conf. Comput. Control. Informatics Its Appl. "New Challenges Oppor. Big Data", IC3INA 2014*, pp. 58–63, 2014.

[13] R. Bonica, "IPV6 Fragmentation The Case For Deprecation." [Online]. Available: https://www.nanog.org/sites/default/files/mon.general.fragmentation.bonica.pdf. [Accessed: 08-Nov-2017].

[14] G. Pavithra and D. T. Santosh, "IP Packet Fragmentation and Reassembly at Intermediate Routers," *International J. Comput. Sci. Int. J. Comput. Sci. Int. J. Comput. Sci. Eng. Eng. Eng.*, vol. 2, no. 4, pp. 168–171, 2014.

[15] W. Commons, “IPv6 Header,” 2014. [Online]. Available: https://commons.wikimedia.org/wiki/File:IPv6_header_rv1.png. [Accessed: 08-Feb-2018].

[16] Cisco Systems, “IPv6 Addressing,” *Cisco Syst.*, vol. 6, p. 28, 2008. [Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/IPv6_WP.pdf. [Accessed: 23-Oct-2017].

[17] N. Ravi, M. A. Saravanan, and M. Periyasamy, “Implementation of IPv6 / IPv4 Dual-Stack Transition Mechanism,” vol. 5, no. 5, pp. 6326–6332, 2014.

[18] Cisco Systems Inc., “NAT64 Technology : Connecting IPv6 and IPv4 Networks,” *Webtorials*, vol. 4, no. June, pp. 1–22, 2011.

[19] P. Yadav and R. Singhal, “Effective Tunneling of Traffic and Data in a Network with L2TP Based ON L2F,” vol. 3, no. 2, pp. 38–42, 2014.

[20] A. Burande, A. Pise, S. Desai, and Y. Martin, “Wireless Network Security by SSH Tunneling,” *Int. J. Sci. Res. Publ.*, vol. 4, no. 1, pp. 2250–3153, 2014.

[21] J. Lin, K. Wang, S. Cheng, and Y. Liu, “On Exploiting SDN to Facilitate IPv4 / IPv6 Coexistence and Transition,” pp. 473–474.

[22] “Open Networking Foundation is an operator led consortium leveraging SDN, NFV and Cloud technologies to transform operator networks and business models.” [Online]. Available: https://www.opennetworking.org/. [Accessed: 16-Jan-2018].

[23] W. Braun and M. Menth, “Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices,” *Futur. Internet*, vol. 6, no. 2, pp. 302–336, 2014.

[24] ETSI, “NFV ETSI 2,” *Terminal. Main Concepts NFV*, no. 1, pp. 1–16, 2013. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf. [Accessed: 08-Nov-2017].

[25] HP, “Technical White Paper- Network Functions Virtualization,” 2014. [Online]. Available: http://www.hp.com/hpinfo/newsroom/press_kits/2014/MWC/White_Paper_NFV.pdf. [Accessed: 08-Nov-2017].

[26] L. F. Chiang and J. W. Dai, “A new method to detect abnormal ip address on DHCP,” *J. Networks*, vol. 4, no. 6, pp. 458–464, 2009.

[27] “DHCP Technology White Paper,” pp. 1–20, 2008. [Online]. Available: http://tiszai.tricon.hu/PDF/DHCP%20Technology%20White%20Paper.pdf. [Accessed: 16-Nov-2017].

[28] M. Khadilkar and N. Feamster, “Usage-based DHCP lease time optimization,” *Proc. 7th ...*, pp. 71–76, 2007.

[29] N. S. Limited, “Virtualized Domain Name System and IP Addressing Environments White Paper Virtualized DNS and IP Addressing Environments,” no. September, 2010. [Online]. Available:

http://www.circleid.com/pdf/White_Paper_Virtualized_DNS_and_IP_Addressing_Environments.pdf. [Accessed: 08-Nov-2017].

[30] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, 2015.

[31] D. Evans, "The Internet of Things - How the Next Evolution of the Internet is Changing Everything," *CISCO white Paper*, no. April, pp. 1–11, 2011. [Online]. Available: http://www.circleid.com/pdf/White_Paper_Virtualized_DNS_and_IP_Addressing_Environments.pdf. [Accessed: 24-Oct-2018].

[32] G. Corser, G. Fink, M. Aledhari, and J. Bielby, "IEEE Internet Technology Policy Community White Paper Internet of Things (IoT) Security Best Practices," 2017. [Online]. Available: https://internetinitiative.ieee.orvg/images/files/resources/white_papers/internet_of_things_may_2017.pdf. [Accessed: 03-Feb-2018].

[33] E. B. Rouse and Sharon Shea, "Smart City- IoT Agenda,Tech Target," 2017. [Online]. Available: https://internetofthingsagenda.techtarget.com/definition/smart-city. [Accessed: 03-Feb-2018].

[34] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[35] M. Weyrich, J. P. Schmidt, and C. Ebert, "Machine-to-machine communication," *IEEE Softw.*, vol. 31, no. 4, pp. 19–23, 2014.

[36] F. Yan, Y. Jian-Wen, and C. Lin, "Computer Network Security and Technology Research," in *2015 Seventh International Conference on Measuring Technology and Mechatronics Automation*, 2015, vol. 1, pp. 293–296.

[37] P. K. Kulkarni, R. Shelk , K. Gaikwad, V. Solanke and S. Gujar, Wireless Sensor Network Security Threats," *Computer (Long. Beach. Calif).*, vol. 5, no. 10, pp. 54–62, 2002.

[38] A. Gupta and R. K. Jha, "Security threats of wireless networks: A survey," in *International Conference on Computing, Communication & Automation*, 2015, pp. 389–395.

[39] L. Ertaul, K. Venkatachalam, and N. Star, "Security of Software Defined Networks (SDN)," pp. 24–30.

[40] Oracle, "Oracle SDN Virtual Network Services Overview," 2016. [Online]. Available: http://www.oracle.com/us/products/networking/virtual-networking/sdn/vns-wp-3229862.pdf. [Accessed: 12-Feb-2018]

[41] M. Kühne, "Rogue DHCPv6 Server," *RIPE NCC*, 2013. [Online]. Available: https://labs.ripe.net/Members/johannes_weber/RogueDHCPv6Server.jpg/image_view. [Accessed: 12-Feb-2018].

[42] OpenDaylight Projects, “Home - OpenDaylight,” 2018. [Online]. Available: https://www.opendaylight.org/ [Accessed: 20-Feb-2018].

[43] C. M. Chan, “IPv6 in Mininet _ mcchan.” [Online]. Available: http://blog.mcchan.io/ipv6-in-mininet. [Accessed: 26-Feb-2018].

[44] G. Gee, “Connecting two Mininet networks with GRE tunnel – Part 2 _ Tech and Trains.” [Online]. Available: https://techandtrains.com/2014/01/20/connecting-twmininet-networks-with-gre-tunnel-part-2/ [Accessed: 02-Mar-2018].

[45] Linux Foundation, “Connecting VMs Using Tunnels — Open vSwitch 2.” [Online]. Available: http://docs.openvswitch.org/en/latest/howto/tunneling/ [Accessed: 02-Mar-2018].

[46] GNS3 Team, “Marketplace - Appliances - GNS3.” [Online]. Available: https://www.gns3.com/marketplace/appliances. [Accessed: 12-Feb-2018].

