Software Defined Virtualized Cloud Radio Access Network (SD-vCRAN) and Programmable EPC for 5G

Pushpanjali Banik

PhD Researcher

Wireless Communications Networking Center

Brunel University

London

Supervisor

Professor Hamed Al-Raweshidy

Abstract

This thesis focuses on proposing a Software Defined Network (SDN) based programmable and capacity optimized backhaul and core network which is critical for 5G network design. Cloud Radio Access networks (CRAN) which is key enabler of 5G networks can address a number of challenges that mobile operators face while trying to support ever-growing end-users' needs towards 5th generation of mobile networks (5G). A novel layered and modular programmable CRAN architecture called Software Defined Virtualised Cloud Radio Access Network (SD-vCRAN) is introduced with Network Function Virtualization (NFV) and Software Defined Network (SDN) capabilities. The SDN-Base Band Unit (BBU) pool is shifted to the programmable core network site, where a centralised SDN controller manages the network servers and virtualised network function entities – Mobile Management Entity (MME), Serving/Packet Data Network Data plane (S/PGW-D), Serving/Packet Data Network Control plane (S/PGW-C), Software Network Defined Baseband Unit (SDN-BBU) and Local controllers (LC) via OpenFlow (OF) protocol. This approach simplifies network operations, improve traffic management, enable system-wide optimisation of Quality of Service (QoS) and network-aware application development. The control plane (excluding the preserved 3GPP standard interfaces: S1-MME, S6a, Gx) managed by the network servers provides load balancing, traffic management and optimisation tools for the data plane.

The proposed work starts by reviewing the requirements of 5G networks, followed by discussion on 5G backhaul and core challenge. Then, an overview of CRAN, Evolved Programmable Core (EPC), SDN, NFV and related works. The simulation details of the proposed architecture are discussed along with the challenges faced by adopting SDN and NFV in mobile core. A thorough assessment of the interfaces and protocols that should be conserved or enhanced on both data and control plane is conducted. The result enables an architecture where the SDN-BBU pool shares a single cloud with the programmable EPC and the control plane is migrated from the network elements to a centralized controller, running on a virtual machine in the mobile core. The data and control plane separation removes overlaps and provides better signalling, as well as efficient network functioning to comply with latency demands. The proposed system performance is validated in terms of throughput, datagram loss, and packet delay variation under three scenarios: 1. single policy installation, 2. multiple policy installation and 3. load balancing. The load balancing performance of proposed system is validated comparing the performance of two different SDN controllers: Floodlight and OpenDaylight, where the later performs better in terms of throughput (no bandwidth restriction), packet loss (below 0.3%) and jitter (below 0.2ms). Furthermore, a detailed comparison of two SDN controller's – Floodlight and OpenDaylight performances is presented, which shows that OpenDaylight performs better only for less dense networks which needs less processing of messages without being blocked, and the Floodlight performs better in ultra-dense network. Some directions and preliminary thoughts for future work and necessary information to operators for building their roadmap to the upcoming technologies is presented.

Preface

This thesis presents a selection of the research work conducted during my PhD study from July 30, 2013 until August 31, 2017 under supervision of Professor Hameed Al-Raweshidy, and Dr Nilavalan Rajagopalan. It is submitted to the Department of Electronics and Communication Engineering at Brunel University in a partial fulfilment of the requirements for the Doctor of Philosophy (PhD) degree.

This PhD project was done in the Wireless Communication Networks Centre at the Department of Electronics and Communication Engineering at Brunel University, London

Acknowledgements

The duration spent as a PhD student is very demanding. Not only does the research require defining a challenging topic but also improve the state of art associated with the topic. During the entire PhD programme, I have had to privilege to attend departmental meetings and conferences, which helped with this research massively. Firstly, I would like to take the opportunity to thank some of the people that made the accomplishment of this thesis possible.

Primarily, I would like to thank my supervisors: Professor Hamed Al-Raweshidy and Dr Rajagopal Nilavalan for their mentoring and support. I am particularly grateful to Professor Hamed Al-Raweshidy and to Wireless Network Communication Center (WNCC) at Brunel University, London for providing me with this excellent opportunity. I am grateful to Mr Muhammad Khan, my fellow colleague for his help in starting this project and asking interesting questions and helping to structure the thesis.

I would like to acknowledge the members of the Wireless Network Communication Center (WNCC) at Brunel University, London for all the discussion, feedbacks and maintaining a pleasant work environment. I would like to extend my gratitude to Mr Sabyasachi Gupta for reviewing the thesis and advising on possible chances of improvements. I am thankful to Brunel University for providing all the required facilities, the environment and multicultural experience during this programme. Most importantly, I am grateful to my parents Mr Tapan Banik, Mrs Shipra Banik and my sibling’s Dr Gitanjali Banik and Mr Debabrata Banik for their consistent support during these four years of my PhD. A most special thanks to Mr Parijat Bhattacharjee who stood by me in all joyful and tough times of my PhD. I had a great privilege to work with so many bright minds, from whom I have learned so much.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1 INTRODUCTION

1.1 MOTIVATION

The fifth generation (5G) of network technologies will be characterised by the shorter setup time, reduced signalling overhead and energy consumption, as well as extremely wide bandwidth that will be available to the user. To provide such services, 5G radio access network (RAN) technologies will require backhaul solutions between the RAN and the packet core capable of dealing with the exponentially increasing traffic load. Operators need to enhance existing technologies and modify certain aspects of traditional systems to cater for the anticipated 1000x capacity [1]. Based on previous consortiums in 5G development, the most promising enabling technologies that are expected to be adopted in 5G networks architecture are: cloud radio access network (CRAN), network function virtualization (NFV) and software defined networking (SDN).

Cloud Radio Access Network (C-RAN) is a new distributed architecture introduced by many organizations such as the Next Generation Mobile Networks (NGMN) project and the European Commission's Seventh Framework Programme to offer a new concept in base station architecture [2] with higher intelligence. The main concept behind C-RAN is to migrate the base band units (BBU) from multiple base stations (BS) into virtualized BBU Pool for centralized processing, while leaving the Remote Radio Heads (RRHs) and antennas at the cell sites. CRAN architecture capitalises on the diversity of traffic peaks, hence, improves the utilisation efficiency of the infrastructure and promotes the green aspect of 5G, owing to close the proximity of cells and users and corresponding lower transmission power requirements [3].

Network Function Virtualization (NFV) technology enables operators to implement network functions in a software running on general purpose computing/ storage platforms [4] to deal with increasing traffic demands. Unlike legacy networks NFV implements network functions on application specific and dedicated hardware and performs software update instead of hardware update. With this migration from hardware to software running in a cloud environment, NFV is expected to lower both equipment cost (CAPEX) and operational cost (OPEX). Services can also be deployed more flexibly and scaled up and down without any delay.

The Software Defined Network (SDN) is referred to as a complementary technology to NFV [5], because it can provide the infrastructure upon which VNF software can run. SDN mainly decouples the network control plane and data plane, allowing direct programmability of network control functions. Furthermore, it enables the underlying infrastructure to be abstracted for applications and network services [6]. SDN architecture (logically) centralizes the network intelligence in software based SDN controllers to gain a global view of the network and allowing the network to appear as a single, logical switch to applications and policy engines. In addition, the network managers can configure, manage, secure, and optimize network resources dynamically via automated SDN programs. Moreover, network managers have independence to write programs on proprietary software. SDN architectures make it possible to implement common network services, including routing, multicast, security, access control, bandwidth management, traffic engineering, quality of service, energy usage, and policy management, as per the operator objectives.

Following the discussions above, CRAN architecture integrated both SDN and NFV technologies has the flexibility and scalability to enable the development of 5G mobile networks.

1.2 PROBLEM STATEMENT

Due to data rate increase in 5G networks, the core and access network will face a sizable growth along with the mobile network densification. 5G networks will use wider channels (e.g., 100MHz channels are currently being discussed), requiring of novel solutions to deliver more capacity, reduced delays, and better reliability while optimizing power consumption in the backhaul link. Furthermore, the different services and applications that will be supported by the 5G networks require efficient bandwidth allocation and management of differentiated QoS. Therefore, the design and operation of the backhaul and mobile core network must take into consideration the QoS levels requirements for different services.

The 5G network architecture has high requirements for latency to support different services, which lead to overloading of one or more gateways and increase in the number of gateway nodes. In the packet core network, a centralized and integrated Serving and Packet Data Network Gateway (S/PGW), which serves a large area of RRHs, such as one province or a state with millions of subscribers. This results in a lot of back and forth IP traffic in the S/PGW [9]. Using traditional packet core network where the gateways unify both data and control will result in a complex gateway service configuration, hence, will immensely increase CAPEX, OPEX, and load balancing issues. The core network control functions in existing architectures cannot be customized for a specific service type and only single set of logical functions are served for diversified services [8]. Moreover, the gateway and control function nodes are deployed by separate dedicated hardware boxes contributing to a larger network size and difficulty in upgrading. The complex interfaces and tightly coupled network control functions affect network management and operation and, service deployment as well. To solve this issues flexible and customizable control function components and implementation of programmable and scalable forwarding plane is an essential necessity of 5G mobile networks. Although, C –RAN architecture is proposed to be the principal part of the next 5G mobile communication networks, the full potential of the centralised base station network architecture cannot be realized until t programmability features are included along with BBU pool virtualisation. This calls for a change in CRAN architecture to increase network efficiency and elasticity.