[47] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, “Generic Routing Encapsulation (GRE),” pp. 127–148, 2000. [Online]. Available: https://tools.ietf.org/html/rfc2784 [Accessed: 03-Mar-2018].

[48] TechTarget, “DHCP (Dynamic Host Configuration Protocol)” [Online]. Available: https://searchnetworking.techtarget.com/definition/DHCP. [Accessed: 03-Nov-2017]

7 Appendix A: Experimentation

A.1 Set-up and configuration

This section describes the actual tools that were used to realize the IPv6 network evaluated in this research work. It also describes the components used to implement IPv6 tunneling over IPv4. This is one of the technologies chosen for transition towards IPv6 only future networks. This section guides the reader through the configuration of the IPv6 only network and IPv6 tunneling over the IPv4 network.

A.2 System Requirements

The Mininet Virtual Machine was hosted in a computer with the following system properties; 4 CPU cores,8GB RAM, 400GB Hard Drive and Ubuntu 17.04 Operating System. For tunneling the two Virtual Machines were hosted in a computer with the same system specifications and the network adapter used in Virtual Box was the Bridged one.

A.3 GNS3 Setup

The network was built in a GNS3 emulator and connected to the Internet. The Mininet VM is connected to the Ubuntu Docker container, NAT Cloud and Firefox Appliance via a legacy switch.

The NAT cloud performs the DHCP function and issues dynamic IP addresses to the network hosts. Connectivity to the Internet is first tested from the network hosts. The results from pings to 'google.com' from hosts in the above network are as shown below.

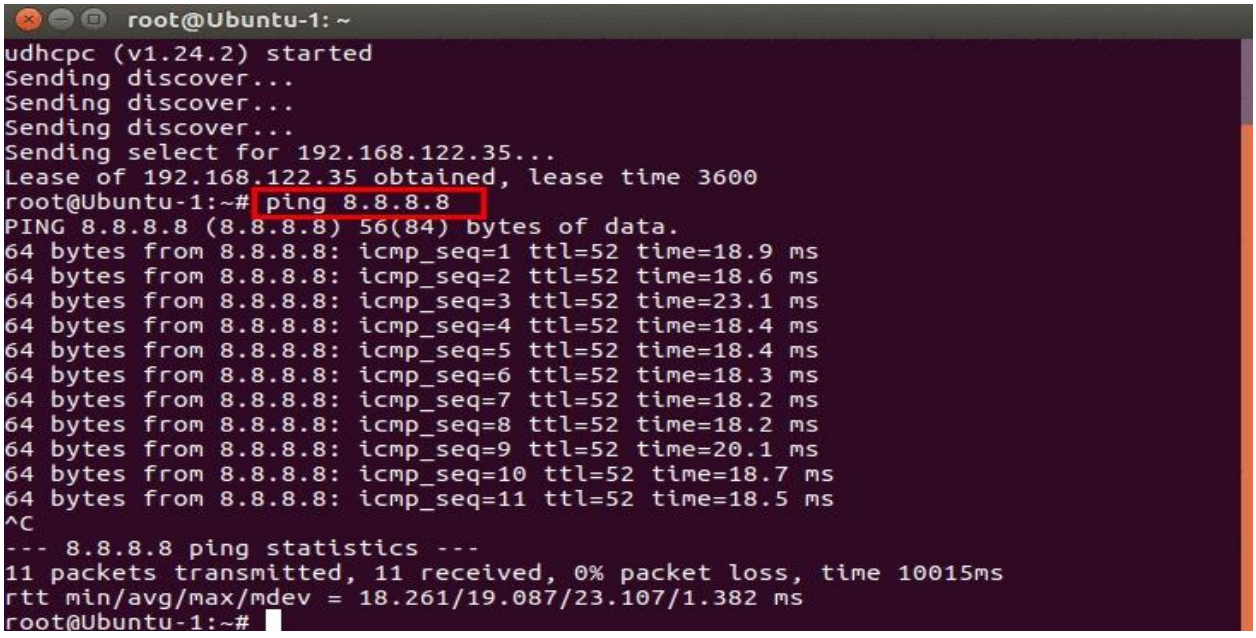

Figure 34 Result from pinging Google.com from the Ubuntu Docker Container.

The Mininet VM connected to the Internet. This was confirmed by pinging google.com and it was successful as shown in the Figure 35 shown below.

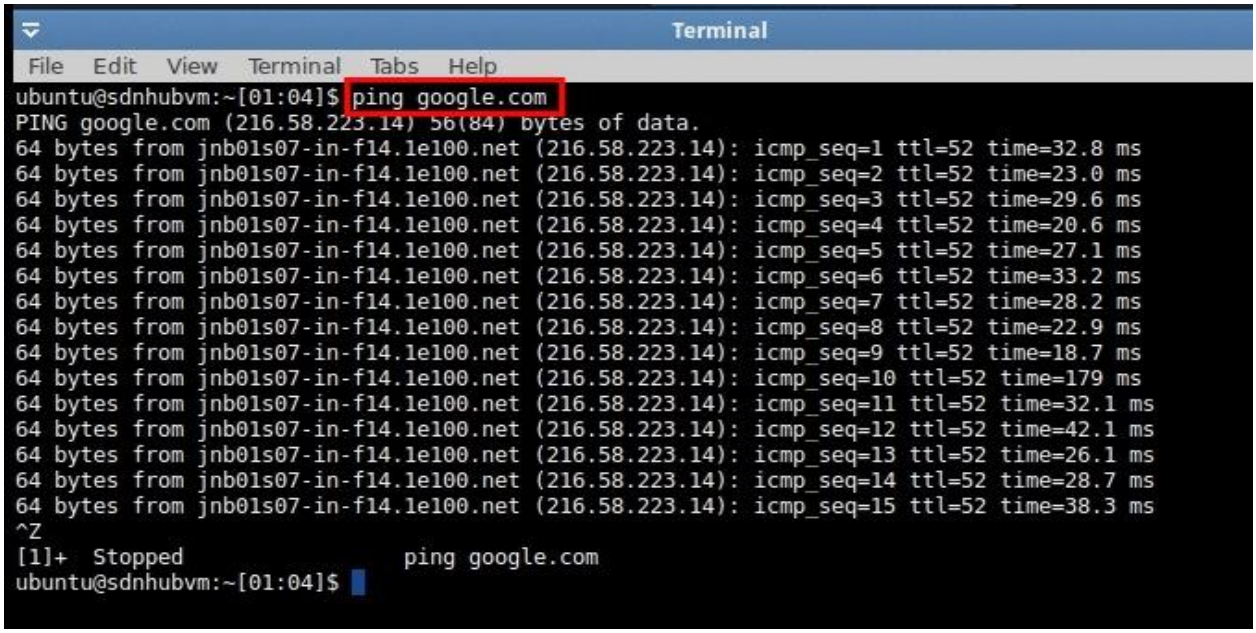

Figure 35 Result from pinging Google.com from the Mininet VM

A.4 IPv6 Only Network Setup

A python script was used to create a custom topology with 5 switches and 16 hosts and run on Mininet. OpenDaylight is installed on the Ubuntu Docker Container. The topology was pointed to the Ubuntu Docker Container which was a remote controller with IP address 192.168.122.52. The reachability of all the hosts was found using the 'pingall' command as shown in Figure 36 below. The network topology was viewed on OpenDaylight as shown on Figure 37 below.