Thus, the application of SDN and NFV technologies seems attractive to address the above discussed challenges. Further, adoption of these concepts offers the freedom to add network functions and quick upgrades without affecting current network services.   In view of the above mentioned 5G deployment challenges, the 5G backhaul and core network research has been triggered, aiming to provide more degree of freedom to control the backhaul and core network depending on 5G-RAN parameters and control topology for 3GPP interfaces. In this context, the following questions arise:

1. How can operators realize a 5G network architecture by combining CRAN with NFV and SDN, and in which parts of the network can these technologies be deployed?
2. How can SDN / NFV virtualize CRAN, backhaul and mobile core architecture, what are the critical interfaces and protocols that should be preserved?
3. How to achieve realistic backhaul traffic management and better QoS with these technologies?
4. What tools can be used and how would these affect the performance of the proposed SD-vCRAN?

An accurate and up-to-date view of the network status, is enabled by SDN to simply network operation, and achieve higher performance through a centralized traffic management approach. Therefore, we adopt a logically centralized and virtualized network architecture following a SDN/NFV approach for flexible management of backhaul and core network. SDN/NFV based virtualized BBU pool has live network updates, allowing it to dynamically handle mobility and classify different traffic packets in each BBU via OpenFlow (OF) protocol controlled Local Controller (LC). The centralized SDN controller, programs the network entities under its control and dynamically changes the network behaviour using OF. It is implemented as part of the core network to enable the functional split and implement forwarding elements in the data plane, such as OpenFlow (OF) switches. The virtualized network is capable of optimally manage the network resources for mobility management (like reduced service disruption time and load balancing), traffic management in backhaul network, and local offloading via distributed data plane. Floodlight (SDN) controller with QoS module installs and control policies on the path configured with bandwidth limitation for different traffic session types.

1.3 AIM AND OBJECTIVES

1.3.1 Aim

The main goal in this research is to integrate the capabilities of SDN and NFV in the mobile network for a programmable access and core network architecture, which will improve QoS support and traffic management that are essential for 5G and beyond.

1.3.2 Objectives

- To acquire detailed knowledge of 5G, CRAN, SDN and NFV technology and identify the associated challenges.
- To assess the major related research and techniques on 5G network design and optimisation.
- To investigate how to add NFV and SDN capabilities (programmability and virtualization) in the access and core network, and identify the standard 3GPP interfaces/protocols which needs to be updated or preserved.
- Validate the proposed system performance using an emulated testbed under three testing scenarios in terms of throughput, packet delay variation and datagram loss.

- Compare the proposed system load balancing efficiency using two different SDN controllers.
- Conclude and provide guidelines for future modifications of the proposed system.

1.4 CONTRIBUTIONS

This section highlights contributions of this thesis. In this thesis, the author considers load-balancing and QoS challenges associated with 5G mobile network by addressing: 1. control plane functionality, 2. date plane functionality and 3. CRAN signalling with EPC.

1. ***SD-vCRAN-*** The main contribution of this thesis is SD-vCRAN and Programmable EPC– a modular and programmable mobile network architecture for 5G. The proposed architecture integrates CRAN, SDN and NFV technologies to address the challenges in 5G backhaul and core network. The system virtualizes the BBU pool, core network control functions to bring mobility management and increased QoS in the network. The functional split of control and data plane is done to introduce software controlled traffic management in the backhaul network. The mobility management entity (MME) and Home Subscriber Service (HSS) and date plane OF switches are controlled by a logically centralized SDN controller in the cloud core.
2. ***Improved CRAN Architecture*** **–** The BBU pool is virtualized using NFV/SDN technology and migrated to the core to enable resource sharing among multiple operators and reduce signalling to the core. SDN via OF protocol controls the Local controllers (LC) in the BBU pool, which is responsible to handle mobility during handovers. The LC is also responsible for packet classification as well. The new pool functionalities include session management, mobility management, QoS and service continuity necessary, which are necessary for seamless connectivity.
3. ***Analysis of SDN/NFV concept adoption in 5G backhaul and core network*** - This thesis addresses the challenges of integrating SDN/NFV and CRAN to address the challenges in 5G network planning. The interfaces changed or preserved in control plane is discussed. The system design is discussed in context of implementation complexity, performance, scalability, and flexibility.
4. ***Congestion Avoidance and QoS in SDN based 5G core network*** **–** The proposed system handles data explosion by equally distributing the traffic flows into different BBUs and it also manages the handovers between BBUs using Round Robin (RR) strategy. To provide services to diverse types of traffic (data, voice, video) the QoS module in modified from the QoS module in [49] to provide service class based routing decisions.
5. ***Verification of SD-vCRAN*** – Practical trial consisted of a network simulation environment to conduct a careful assessment of SD-vCRAN performance for different scenarios and topologies. Mininet was used for simulation purpose and traffic was generated using *iperf*. The virtual testbed focused on load balancing and QoS for proposed system and examining the date plane performance under different scenarios with different loads. The system considers a Floodlight controller for deployment and compares the performance of the controller with an OpenDaylight controller under specified configurations.

1.5 THESIS OUTLINE

This thesis approaches the issues starting from a broader definition of technologies and introduction to the context, followed by a proposed system design and practical trial. Hereby, the work has been structured in five main chapters as follows:

Chapter 2 presents an overview of 5G network, CRAN and its drawbacks and solutions, EPC architecture and its functionality along with the protocol governing the data plane. NFV and SDN are discussed in context of 5G network enabling technologies.

Chapter 3 the proposed system for 5G backhaul and core network – SD-vCRAN along with associated design challenges is discussed. The optimization of the proposed network is analysed in terms of flexibility, complexity, and performance. The tools required for system verification are briefly discussed.

Chapter 4 the virtual testbed is examined for different scenarios to check if functioning is as expected. This chapter consists of an analysis of the obtained results and a comparison between the two controllers.

Chapter 5 aims to draw the final remarks and conclusions of the presented work. Proposed optimizations and future research directions are also presented.

CHAPTER 2

2 OVERVIEW OF 5G NETWORKS, CLOUD RADIO ACCESS NETWORK, EVOLVED PACKET CORE, NETWORK FUNCTION VIRTUALIZATION AND SOFTWARE DEFINED NETWORKING CONCEPTS

Current mobile network architecture was designed to provide services for voice and conventional mobile broadband (MBB), thus the network fails to be flexible and support diversified services requiring more data rate wide range, less latency, better traffic management. Additionally, the network has complex interfaces and substantial number of network elements. This chapter presents the 5G architecture focusing on the core and backhaul network. The challenges in 5G backhaul and core network along with proposed solutions are identified. CRAN adoption allows end-to -end network slicing, on demand deployment of service anchors and functions needed for 5G network. NFV/SDN support the underlying physical infrastructure required to virtualize access, transport, and core networks essential for 5G services. Therefore, a study of driving forces behind 5G realization, survey on literature and implementation concepts of these techniques is presented and the interdependencies between them.

2.1 STATE OF THE ART

The backhaul and core network research is triggered by defining and understanding the features that affect the backhaul and core network, as well as technologies that will be incorporated into the network. 4G America [16], IMT-2020 [17], 2020 and beyond [18], 5G forum [19] and 5G Private Public Partnership (5G PPP) [20] are the major global 5G projects that are responsible to deliver 5G specifications by 2020. According to Mobile and Wireless Communications Enablers for the Twenty-Twenty Information Society (METIS), 5G network will target 1000× capacity increase, 10-100 more connected devices, extreme data rate hike (in Gbps) and low latency (less than 1ms) [13], [21]. 5G needs to push the envelope of performance to provide increased throughput, lower latency, ultrahigh reliability, higher connectivity density, and higher mobility range [42]. These stringent requirements directly affect the backhaul and will cause a data explosion. In [14] a detailed listing of typical 5G user experience and system performance requirements are presented. Therefore, a 5G backhaul and core should flexible and adaptive in such a way that all services are catered with efficient traffic management and considering QoS obligations.

It becomes the convincing view worldwide that the 5G architecture will be based on CRAN, SDN and NFV, due to the simplicity, flexibility, and openness [43]– [47]. However, one of the challenges lies in how to virtualize BBU pool to gain better mobility and maximum resource sharing. Though, limited work has focused on specific aspect of the C-RAN virtualization technique. The authors of the [22] paper have outlined virtualization requirements and propose network architecture for

Cloud-RAN base stations. They also present arguments for abstracting the EPC's view of the physical hardware connected to a C-RAN BS. In [3] an overview of optical fronthaul technologies, efficient and scalable solutions are presented, and finally an overlook into which 5G service requirements may further impact future backhaul, midhaul, and fronthaul networks was provided. In [15], the novel RANaaS concept is proposed, which leverages cloud technologies to implement a flexible functional split in 5G mobile networks enabling optimized usage of spectral, energy, and computational resources in ultra-dense deployments. RANaaS will allow for more flexibility of RAN deployments under homogeneous and heterogeneous backhaul.

SDN and NFV in 5G backhaul promises new network efficiencies and more dynamic, application aware backhaul network. 5G architecture based on SDN/NFV, can utilize the global view feature to get information for both core and backhaul networks, various network functions, readily deploy load balancing and traffic engineering in the centralized controller to improve flexibility, efficiency, and robustness in the network [48]. Network planners responsible for backhaul and core networks design need a set of practices to guide design and implementation decisions for SDN and NFV. Major concern for network planners is how SDN fits into access and aggregation domains for mobile networks. In paper [23] the ETSI NFV standardization body has defined several use cases for a virtualized evolved packet core (EPC). One of the major research topic is aimed to optimize the existing mobile architecture for data increase and efficient signalling. OpenFlow protocol is extended in [24] to migrate control plane components into cloud and preserve Tunnel Endpoint Identifier (TEID) for GPRS Tunneling Protocol User Plane (GTP-U) encapsulation/ decapsulation support. The authors in [24] addresses the same challenge of implementing the user plane of the GTP protocol in an SDN-based switch-Open vSwitch, without considering overloaded packet header scenario. In [25] enhanced OF network element Networking Element (NE)+ is proposed to address the GPRS Tunneling Protocol User Plane (GTP-U) encapsulation/ decapsulation challenge. In [26] a novel mobility management scheme for the Long-Term Evolution (LTE) networks is proposed, utilizing SDN concept. The architecture moves a part of mobility management tasks from the EPC to the backhaul network. Further, in [27], an OpenFlow-based control plane for LTE/EPC architectures is presented, to guarantee on demand connectivity service using load balancing use cases. In [28], a scalable architecture enables operators to realize high-level service policies by performing fine-grained packet classification at access switches. The idea completely replaces 3GPP functions and brings changes in data and control plane. The only backdraw is its interoperability with legacy networks.

Load balancing solutions by integrating SDN concept has gained momentum due to bandwidth and scalability demand in future networks. Hedera system is proposed in [29], to enable dynamic flow scheduling using commodity switched and unmodified hosts for a multi-stage switching fabric and efficiently utilize aggregate network resources. This paper examines the performance of state-of-the-art hash-based ECMP load-balancing algorithm with of two dynamic scheduling algorithms - Global First Fit and Simulated Annealing. The research shows that Simulated Annealing almost always can deliver optimal bisection (approx.) bandwidth for data centres networks consisting of thousands of nodes by reducing the searching space. In paper [30] an adaptive algorithm based on linear prediction is presented to determine the load on each switch and the overhead in the network produced by the load information. The primary goal is to minimize the load on the switch flow tables and measure flows for anomaly detection applications. The greedy based rule algorithm dynamically zooms into the flow space to reduce rule assignment in switches. Further, in [31] the authors proposed a single flow table and group flow table based dynamic algorithm targeted to reduce the large flow table matching range. The dynamic flow table design provides enhanced traffic scheduling and network performance. The paper [32], present a strategy for load balancing based on SDN, using round robin algorithm, and draws a comparison with existing random based algorithm in load balancers. A study of virtual machine use cases is conducted and an optimal traffic engineering to balance load of virtual machines is conducted in [33]. Addressing load balancing challenges in networks in [34], SDN based hybrid routing (combination of destination based and explicit routing) for multiple traffic matrices is presented. Focusing on the underutilization for the ECMP supported by the fat-tree topology in data centres, in [35] an algorithm is implemented on current network infrastructure to achieve efficient load balancing in the network. The algorithm checks for network performance status and looks for alternative path for re-routing based on the information from the centralized controller. The authors in [36], a dynamic load balancing algorithm is implemented in large networks to shift load from best calculated path to avoid data loss, only when the link on load is greater than link bandwidth. A similar work is presented in [37] based on round robin strategy. SDN is adopted in LTE core to gain load balancing by deploying a system which collects statistics from the OF switches periodically at 1-second intervals and forwards to the controller, which keeps the general state of traffic load in the network [38].

For SDN-based 5G networks a traffic engineering (TE) framework that incorporates the data gateway (D-GW) selection and exploits the topology and traffic information of both core and backhaul networks is proposed in [39]. The framework deploys multiple BSs to one data gateway association (MBODA) and multiple flows to one data gateway association strategy. The placement of SDN controller for a 5G network with virtualized network entities is examined in [40]. Focusing on the trade-off needed between reducing the SGW relocation frequency and balancing the traffic load among the underlying SGW control (SGW-C) VNFs, an optimisation model and fair solution is derived using Nash Bargaining game and the threat point.

In [41], the approach of dynamic load balancing is offered to set optimal and reserve routes according to the set indicator of QoS, load balancing along the paved routes based on paired shifts data, optimization, and alignment of load according to indicators of route variation and jitter value. The system uses statistics and modelling of different topologies for forecasting of further work of real networks to provide reliable service. OpenQoS based on QoS routing, where the routes of multimedia traffic are optimized dynamically to serve end-to end QoS requirement is shown to reduce effects of packet loss and latency [49]. Further, in [50] the authors highlight the fact that SDN and OpenFlow assist adoption of traditional-based networking techniques like QoS, load balancing and routing to SDN approaches. Nevertheless, there are several drawbacks which should be examined or considered when integrating these solutions into existing architectures such as: the complexity of the proposed algorithms and 5G networks stringent QoS obligation.

2.2 FIFTH GENERATION MOBILE NETWORK (5G)

Societal changes, witnessed since the explosion of data services, and the growing appetite for wireless broadband have incentivised the need to change the way mobile and wireless communications are used. The need for speedy 5G system development is inevitable to provide the proliferating essential services, which are going to create an avalanche of mobile and wireless traffic volume over the next decade [52] [53]. Further, the human-centric communications will contribute to massive increase in number of connected devices (total of 50 billion connected devices by 2020) [54]. The new different applications impose tight requirements which the generation (5G) will have to support such as - stringent latency and reliability requirements, excellent range of data rates (multiple gigabits per second), scalability and flexibility of network without adding complexity and power consumption. Efficiency and scalability are therefore key design criteria in 5G networks to satisfy all these requirements or replace legacy networks. 5G vision as mentioned in [13] [17] [20] [21] [55], can be realized through a flexible combination of evolved existing technologies and new radio concepts. The key performance indicators in a 5G environment for the below scenarios are summarized in Table 1:

- Providing very high data rates for future Mobile Broad Band (MBB) users to get fast connectivity without delays.
- Congestion control and excellent user experience in ultra-dense networks (UDN) without facing service denials due to traffic overload.
- Improved quality of experience (QoE) such that user is provided data rate of at least 100 Mb/s in the downlink and 20 Mb/s in the uplink, while maintaining end-to-end latencies below 100 ms. The network should have less cost pressure as well as optimized power consumption.
- Increased QoS to offer different services with very strict requirements on latency and reliability.
- Flexibility and scalability to integrate innovative technologies and support legacy networks.

Requirements	**Expected Value**
Data Rate	Downlink peak is 20 Gbps Uplink peak is 10Gbps
Latency	4ms for eMBB 1ms for URLLC
Connected Devices	300 million/ AP
Data Volume	9 Gb/host (busy period), 500Gb/mobile/subscriber
Reliability	99.99%

Table 1: 5G Key Performance Indicators, source [55]

2.2.1 Challenges in 5G backhaul and core network

2.2.1.1 Challenges

Novel traffic differentiation schemes unlike the legacy network is a key requirement in 5G networks to satisfy the end to end (E2E) QoS requirements. The diversified services in 5G which will be hosted in multi domain should be supported by dynamic and flexible provisioning of network functionalities. Further, to guarantee service delivery to end users without facing over the top (OTT) issue while sending traffic across multiple networks, the need for tailored autonomic network management platform arises. The large data volume requirement for UDNs and processing of the massive data in 5G network needs efficient network management to assure QoS even if network context changes. Additionally, in the backhaul the cost to transport traffic to core network increases because of user throughput requirement (ranges from 20G for downlink and 10G for uplink). Therefore, the huge data load lead to load imbalance and congestion in network. To achieve efficient resource utilization, traffic management is essential in 5G timeframe to handle the huge amount of traffic with strict QoE requirement. This situation calls for exploitation of traffic and network information in backhaul and core network and updating RAN architecture to gain increased network efficiency. The network will benefit by gaining flexibility in assigning network resource to service flows and the individual flow information from the backhaul and core networks.

2.2.1.2 Solutions

Network Programmability: a concept that involves network softwarization and virtualization of infrastructure and network functions [20]. Network softwarization is key enabler for incorporating E2E service and management platforms, facilitate network slicing (separation of planes and resources) for operational purposes and manage control plane functions responsible for QoS control. Software-based implementations and virtualization technologies will be essential in 5G networks for flexibility and cost control [20]. Softwarization will enable network to optimize and configure resources and functionalities to serve QoS requirements without increasing cost pressure. A programmable backhaul network will serve as a platform which will adapt the operation of backhaul to the RAN needs in context of applications and services (user and network). SDN automated network optimization ensure optimal routing across the backhaul networks. In 5G environment, majority core network functions and planes are expected to be deployed VNFs to avoid massive hardware change and improve portability, while each function remains unaware of the underlying operating system. SDN principles will help service plane splits and flexible VNFs use in core site depending on QoS requirements, processing, and storage capacity. SDN/NFV software network technologies deployed in the backhaul and core network will allow it to be programmable and apply policies that will adapt as per the network obligations and operating conditions. Softwarization will add increased flexibility and access to real time network view always, to gain information needed for global optimal resource allocation in backhaul and core network.

2.2.2 5G Enablers for core and backhaul network

The key enablers to satisfy the 5G backhaul and core network demands are briefly outlined in this section. CRAN recently attracted a great deal of attention as one possible way to realize 5G KPIs.

In C-RAN, multiple sites are connected to a central data centres where all the baseband (BB) processing is performed [4]. The centralized processing and multi-connectivity attributes allow servicing of RAN resources (non-real time) via on demand network utilization. On the other hand, 5G networks needs to be flexible and adaptive to network changes and use cases. SDN/NFV is a potential solution to gain the previously mentioned features through simplified network view and centralized control. Figure 2.2-1, shows a SDN based virtualized CRAN and programmable backhaul and core network for 5G. The proposed architecture is discussed in detail in Chapter 3. The core network consists of logically centralized SDN controller and BBU pool, and OF switches as forwarding elements to enable. Software controlled core network allows gateway control and data plane separation to reduce cost of distributed gateways, load balancing and better signalling interaction with BBUs. The SDN/NFV concept based architecture implements policies depending on user and static network data stored in the core network database. Programmable data plane and control plane allows network to select control or data plane based on different service. The routing decision is done based on the network topology and QoS requirements to achieve efficiency goals. The resource management is practised in the control plane to load balance user traffic based on actual network congestion for better backhaul utilization [56] and absorb traffic bursts for improved user QoE. Therefore, it is safe to comment that SDN enables the RAN and backhaul network to correlate end user performance with current network status to optimize routing, QoS, and resource allocation for better network performance.