Figure 36 Reachability of all the Hosts in the Network

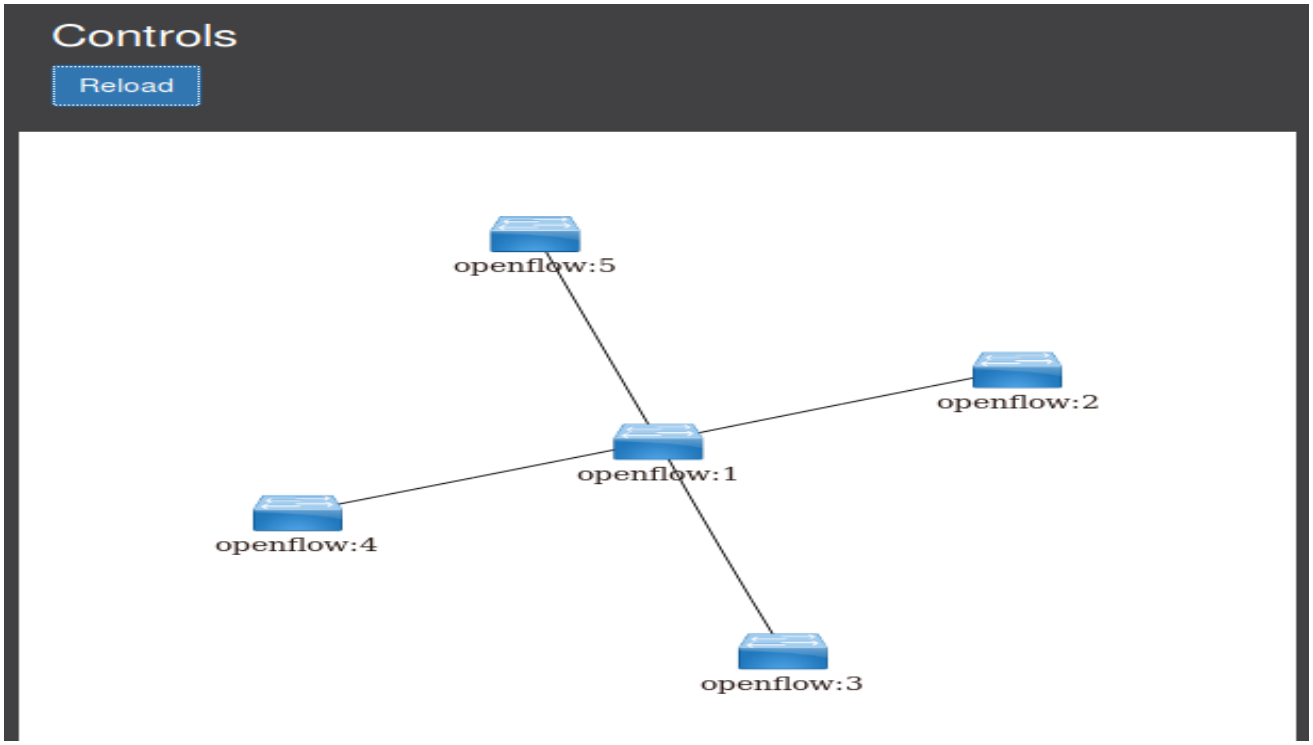

Figure 37 Network Topology as viewed on OpenDaylight

A.5 IPv6 Traffic Analysis

The IP traffic in the network was monitored. Wireshark was started to analyze and capture the packet flow in the network. The packets from the ICMP requests between Host 7 and 15 were captured as shown in the Figure 38 below.

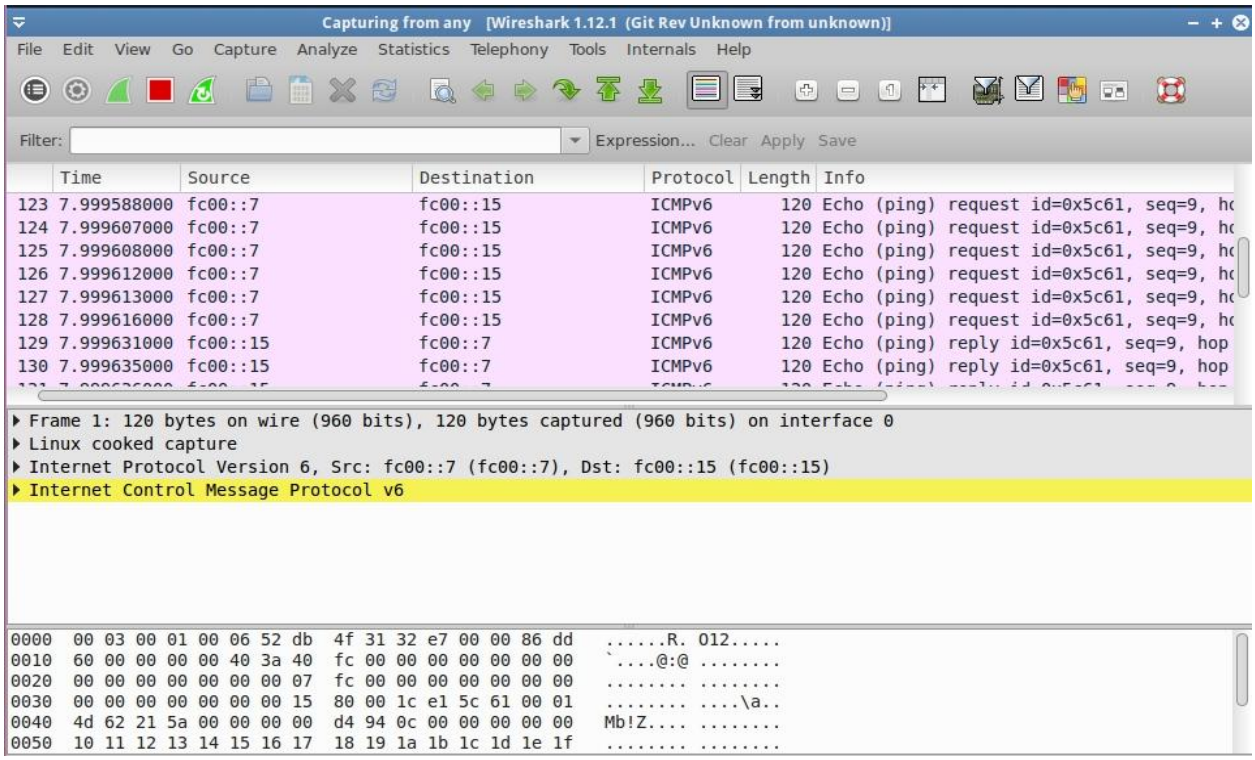

Figure 38 Packet capture of Pings between Host 7 and 15

A.5.1. IPv6 Throughput

The TCP traffic between sampled Host 7 and 15 was analyzed as shown below. The command '*iperf3 -s -i 1 -p 5566'* was used on Host 7(Server) to make it listen to Host 15(Client) as shown in Figure 39 below. The command *'iperf3 -c fc00::7 -t 15 -p 5566'* was used on Host 15(Client) to send a data stream to Host 7(Server) as shown in the Figure 40 below.