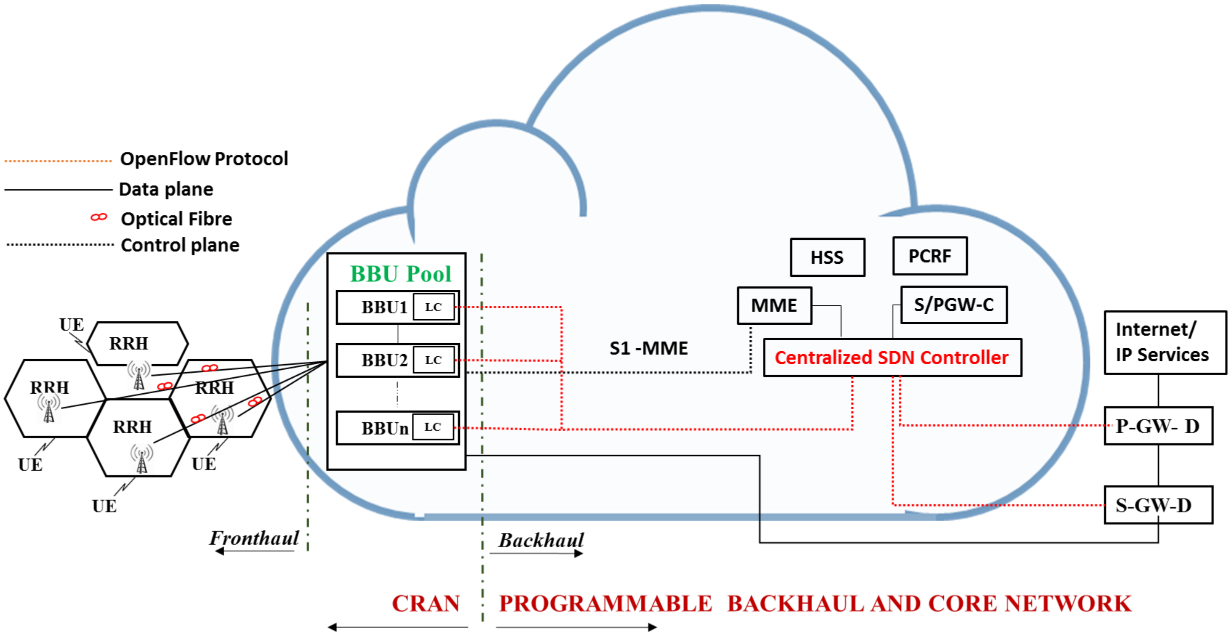


Figure 2.2-1 Proposed architecture - Software Defined Virtualized Cloud Radio Access Network (SD-vCRAN) and Programmable EPC for 5G

2.3 CLOUD RADIO ACCESS NETWORK (CRAN)

As operating frequency for the LTE standard is approaching the Shannon limit, the most prominent way to increase coverage is by either adding more cells, resulting in massive estate requirements. Additionally, RAN power consumption and complex interfaces due to heterogenous networks contribute to cost hike. On the other hand, 5G networks will be characterized by massive data traffic, higher mobile subscriptions and new services with strict QoS requirements. Therefore, operators need an intelligent approach guaranteeing network management and service.

In [57] a novel mobile architecture C-RAN was first proposed, which has the potential to answer the above-mentioned challenges. In C-RAN, baseband processing is centralized in a virtualized BBU Pool. The BBUs adapt to non-uniform traffic and implement joint processing technologies to improve system performance necessary for future mobile networks. The reduction in number of BBUs lead to decrease in network operation cost, as improved energy optimization is attained compared to traditional architectures. The BBU pool to be shared by different network operators and interact with lower delays to increase spectral efficiency because of virtualization of the same. Hence, CRAN acts as a platform for coexistence of diverse services. Compared to traditional network, CRAN has the potential to facilitate easy upgrades, load balancing between cells and efficient intra-BBU pool handover, thereby improving scalability and performance. Even though there are other prospective candidates to address traditional network challenges but they fail to employ collaborative features like CRAN, remain underutilized during off-peaks and equally difficult to upgrade and repair [59] [60] [61].

2.3.1 Architecture

In C-RAN, baseband processing is centralized in a virtualized BBU Pool which is connected to RRH via optical fiber. The link connecting BBU and RRH is called fronthaul and BBU and EPC is called backhaul. BBU is responsible to provide MAC PHY and Antenna Array System (AAS) functionalities, and RRH functions to provide high data rate for UEs [58].

For a given area all the RRHs are handled by a single BBU pool. Inter cell coordination (ICI) is made significantly easier due to direct communication within BBUs. Figure 2.3-1 shows CRAN architecture for LTE EPC.

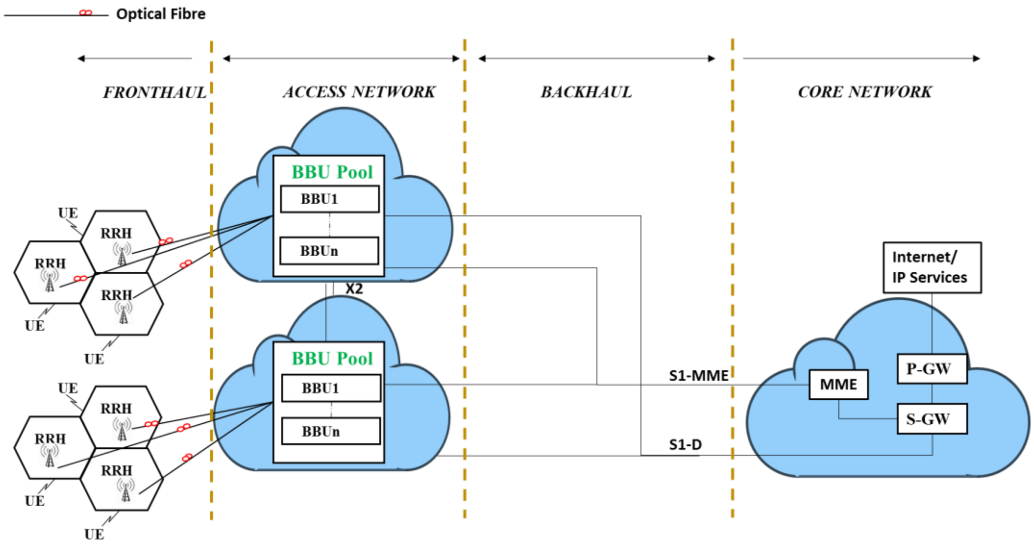


Figure 2.3-1 CRAN architecture for LTE EPC

The CRAN can be deployed in three different structures depending on the degree of centralization as presented below:

1. ***Full centralization***: In this structure, layer 1, layer 2 and layer 3 base station functions are in BBU [2]. This solution provides several benefits like easier upgrading, efficient resources sharing and support sharing between different network operators. However, the need for high bandwidth in future networks like 5G will lead to challenges in fronthaul in networks having full centralization.
2. ***Partial centralization***: The RRH incorporates layer 1 and radio functionalities. Layer 2 and 3 functions are continued in the BBU.
3. ***Hybrid centralization***: In this structure, Layer 1 functions in the BBU pool are partly migrated to a separate processing unit, which may be a part of the BBU pool to gain maximum resource sharing and reduce BBU energy consumption.

2.3.2 CRAN benefits towards 5G

The telecommunication industry now has a reasonable view of 5G service requirements, and network providers with cutting deployment timelines for rapid deployment of 5G want to introduce CRAN paradigm and concepts to meet the timelines. The operators are driven by the benefits CRAN can offer towards 5G. Some of the major advantages are briefed below-

- Cost and Energy Optimization: BBU pooling and placement in cloud both cost and energy needed for network operation is reduced.
- Spectral efficiency enhancement: Frequency reuse at the cell site improves the spectral efficiency by decreasing resource allocation and base station densification at least provides a marginal gain in resource allocation by improving Signal to Interface (SIR) reliability.
- Support legacy network and multiple standards: Traditional network and technologies, including GSM, UMTS, HSDPA, LTE and LTE-A are supported by CRAN by interconnecting and managing with these technologies directly through network elements.
- Improved security and adaptability: Centralized processing and encryption permits to adapt to non- uniform traffic and develop better security algorithms for future mobile networks.
- Improving network management and flexibility: Management of resources and control of network functionalities is easier in CRAN due to centralized processing.
- QoS support: CRAN supports different services with specific QoS metrics through hierarchal QoS (QoS) [58].
- Decrease CAPEX and OPEX: CRAN can attain 15% CAPEX and 50% OPEX decrease owing to its centralized processing characteristic.

The essential drivers for operators wanting to deploy CRAN as a solution for network challenges are identified in a study conducted by Light Reading [63].

Driver	Percentage
Reduce OPEX	27%
Reduce CAPEX	25%
Increase Flexibility	23%
Improve Scalability	12%
Other	13%

Table 2: Essential Drivers for operators to deploy CRAN

2.3.3 Challenges in CRAN architecture

2.3.3.1 Fronthaul

In C-RAN with centralization, the fiber fronthaul adds a huge data volume. The bandwidth and capacity requirement in fronthaul is comparatively 50x higher than in backhaul [62]. Further, depending on the deployment structure of CRAN, the degree of centralization directly affects the amount of fiber resources needed. In other words, full centralization needs considerable number of fiber resources leading to cost increase.