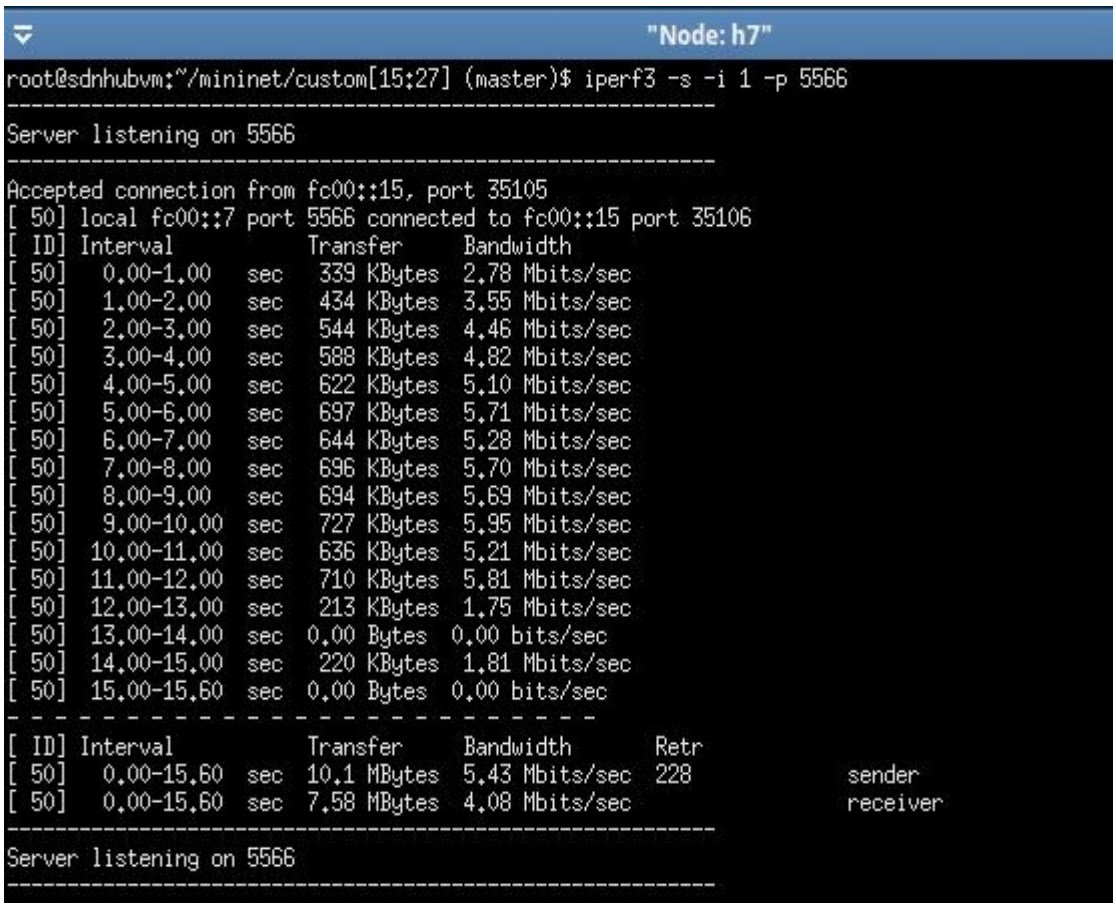

Figure 39 Host 7 (Server)) TCP connection to (Client)

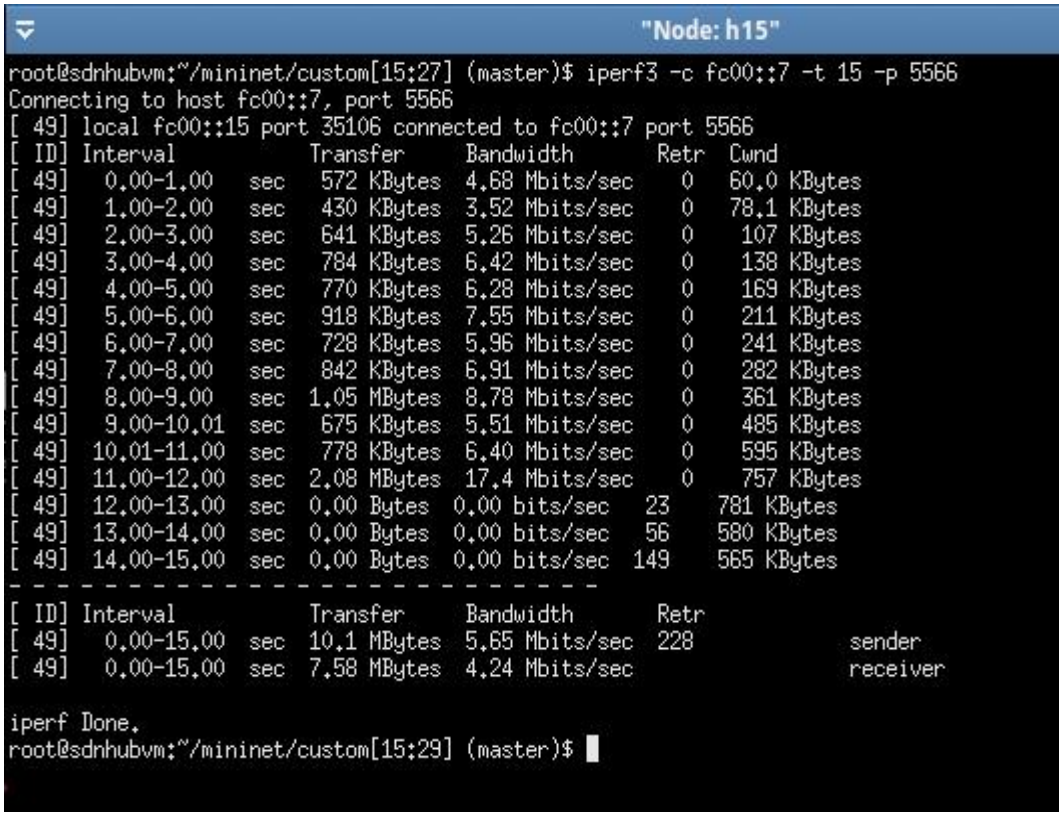

Figure 40 Host 15(Client) TCP Connection to Host 7(Server)

A.6 IPv6 Tunneling Over IPv4

Two IPv6 networks each with a switch and two hosts were created separately on two Mininet VMs and connected over the computers' IPv4 network via a GRE tunnel. The first network on VM1 had host 1(h1) and host 2(h2) linked to Switch 1(s1) whereas the second network on VM2 had host 3(h3) and host 4(h4) linked to Switch 2(s2). This is as shown in the Figures 41 and 42.

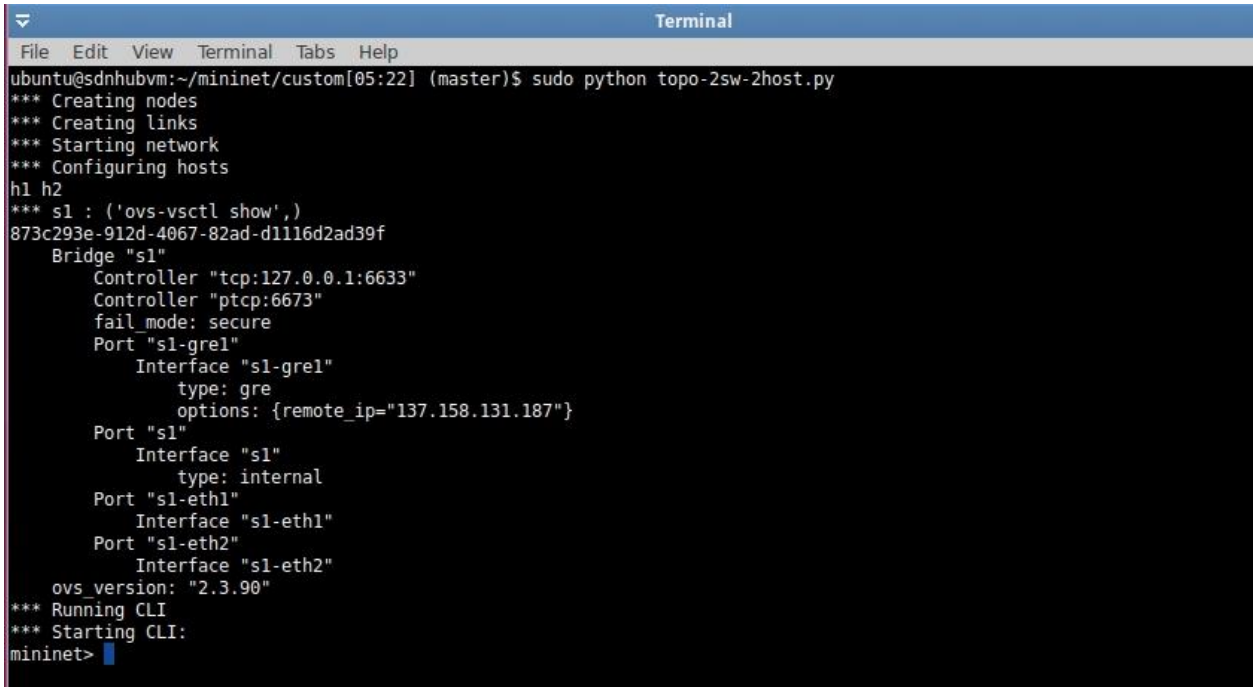

Figure 41 Network 1 setup on VM1 connected to Network 2 via a GRE Tunnel

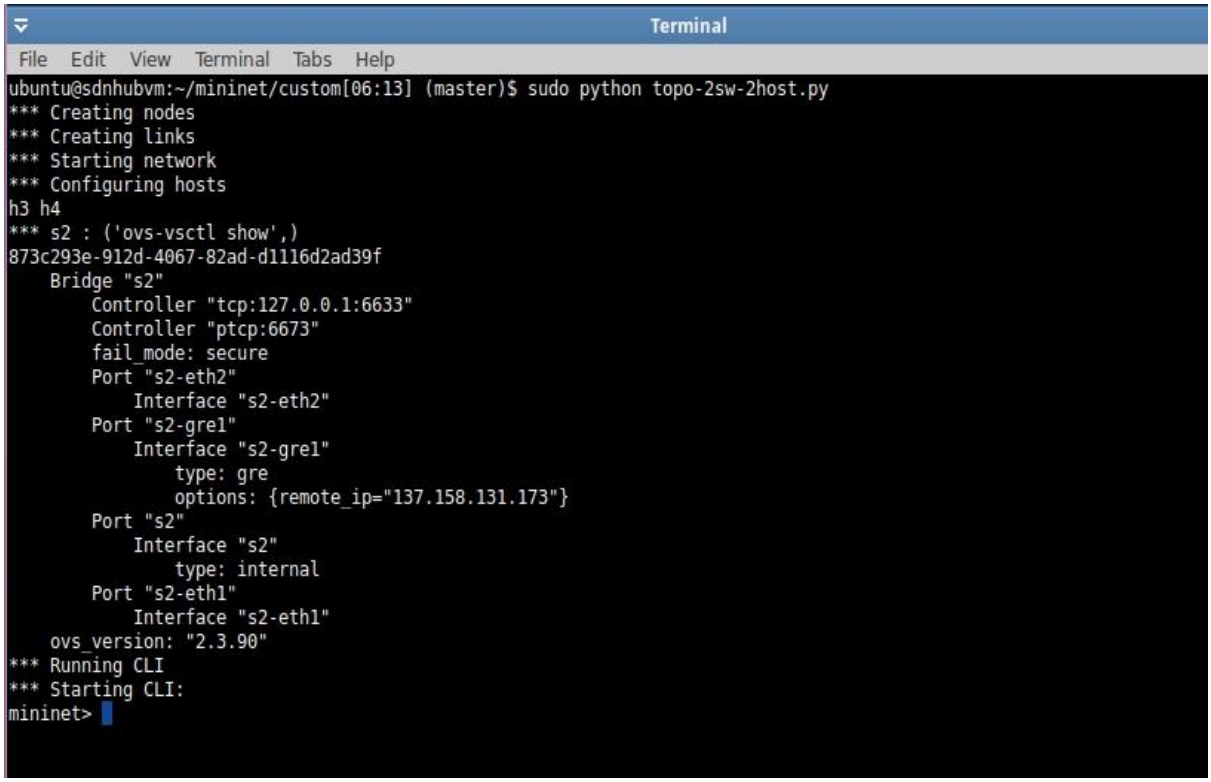

Figure 42 Network 2 setup on VM2 connected to Network 1 via a GRE Tunnel

The effectiveness of the tunnel was tested by issuing ICMPv6 requests from hosts in one network to hosts in the other network. The Figures 43 and 44 below show successful pings from h1 network 1 on VM1 to h3 and h4 on Network 2 on VM2.