Furthermore, the increased elasticity and efficiency on the radio access side comes at the expense of fronthaul connectivity challenges. The fronthaul interface - CPRI is a digitised, serial radio interface with capacities in the range of multiple Gbps. CPRI has very strict requirements on transmission latency and jitter to assure faultless system operation and therefore superior user experience. Major research has been done to propose novel approaches able to solve the fronthaul challenge, for example the following:

1. Cable/Optical solution- Optical fronthaul solutions such as dedicated fibers, Wavelength Division Multiplexing Passive Optical Network (WDM-PON), Coarse-WDM (CWDM) PON, and Dense WDM (DWDM) PON can be employed to mitigate the fronthaul constraints [95] [96].
2. Wireless Fidelity (Wi-Fi) solution- Telecommunication service providers like Huawei, Qualcomm, and Ericsson propose to employ the unlicensed spectrum around 5 GHz frequency band that has been utilized by the wireless fidelity (Wi-Fi) for the 5G LTE-Unlicensed (LTE-U) mobile communications as a fronthaul solution [97]. By this separate frequency for the fronthaul is not required by the MNOs and the unlicensed spectrum can be efficiently reused in both fronthaul and access links.
3. Microwave transmission (MM-Wave) solution- The mm-wave solutions are feasible because of the availability of unexploited huge spectrum capacity in the 70-80 GHz E-band with 60 GHz more appropriate for 5G mobile communications [98] [99]. The mm-wave fronthaul solution can be achieved by the in-band (the fronthaul and the access links share the mm-wave spectrum) and out-of-band (the band employed by the fronthaul link is different from that of the access link) schemes.

2.3.3.2 CRAN Virtualization

New virtualization techniques need to be proposed, which will permit the BBU pool to realize the full potential of cloudification feature. On the other hand, the processing in access networks is complicated which cannot meet the requirement of dynamic real-time processing ability. The possible solution will be proposing a virtual network framework which will permit moving mobility to BBU and reduce signalling load in the core network. The ability to classify traffic types will benefit the CRAN architecture in scenarios including machine to machine communications which have low service level demanding traffic compared to user equipment (UE). The packet classification feature will be a bonus in resource sharing scenarios for multi-standard operators and support QoS requirements. Each operator can implement different mobility standards and offer different service levels depending on the other operator infrastructure requirement. NFV and SDN are possible candidates that can introduce virtualisation and programmability in C-RAN. In this thesis, we focus on the virtualization challenge in CRAN architecture.

2.4 EVOLVED PACKET CORE (EPC)

The EPC resulted from the progression in mobile core network and combined with LTE form the Evolved Packet System (EPS). In contrast to its predecessors like Global System Mobile Communication (GSM)/ General Packet Radio Service (GPRS) it is entirely packet switched with all data sent using IP. The EPC has a flat architecture with less network nodes, less delay, and better data rate support. The EPC is mainly responsible for mobility management and support QoS demand for MBB network. Packet network connectivity is ensured via mobility management when user moves from one BS. EPC functions to provide the user with acceptable QoS through efficient resource allocation to the terminals. Figure 2.4-1 shows the LTE/EPC architecture for 3GPP access systems [64]. The relationship between the network entities and the interfaces that connect them are discussed in this section.

2.4.1 Core Components and Functions

2.4.1.1 Mobility Management Entity (MME)

The mobility management is the main control node in LTE access network, as it handles mobility related signalling in the control plane (S1-MME interface). The MME functions are as follows:

- Bearer activation/deactivation procedures and selection of SGW for UE during *initial attachment* or intra BS handover.
- Control UE mode and support Tracking Area (TA), as well as paging procedures during retransmission.
- Authentication of UE towards HSS
- Termination of S6a interface and Non-Access Stratum (NAS) signalling which generates and allocates UE identities like Global Unique Temporary ID (GUTI) at MME point, when user is in roaming.
- Provide mobility between LTE and previous technologies like 2G/3G via S3 interface.
- Provide ciphering and encryption for NAS signalling.

2.4.1.2 Home Subscriber Service (HSS)

The HSS is responsible for user authentication, authorization, and accounting. The server contains all information related to users and subscribers, as well as support functions for mobility management.

2.4.1.3 Serving Gateway (SGW)

The serving gateway is the data plane component in EPC, which provides connection with RAN through S1-U interface. The main SGW functions are:

- Provide packet transfer along user plane and act as the mobility anchor in case of inter BS handover.
- Serve all UE associated with the EPC at a certain point of time
- Monitor and hold information related to idle mode of UE and generate paging request when data arrives from internet toward UE
- Manage mobility interface to other networks (2G/3G).

Table 3 lists the S-/PGW functions for main EPC scenarios [25].

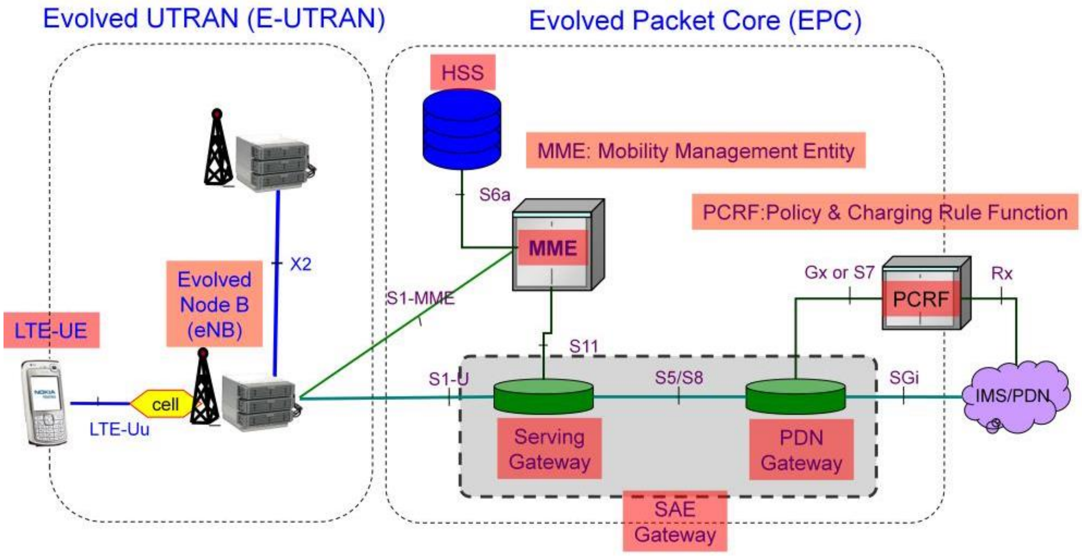


Figure 2.4-1 EPC architecture for 3GPP access system, source [64]

2.4.1.4 Packet Data Network Gateway (PDN-GW)

The packet data network gateway connects the EPC to external networks such as the internet and corporate intranets by *SGi* interface. Main functions are:

- Responsible to act as mobility anchor point between 3GPP and non-3GPP technologies.
- Provide connectivity to UE by being the point of entry or exit of traffic for the UE.
- Handle policy enforcement, packet filtration for users, charging support and Lawful Interception (LI).

2.4.1.5 Policy Control Enforcement Function (PCRF)

The PCRF is responsible for implementing policy with respect to QoS, and controlling the charging functionalities in the Policy Control Enforcement Function (PCEF), which located in the PGW. It controls and manages the data flow during transmission and provides appropriate interfaces towards charging and billing systems, as well as it enables new business models.

Scenario	Entity	Signaling	Resource Management	User plane Forwarding Rules	User Plane Forwarding	GTP	Filtering	Charging
UE Attach	SGW PGW	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	No Yes	No Yes
UE Detach	SGW PGW	Yes Yes	Yes Yes	Yes Yes	No No	No No	No No	No No
S1-U Bearer Release	SGW PGW	Yes No	Yes No	Yes No	No No	No No	No No	No No
Service Request (Default-B)	SGW PGW	Yes No	Yes No	Yes No	Yes No	Yes No	No No	No No
Service Request (Dedicated-B)	SGW PGW	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	No Yes	No Yes
Tracking Area Update	SGW PGW	Yes Yes	Yes Yes	Yes Yes	No No	No No	No No	No No
Intra Handover with/without X2 Support	SGW PGW	Yes No	Yes No	Yes No	Yes No	Yes No	No No	No No
Handover from LTE to 2G/3G	SGW PGW	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	No Yes	No No

Table 3: S-/PGW functions for main EPC scenarios

2.4.2 GPRS Tunneling Protocol (GTP)

The GTP provides the transport and signaling for implementing the mobility management and QoS functions across base stations. It is the main protocol in EPC, which basically adds a small header consisting of two identifiers: source and destination tunnel endpoint identifier for the uplink and downlink respectively, along with all other information in the GTP tunnel. The unidirectional GTP tunnels extends as one set between the base station and the SGW and another set between the SGW and PDN GW. Based on the control and user plane the GTP protocol has two components:

1. ***GPRS Tunneling Protocol Control Plane (GTP-C)***: a control plane protocol that supports establishment of tunnels for mobility management and bearers for QoS management that matches wired backhaul and packet core QoS to radio link QoS [65]. UDP is used as transport protocol for core network signaling by GTP-C.
2. ***GPRS Tunneling Protocol User Plane (GTP- U)***: a data plane protocol used for implementing tunnels between network elements that act as routers, specifically the Serving Gateway, PDN Gateway, and the wired network side of the radio base station. GTP-U runs as an application on top of UDP, and the UDP destination port number for GTP-U is fixed, while the source port number is dynamically allocated by the sending node [66].

The bearer which is transmitted on the LTE radio interface from UE to eNodeB, is mapped onto a GTP tunnel by the PCRF. The GTP tunnel is transmitted over IP network towards a SGW, which further encapsulate it into a new GTP tunnel towards a PGW. Finally, the PGW decapsulates the traffic and transmits it out of the EPC (the *SGi* interface in Figure 2.3-1), towards its destination (e.g. the Internet or a service in the mobile network). The PGW is the default GW for the UEs since it assigns the IP address to UE. The PDN provides IP connectivity to the UE throughout the UE and PDN connection lifetime, once one EPS bearer is established. An EPS bearer is the scale of QoS control for bearer level in EPC, which is responsible to treat all the mapped traffic with same bearer level of packet forwarding. The established bearer is referred as the default bearer and any new EPS bearer for the same PDN connection is referred as dedicated bearer.