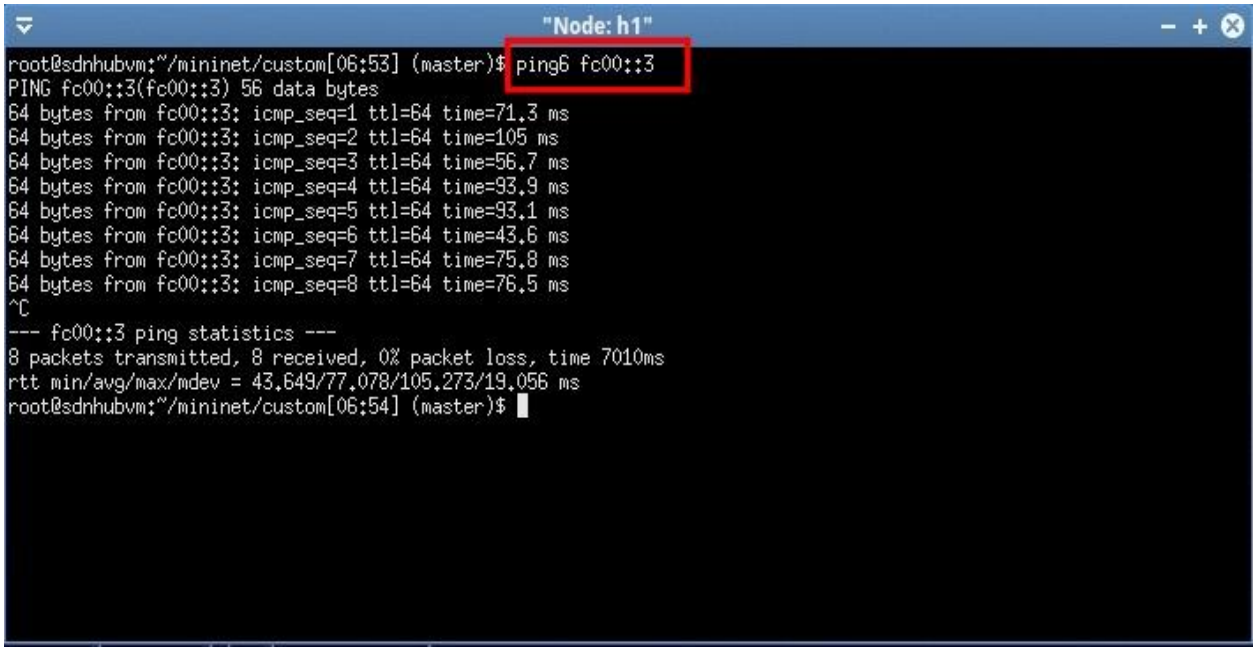

Figure 43 Successful pings of Host 3 from Host 1

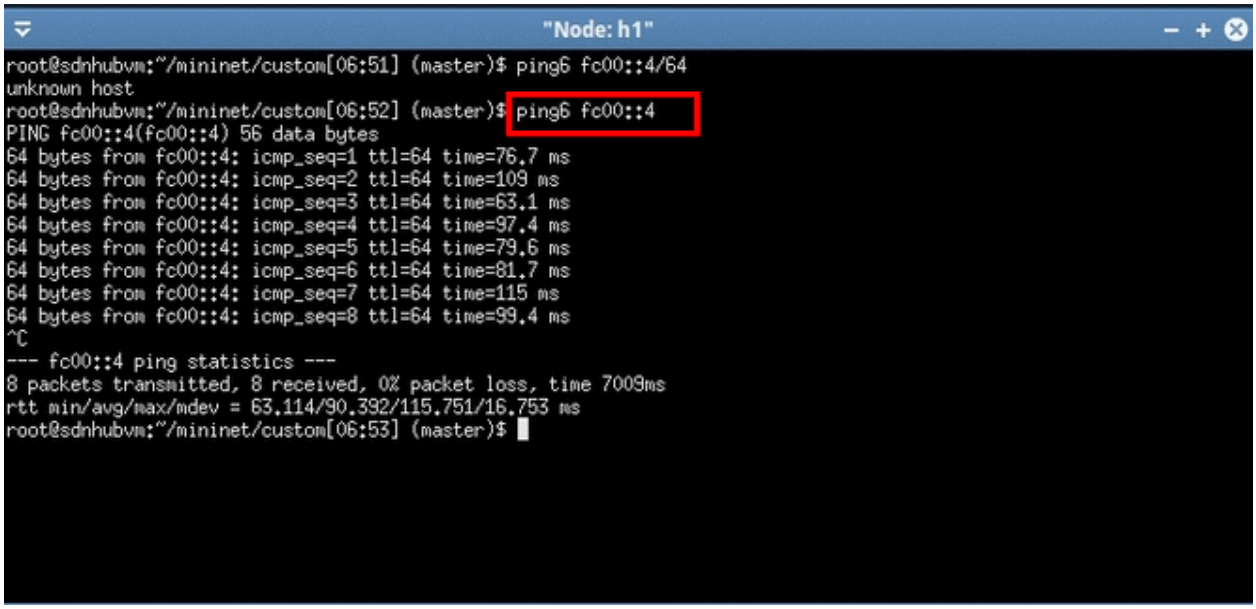

Figure 44 Successful pings of Host 4 from Host 1

Similarly, the GRE Tunnel was tested by pinging hosts in Network 1 from hosts in Network 2. The Figure 45 below shows successful ICMPv6 requests from h3 to h1.

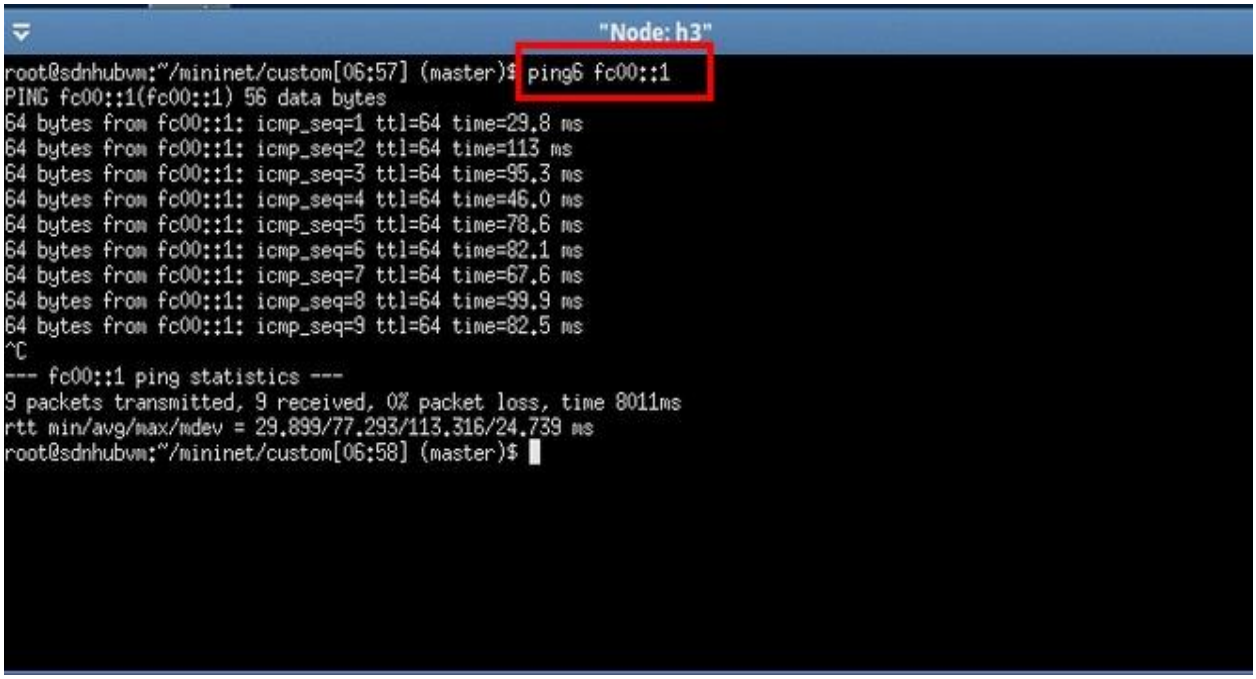

Figure 45 Successful pings of Host 1 on VM1 from Host 3 on VM2

A.6.1. TCP and UDP Throughput

The command *'iperf -s -p 8042'* was issued on Host 3-VM2(Server) to make it listen to incoming connections on TCP port 8042 whereas the command *'iperf3 -s -V'* was issued on Host 3-VM2 to make it listen to incoming connections on UDP port 5201 as shown in Figures 46 and 47 below.