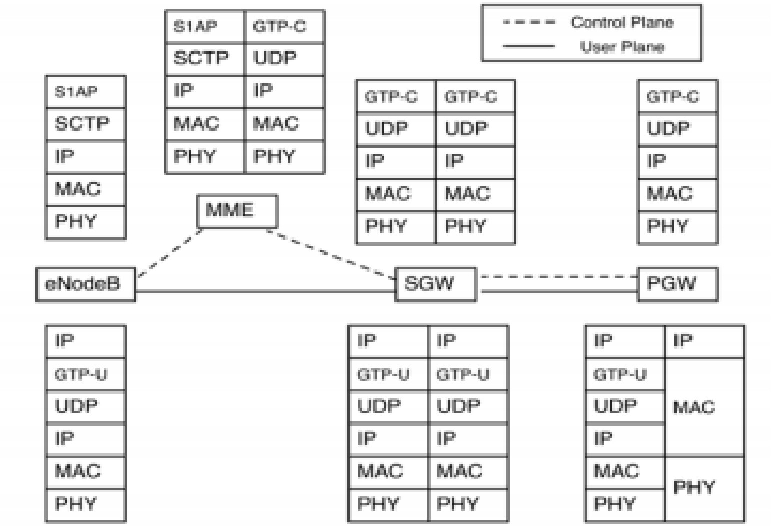


Figure 2.4-2 Protocol Stack in LTE-EPC, source [65, 66]

2.4.3 QoE and QoS in LTE-EPC

LTE has a standardized QoS architecture defined under 3GPP specifications, which is mainly used to handle MBB for different data and voice services. Flexible LTE QoS framework is based on bearer concept, which provide services to applications requiring increased data rate with reliable and secure connections. The architecture sets QoS priorities like bit rates, packet delay, packet loss, bit error rate and scheduling policy, of the transport channel depending on its class and based on the level of priority the bearer is associated with a guaranteed bit rate (GBR) or non-guaranteed bit rate (non-GBR).

In LTE networks QoS is implemented between UE and PDN and it is applied to a group of bearers. The set of bearers (which may comprise of radio bearer, S1 bearer and S5/S8 bearer) collectively forms the Evolved Packet System (EPS). Bearer provides a virtual path with distinct QoS between UE and PDN-GW, where the flows contained in a single bearer are forwarded and treated in the same manner (e.g. scheduling, queue management, rate shaping, link layer configuration, etc.) between the UE and PDN-GW. There are two types of bearer as discussed below:

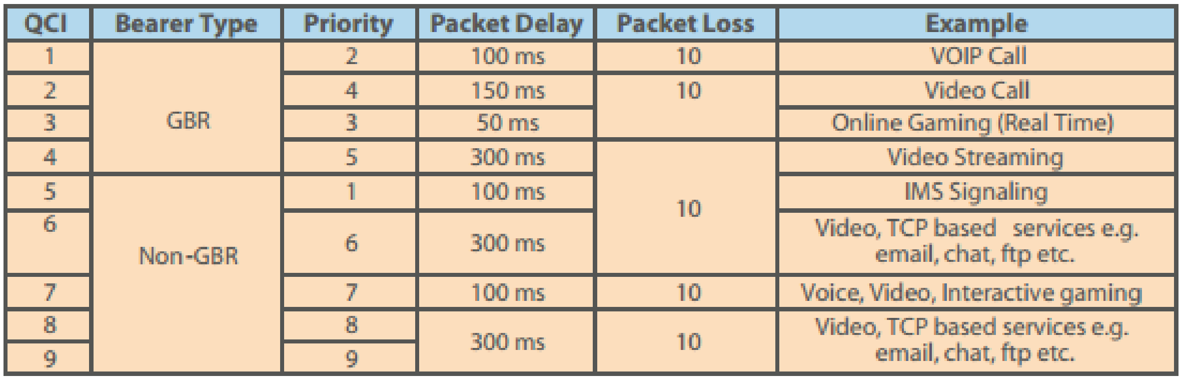
1. ***Default Bearer***: During UE initial attachment to the LTE network it is assigned the default bearer based on its service demands, which is maintained throughout its connection with the mobile network. Each default bearer has separate IP address and supports only best effort (BE) services.
2. ***Dedicated Bearer***: Dedicated bearers function as dedicated tunnels to provide suitable treatment to specific services like VoIP and video. Each dedicated bearer doesn't require separate IP address and supports both GBR and non-GBR services.

Dedicated bearer is classified further into:

1. ***Guaranteed Bit Rate (GBR) bearers*****:** are mainly used for real-time applications, such as conversational voice and VoIP. A GBR bearer blocks all transmission resources since it has a minimum amount of bandwidth that is reserved by the network during admission control function. GBR bearers do not experience congestion related to packet loss, if the traffic carried by the GBR bearer adheres to the QoS value associated with the bearer.
2. ***Non-GBR bearers*****:** do not have specific network bandwidth allocation, as well as doesn't guarantee any specific bit rate service. Non-GBR bearers are highly prone to congestion and are dedicated to best-effort services, such as FTP transfer and Internet browsing.

2.4.3.1 QoS Parameters

A bearer is characterized by two or four QoS parameters mentioned below, depending on whether it is a real-time or best effort service:



QCI	Bearer Type	Priority	Packet Delay	Packet Loss	Example
1	GBR	2	100 ms	10	VOIP Call
2		4	150 ms	10	Video Call
3		3	50 ms		Online Gaming (Real Time)
4		5	300 ms	10	Video Streaming
5	Non-GBR	1	100 ms		IMS Signaling
6		6	300 ms		Video, TCP based services e.g. email, chat, ftp etc.
7		7	100 ms	10	Voice, Video, Interactive gaming
8		8	300 ms	10	Video, TCP based services e.g. email, chat, ftp etc.
9		9			

Figure 2.4-3: 3GPP Standardized QCI, source [67]

1. ***QoS Class Indicator (QCI)*****:** specifies the forwarding treatment of the IP packets received on a specific bearer. Each functional node in the mobile network handles the packet forwarding of traffic traversing a bearer. The QCI values influence several node-specific parameters, such as link layer protocol configuration, admission thresholds, scheduling weights, and queue management thresholds.
2. ***Allocation and Retention Priority (ALRP)*****:** This priority parameter specifies the control-plane data flow treatment. The ALRP enables bearer establishment and modification, as well as connection setup and release during network congestion or conflicts in demand

for network resources. Address Resolution Protocol (ARP) can be used by the eNodeB to decide which data flow should be released to lower pressure on network capacity.

3. ***Maximum Bit Rate (MBR)*****:** is applicable only for real-time services and it is specified for GBR bearers. The MBR specifies the bit rate that the traffic carried by the bearer should not exceed. 3GPP adds an additional parameter called the Aggregate Maximum Bit Rate (AMBR) for all non-GBR bearers, which specify the total amount of bit rate consumed per subscriber.
4. ***Guaranteed Bit Rate*****:** is defined for GBR bearer only. GBR specifies the guaranteed bit rate for a specific bearer in the mobile network. GBR is the maximum bit rate that is allowed by the system as per the 3GPP Release 8 and beyond.

2.4.3.2 Policy Control and QoS Provisioning

The EPC includes a Policy Control and Charging (PCC) architecture that provides support for QoS and enables dynamic control of the QoS and charging requirements of the services which the application servers deliver. The PCC offers improved support for roaming and assist network operators to monetize their investment by providing customers with a variety of QoS and charging options when choosing a service. The PCRF is the part of the PCC architecture that aggregates information to and from the network, operational support system and other sources (such as portals) in real time, supporting the creation of rules and then automatically making policy decisions for each subscriber active on the network [68]. PCRF establishes the calls and allocates the requested bandwidth to the call bearer with configured attributes, which enables an operator to offer differentiated voice services to customers at a premium charge, Further, it provides options for prioritizing the calls to emergency numbers in the 5G timeframe. The PCRF filters the data flows based on an IP five-tuple: source IP address, destination IP address, source port number, destination port number, and protocol identification (Transmission Control Protocol (TCP) or UDP). The availability of varied interfaces in the EPC make it easy to integrate the PCRF with any type of mobile or fixed broadband network. PCRF interfaces with the PGW and determines the QoS requirements upon three criteria: application requirement, subscription information and policy of the operator. It takes charging enforcement decisions and based on the PCRF decision a dedicated bearer is established. For example, depending on subscriber demand network providers can use PCRF to charge them based on volume of high-bandwidth applications usage, extra QoS guarantees or roaming usage.

QoS provisioning in LTE includes various mechanisms as mentioned below [69]:

- ***Scheduling***: Resource allocation process among users involved in data transmission. In LTE, Orthogonal Frequency Division Multiple Access (OFDMA) is used for downlink transmission and Single-Carrier Frequency Division Multiple Access (SCFDMA) is used for uplink transmission.
- ***Inter-Cell Mitigation***: In LTE network, inter-cell interference mitigation techniques are used to cancel, randomize and co-ordinate interference through receiver processing, frequency hopping and resource usage limitation respectively.

- ***Power Control***: To improve system capacity, coverage, increase data rates and optimize power usage in LTE networks power control mechanism is used. SC-FDMA is used for setting power levels in uplink transmission.
- ***Rate Policing***: For better QoS rate policing mechanism is applied on each bearer during uplink and downlink transmission. The MBR provides upper limit for GBR bearer and AMBR provides upper limit for non-GBR bearer.
- ***Pre-emption Handling***: The pre-emption mechanism is applied to avoid congestion due to network overload by allowing higher priority process to pre-empt resources from lower priority process.