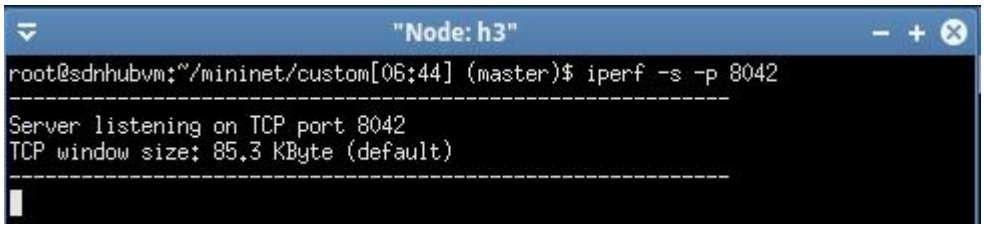

Figure 46 Host 3(Server) listening to Host 1 on TCP port 8042

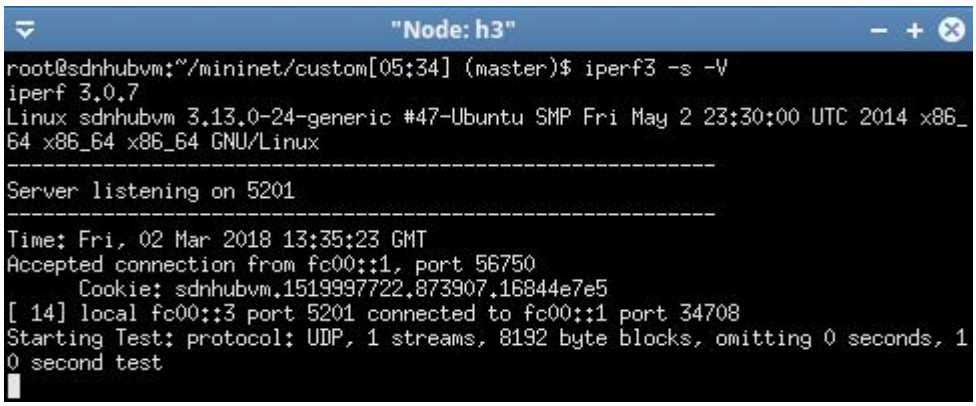

Figure 47 Host 3(Server) listening to Host 1(Client) on UDP port 5201

TCP and UDP streams were then sent to Host 1-VM1(Client) on port 5566 and 5201 respectively as shown in Figures 48 and 49 below.

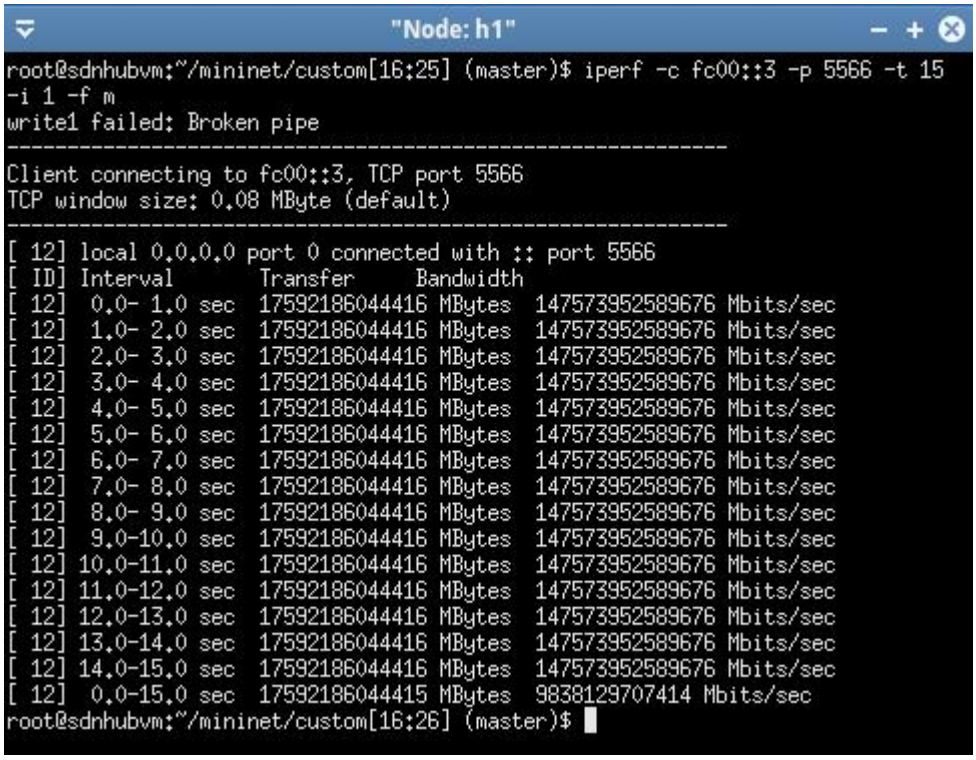

Figure 48 TCP Throughput between h1 and h3

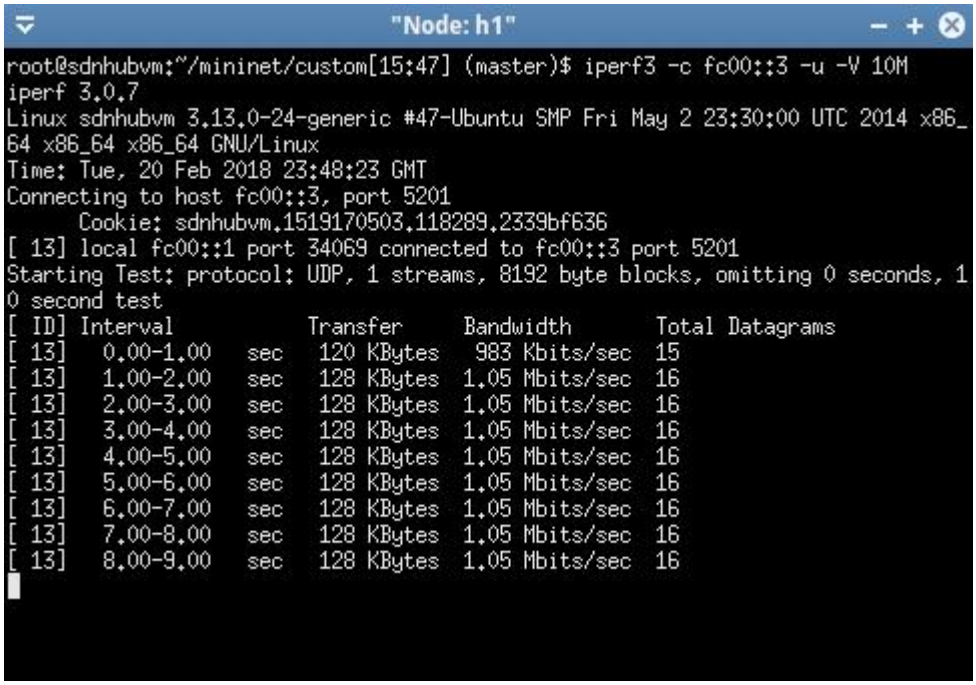

Figure 49 UDP Throughput between h1 and h3

8 Appendix B: Python Scripts

B.1 IPv6 Network Topology Code

The following code was used to create a custom tree network topology on mininet with 5 OpenFlow Switches and 16 IPv6 hosts.

```
#!/usr/bin/python
from mininet.net import Mininet
from mininet.node import Controller, RemoteController,
OVSKernelSwitch, UserSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.link import Link, TCLink,Intf

def topology():
 "Create a network."
net = Mininet( controller=RemoteController, link=TCLink,
switch=OVSKernelSwitch )

print ("*** Creating nodes")

h1 = net.addHost( 'h1', mac='00:00:00:00:00:01' )
h2 = net.addHost( 'h2', mac='00:00:00:00:00:02' )
h3 = net.addHost( 'h3', mac='00:00:00:00:00:03' )
h4 = net.addHost( 'h4', mac='00:00:00:00:00:04' )
h5 = net.addHost( 'h5', mac='00:00:00:00:00:05' )
h6 = net.addHost( 'h6', mac='00:00:00:00:00:06' )
h7 = net.addHost( 'h7', mac='00:00:00:00:00:07' )
h8 = net.addHost( 'h8', mac='00:00:00:00:00:08' )
h9 = net.addHost( 'h9', mac='00:00:00:00:00:09' )
h10 = net.addHost( 'h10', mac='00:00:00:00:00:10' )
h11 = net.addHost( 'h11', mac='00:00:00:00:00:11' )
h12 = net.addHost( 'h12', mac='00:00:00:00:00:12' )
h13 = net.addHost( 'h13', mac='00:00:00:00:00:13' )
h14 = net.addHost( 'h14', mac='00:00:00:00:00:14' )
h15 = net.addHost( 'h15', mac='00:00:00:00:00:15' )
h16 = net.addHost( 'h16', mac='00:00:00:00:00:16' )

s1 = net.addSwitch( 's1', listenPort=6673, mac='00:00:00:00:00:17' )
s2 = net.addSwitch( 's2', listenPort=6673, mac='00:00:00:00:00:18' )
s3 = net.addSwitch( 's3', listenPort=6673, mac='00:00:00:00:00:19' )
s4 = net.addSwitch( 's4', listenPort=6673, mac='00:00:00:00:00:20' )
s5 = net.addSwitch( 's5', listenPort=6673, mac='00:00:00:00:00:21' )

c0 = net.addController( 'c0', controller=RemoteController,
ip='127.0.0.1', port=6633 )

print ("*** Creating links")

net.addLink(s1, s2)
net.addLink(s1, s3)
net.addLink(s1, s4)
net.addLink(s1, s5)
```

```
net.addLink(s2, h1)
net.addLink(s2, h2)
net.addLink(s2, h3)
net.addLink(s2, h4)
net.addLink(s3, h5)
net.addLink(s3, h6)
net.addLink(s3, h7)
net.addLink(s3, h8)
net.addLink(s4, h9)
net.addLink(s4, h10)
net.addLink(s4, h11)
net.addLink(s4, h12)
net.addLink(s5, h13)
net.addLink(s5, h14)
net.addLink(s5, h15)
net.addLink(s5, h16)

print ("*** Starting network")

net.build()
c0.start()

s1.start( [c0] )
s2.start( [c0] )
s3.start( [c0] )
s4.start( [c0] )
s5.start( [c0] )

h1.cmd("ifconfig h1-eth0 inet6 add fc00::1/64")
h2.cmd("ifconfig h2-eth0 inet6 add fc00::2/64")
h3.cmd("ifconfig h3-eth0 inet6 add fc00::3/64")
h4.cmd("ifconfig h4-eth0 inet6 add fc00::4/64")
h5.cmd("ifconfig h5-eth0 inet6 add fc00::5/64")
h6.cmd("ifconfig h6-eth0 inet6 add fc00::6/64")
h7.cmd("ifconfig h7-eth0 inet6 add fc00::7/64")
h8.cmd("ifconfig h8-eth0 inet6 add fc00::8/64")
h9.cmd("ifconfig h9-eth0 inet6 add fc00::9/64")
h10.cmd("ifconfig h10-eth0 inet6 add fc00::10/64")
h11.cmd("ifconfig h11-eth0 inet6 add fc00::11/64")
h12.cmd("ifconfig h12-eth0 inet6 add fc00::12/64")
h13.cmd("ifconfig h13-eth0 inet6 add fc00::13/64")
h14.cmd("ifconfig h14-eth0 inet6 add fc00::14/64")
h15.cmd("ifconfig h15-eth0 inet6 add fc00::15/64")
h16.cmd("ifconfig h16-eth0 inet6 add fc00::16/64")


print ("*** Running CLI")
CLI( net )

print ("*** Stopping network")
net.stop()


if __name__ == '__main__':
    setLogLevel( 'info' )
    topology()
```

B.2 GRE Tunneling Code

The python scripts used to create the two IPv6 networks used are as shown below. The scripts were saved in a file known as topo-2sw-2host.py on mininet. The scripts used to create Network 1 and Network 2 on VM1 and VM2 respectively are as shown below.

Network 1 on VM1

```
#!/usr/bin/python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController,
OVSKernelSwitch, UserSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.link import Link, TCLink,Intf

 def topology():

    NODE1_IP='137.158.131.173'
    NODE2_IP='137.158.131.187'

    "Create a network."
    net = Mininet( controller=RemoteController, link=TCLink,
switch=OVSKernelSwitch )

    print ("*** Creating nodes")
    h1 = net.addHost( 'h1', mac='00:00:00:00:00:01',
ip='10.0.0.1/24')
    h2 = net.addHost( 'h2', mac='00:00:00:00:00:02',
ip='10.0.0.2/24')
    s1 = net.addSwitch( 's1', listenPort=6673,
mac='00:00:00:00:00:03' )
    c0 = net.addController( 'c0', controller=RemoteController,
ip='127.0.0.1', port=6633 )

     print ("*** Creating links")
    net.addLink(h1, s1)
    net.addLink(s1, h2)

    print ("*** Starting network")
    net.build()
    c0.start()
    s1.start( [c0] )
    h1.cmd("ifconfig h1-eth0 inet6 add fc00::1/64")
    h2.cmd("ifconfig h2-eth0 inet6 add fc00::2/64")

    # Configure the GRE tunnel
    s1.cmd('ovs-vsctl add-port s1 s1-gre1 -- set interface s1-gre1
type=gre options:remote_ip='+NODE2_IP)
    s1.cmdPrint('ovs-vsctl show')
    Intf( 's1-gre1', node=s1 )

    print ("*** Running CLI")
```

```
  CLI( net )

  print ("*** Stopping network")
  net.stop()

if __name__ == '__main__':
  setLogLevel( 'info' )
  topology()
```

Network 2 on VM2

```
#!/usr/bin/python
from mininet.net import Mininet
from mininet.node import Controller, RemoteController,
OVSKernelSwitch, UserSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.link import Link, TCLink

def topology():
    NODE1_IP='137.158.131.173'
    NODE2_IP='137.158.131.187'

    "Create a network."
    net = Mininet( controller=RemoteController, link=TCLink,
switch=OVSKernelSwitch )

    print ("*** Creating nodes")
    h3 = net.addHost( 'h3', mac='00:00:00:00:00:03',
ip='10.0.0.3/24')
    h4 = net.addHost( 'h4', mac='00:00:00:00:00:04',
ip='10.0.0.4/24')
    s2 = net.addSwitch( 's2', listenPort=6673,
mac='00:00:00:00:00:05' )
    c0 = net.addController( 'c0', controller=RemoteController,
ip='127.0.0.1', port=6633 )

     print ("*** Creating links")
    net.addLink(h3, s2)
    net.addLink(s2, h4)

    print ("*** Starting network")
    net.build()
    c0.start()
    s2.start( [c0] )
    h3.cmd("ifconfig h3-eth0 inet6 add fc00::3/64")
    h4.cmd("ifconfig h4-eth0 inet6 add fc00::4/64")

    # Delete old tunnel if still exists
    s1.cmd('ip tun del s1-gre1')

    # Configure the GRE tunnel
    s2.cmd('ovs-vsctl add-port s2 s2-gre1 -- set interface s2-
gre1 type=gre options:remote_ip='+NODE1_IP)
    s2.cmdPrint('ovs-vsctl show')
```

```
    Intf( 's2-gre1', node=s2 )

    print ("*** Running CLI")
    CLI( net )

    print ("*** Stopping network")
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    topology()
```