2.4.4 Challenges in 3GPP Architecture

Current 3GPP architecture faces significant challenge in signalling load as compares to its previous standards like 2G/3G/HSPA. Due to the user demands the LTE system must deal with a massive signalling requirement hike per subscriber. The GTP-C interface efficient management is essential to avoid loss of connectivity to the PDN-GW and IP services, maintain ability to set and release bearer during network overload and transfer user information to PCRF. However, in the 3GPP architecture the interface calls for an update to support dynamic load balancing and congestion prevention in the mobile network. Therefore, an efficient approach to mitigate overloading on GTP-C needs to be introduced in the EPC.

Domain Name System (DNS) weights are used by the GTP-C for load balancing mechanism, where the weights are set according to the capacity of a GW node relative to another GW node. Based on DNS weights, when a failed GW recovers, the traffic is balanced among various GWs even if the recovered GW is not loaded or the other GWs can handle more load. The load balancing fails to distribute traffic among GWs when a SGW fails fully or partially. Hence, it is safe to comment that the GTP-C protocol is not efficient to handle traffic load explosion or congestion issues in both user and control plane. In [70], the GTP- C protocol drawbacks along with possible ways to mitigate them have been discussed. The proposed solutions demand major modifications in the EPC and updating the interfaces, which directly impacts the network operations and cost.

The tunnel establishment during initial attachment procedure is another significant drawback in the 3GPP architecture. Even when there is no data to be transmitted the TEID values are allocated by each GW node, thus TEID values get exchanged for each GW nodes relocation affecting the network elasticity.

2.4.4.1 Network Congestion Scenario

In [70], various mobility and session management scenarios are defined as mentioned below, which affect the GTP-C and cause overload or congestion in the mobile network.

- Idle to Connected and vice versa transitions: Session overloading may occur in a single SGW or set of SGWs managing TAs based on the value of eNodeB idle timers, as result of enormous number of SERVICE REQUESTS from UE during peak hours.
- Overload in the PGW affect the SGW capacity due to GTP-C signalling retransmissions.
- Single EPC node (e.g. SGW) failure invokes increase in GTP-C signalling as the network attempts to re-establish the GTP-C session through a new EPC node (new SGW). The

increase signalling to restore PDN connection will cause an overflow in other EPC nodes and transform into a network issue.

- A non- overloaded GTP-C interface traffic may not be handled rightly by the GTP-C node when another of its interfaces is overloaded.
- Large amount of create and update bearer request are sent from PGW to MME when many users start application related interactions concurrently in case of special social or cultural events.

2.4.4.2 Proposed solutions

The authors in [70] proposed viable solutions for the issues mentioned before, such as load balancing by introducing gateway load or overload information during SGW/PGW selection. Some of the solutions are illustrated below:

1. ***Using load / overload information for dynamic load balancing of gateway nodes:*** PGW shares it load information to MME/SGSN through SGW over current signalling messages like create or session request. The load information details the resource status in the originating GTP-C node and has a range of value between 0-100 depending on the load of the node. The information is then forwarded in the same fashion from SGW to MME/SGSN. However, there is no standardised method in which the node will compute the load information neither a standard format of the load information is not defined. Hence, the accuracy of load information which will be synchronised with the DNS weights is questionable.
   On the other hand, the overload information reflects the originating node information when it functions beyond its capacity to handle traffic. This information forwarding is difficult as the overload action taken for different GTP-C nodes vary. For example, when the MME address gets changed and the overload information is transmitted from MME/SGSN towards SGW/PGW, the PGW which is unware of the address change acts on the overload information of the old MME. Moreover, the process is reliant on peer node capacity, any overload information not supported by the peer node gets discarded.
2. ***Use of Gateway Load Manager for DNS based load balancing:*** The Gateway Load Manager (GLM) is located between MME/SGSN and DNS server and assists in the SGW/PGW selection procedure considering the current network load condition. The GLM which has the load status of the SGW/PGW receives the DNS query transmitted from MME/SGSN. In [70], couple of methods have been proposed for the GLM to be able to retrieve load information from the SGW/PGW: 1. Use protocol (e.g., Simple Management Protocol), 2. Use Operations, Administration, and Management (OAM). The authors in [70] highlight standardizing a new interface for multi-vendor environment as another possible option. All the proposed solutions to enable the GLM to retrieve the load information fail because the current load factor doesn't not truly imply the real-time network statistics. For example, in case of SGW/PGW overload situation the GLM cannot balance load in the network, since it cannot get any SGW/PGW address in response to DNS query sent from MME/SGSN. Moreover, the signaling between MME/SGSN and DNS rises due to delay introduced by changing SGW/PGW load information, which makes GLM a network failure point.

2.4.5 NFV/SDN based solutions

Moving ahead of the various proposed solutions for 3GPP drawbacks, NFV and SDN based solutions can provide more reliable and flexible long-term solutions for load-balancing and QoS mechanism challenges. Prior to introducing the NFV/SDN concepts to build a new solution framework for future demands it is essential to fully understand the characteristics and capabilities of these technologies and the interdependency between them.

2.5 NETWORK FUNCTION VIRTUALIZATION (NFV)

Mobile networks are dependent on their underlying hardware and presence of special hardware appliances, such as, routers, firewalls, and deep packet inspection (DPI) have increased the challenges faced by network operators. Also, CAPEX and OPEX figures rises because the lifecycle of these hardware's is reduced to keep up with the innovation trends. Therefore, the Information technology (IT) specialists with an aim to reduce the CAPEX and OPEX have shifted to the idea of virtualising underlying hardware in the network and run as a software application. Thus, the hardware is emulated, and OS runs over the emulated hardware in a virtual environment, which enables resource sharing among multiple virtual machines and attain better network efficiency and power optimization.

2.5.1 NFV Architecture

The NFV framework is composed of:

1. ***Physical Server***: Hardware machines consisting all physical resources.
2. ***Hypervisor***: The virtual machine monitor is the software responsible for managing all the physical resources and providing platform on which virtual machines are executed.
3. ***Guest Virtual Machine***: Software which emulates the architecture and its node functionalities where the desired applications are executed.

2.5.2 NFV Benefits and Use Cases

The telecommunication greatly benefits from the NFV technology, such as scalability, flexibility, increased network efficiency, reduced development cycles, transparency of platforms, and most importantly reduced CAPEX and OPEX investments.

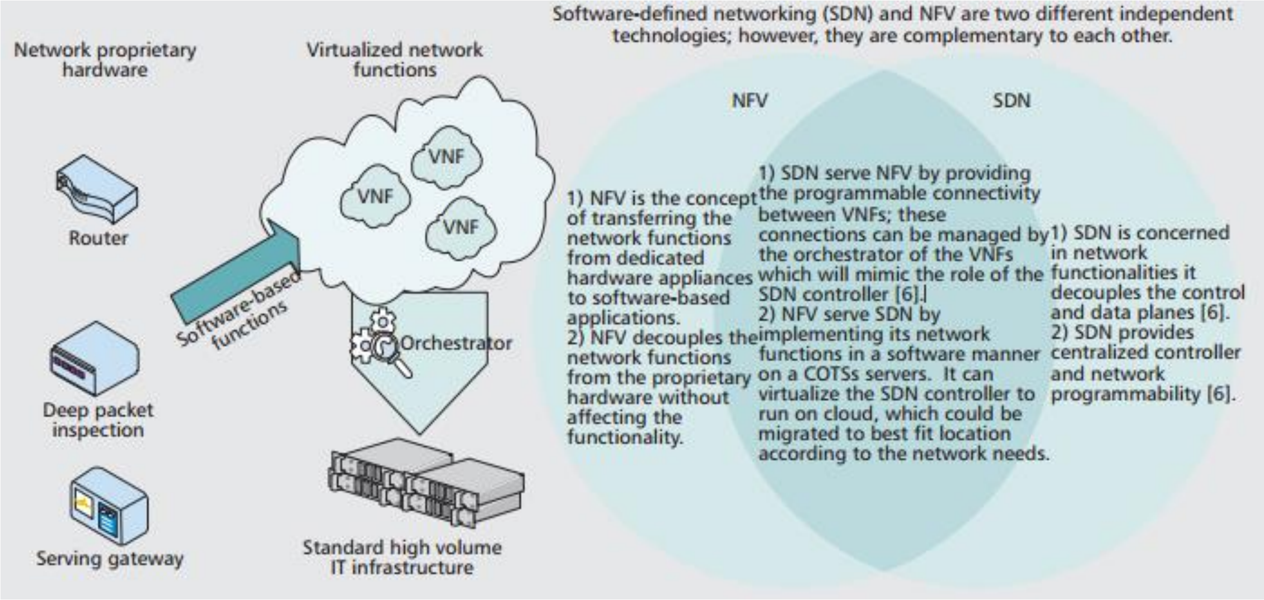
2.5.2.1 Mobile network virtualisation (BS and EPC)

The possibility of virtualising the BS has gained quite a momentum among service providers so that they can unite as many network functions in a single virtualized BS. A virtualised BS will enhance the network performance but the implementation of a virtualised BS is a complex and challenging procedure. The physical layer hosts the signal processing functions in 3GPP architecture so the virtualisation is implemented in higher network stack layers: Layer 3 followed by Layer 2. Virtualising these two layers of the BS permits the mobile network to have a centralized computing infrastructure for multiple BSs, as well as have low-cost BSs. Additionally, the network highly benefits in terms of coverage and minimum CAPEX and OPEX investments due to sharing of remote BS infrastructure.

On the other, the mobile core network is under intense pressure due to the rapid increase in connectivity demand. Therefore, the mobile core needs to evolve to a highly flexible, robust, and easily manageable network, which in real time can be scaled on-demand and managed. EPC network virtualisation provides the network operators with all these benefits functionalities because instantiation of VNF in the cloud enhances the network performance massively. Also, optimizing network configuration and topology in real time based on the actual traffic or mobility patterns and service demands can be a reality with NFV technology. NFV benefits also include the possibility of running production, test, and reference facilities on the same infrastructure [71].

2.5.3 NFV differs from SDN

As shown in figure 2.5-1, SDN and NFV are two different independent technologies; however, they are complementary to each other [72]. NFV and OpenFlow-enabled SDN proposed by the Open Networking Foundation (ONF) aims to achieve the same benefits. NFV technology targets optimisation of the deployment of network functions and SDN technology focuses on routing and optimizing the underlying networks. Although, the two technologies differ yet combing them network providers can achieve excellent value in the mobile network.



Software-defined networking (SDN) and NFV are two different independent technologies; however, they are complementary to each other.

1) NFV is the concept of transferring the network functions from dedicated hardware appliances to software-based applications.
2) NFV decouples the network functions from the proprietary hardware without affecting the functionality.

1) SDN serve NFV by providing the programmable connectivity between VNFs; these connections can be managed by the orchestrator of the VNFs which will mimic the role of the SDN controller [6].
2) NFV serve SDN by implementing its network functions in a software manner on a COTSs servers. It can virtualize the SDN controller to run on cloud, which could be migrated to best fit location according to the network needs.

1) SDN is concerned in network functionalities it decouples the control and data planes [6].
2) SDN provides centralized controller and network programmability [6].

Figure 2.5-1: NFV concept, NFV differs from SDN, source [71]

Table 4 shows the applicability of NFV and SDN in the OSI layer.

Applicability	OSI Layer
NFV	Application
NFV	Presentation
NFV	Session
NFV	Transport

SDN	Network
SDN	Data Link
	Physical

Table 4: NFV and SDN applicability

2.6 SOFTWARE DEFINED NETWORKING (SDN)

The essential enablers to realize the 5G network KPIs - programmability, flexibility (e.g., reusability, reconfigurability and resource sharing), compliancy and special features like mobile edge computing, self-optimisation, network slicing, automatic network management are software network technologies. The advantages of softwarization include OPEX and CAPEX reduction, service lifecycle management, optimise energy consumption and high QoE. This section provides more details on the 5G enabling software network technology: Software Defined Networks, as well as their impact on the network architecture.

2.6.1 SDN Architecture

SDN refers to the ability of software applications to program individual network devices dynamically and therefore control the behaviour of the network [74]. SDN helps delivering a programmable network design and operation management in a dynamic and scalable fashion. The forwarding and control plane is separated in SDN framework through abstraction between the legacy forwarding and control plane, which allows to control the network through programming. SDN aims to reduce complexity and assist in fast innovation at forwarding and control plane [75].

1. **Application layer and Northbound Application Programming Interface (API):** the applications excluding the ones which in any form support the operation of the forwarding plane (e.g. routing in the control plane) and services which define network behaviours are in this layer. The software interface between the controller platform and VNF applications running on the top layer is called the *Northbound Open API*. The open source community, customers and partners are working towards the development of these interfaces. OpenStack which is an open source provider aims to host NFV API into the SDN architecture.
2. **Control layer:** this layer is responsible for making routing decisions and pushing the decisions down the network devices for execution. The device forwarding plane is mostly the area of focus rather than the operational plane of the device. For few instances, the operation plane information such as the current state of a port or its capabilities is required by the control plane. The main function of the control plane is tune the flow tables based on the network topology and external service requests. In figure 2.6-1, OpenDaylight (ODL) controller is placed in the core of the architecture, which consist of the programmable modular structure. Note that any of the existing controllers can be used in the controller platform.
3. **Infrastructure layer:** the control communications between the controller platform and data plane devices (physical and virtual switches) is defined by the *Southbound Protocols.*

It considers the flexibility requirements in planning, and deploying SDN. OpenFlow (OF) is one the main protocols defined for the *Southbound interface,* however other protocols such as Border Gateway Protocol (BGP) and SNMP is also supported.

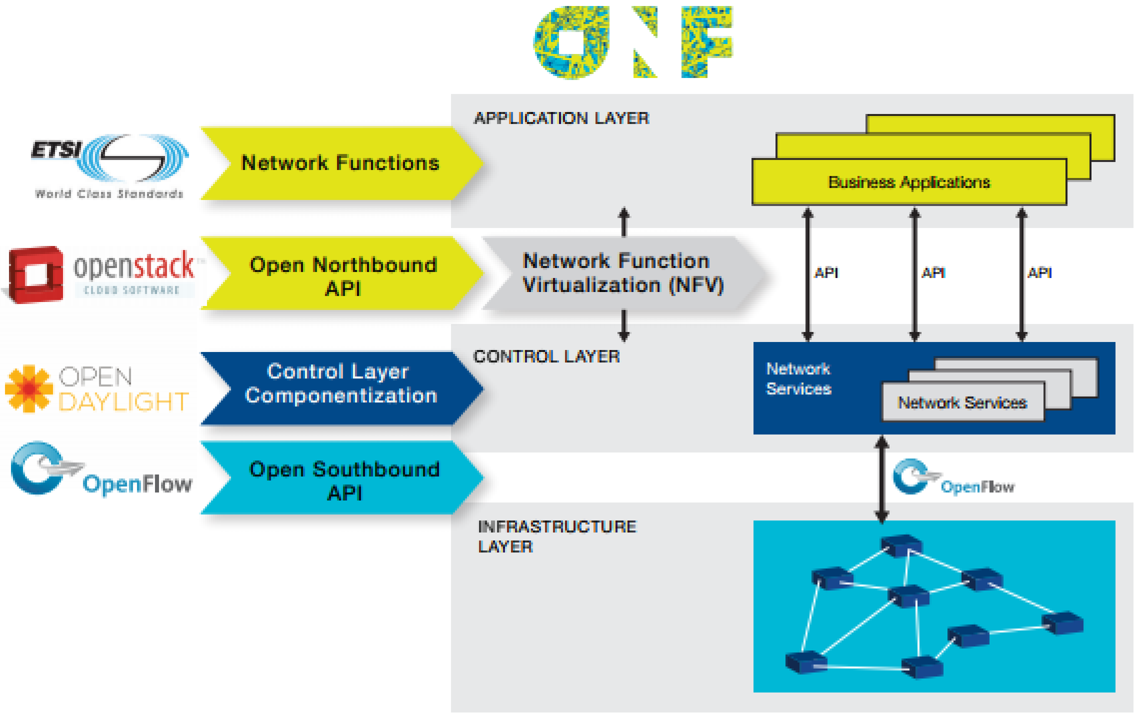


Figure 2.6-1: Open SDN Architecture, source [73]

2.6.1.1 Abstraction Layers

SDN structure is composed of four abstraction layers:

1. **Device and resource Abstraction Layer (DAL):** network devices forwarding and operational planes resources are abstracted to the control and management planes. Both forwarding and operation planes or any other plane of the device is abstracted either to the control or management plane in some cases.
2. **Control Abstraction Layer (CAL):** the applications and services of the control plane is separated from control plane southbound interface and DAL.
3. **Management Abstraction Layer (MAL):** splits the management plane southbound interface and DAL from the applications and services of the management plane.
4. **Network Services Abstraction Layer (NSAL):** the applications and other services receives service abstraction in this layer.

2.6.1.2 OpenFlow Protocol (OF)

Stanford University originally developed the OF framework which is currently under development through the Open Networking Foundation (ONF). This section reflects the OF version 1.4. OF is the protocol used by the SDN controller to add, update, delete entries and manage the flow tables either reactively or proactively. The OF uses three types of flow instantiation in response to the packets as mention below:

1. ***Reactive flow instantiation*:** OF lookup in the flow table for a match when a new flow comes into the switch and if no match is found for the flow, it created an OF PACKET_IN message which is forwarded to the controller for instructions. A rule is created in the flow table depending on the traffic and commands from the OF controller in reactive mode.
2. ***Proactive flow instantiation*:** OF doesn't not lookup in the flow table and pre-sets flows which helps in reducing latency due to contacting the controller for commands.
3. ***Hybrid flow instantiation*:** uses the best of reactive and proactive flow instantiation to gain flexibility in controlling granular traffic and maintaining low latency for remaining traffic.

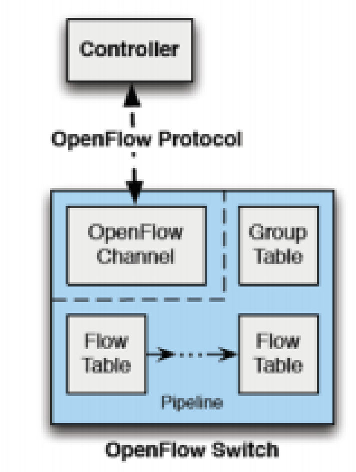


Figure 2.6-2: OF switch, source [9]

Every OF switch performs packet forwarding and lookups by maintaining one or more flow tables. The OF switch version (1.4) is composed of three main modules, as shown in Figure 2.6-2:

1. ***OpenFlow Channel***: acts as a medium to send commands and packets between the switch and the controller. OF protocol which runs over Secure Sockets Layer (SSL) is used by the SDN controller to manage the switch and that's why the term *Secure Channel*.
2. ***Flow Table**:* for each flow entry, there is an action listed in the table which guides the switch how to process the flow. The flow table has a set of flow entries namely match fields, instructions, and counters.
3. ***Group Table*:** the list of actions for each group entry is contained in the group table and these actions are applied to packets sent to the group entries [9].

2.6.1.3 OpenFlow Switch

The collection of flow tables which contain the rules, actions and counters and a group of tables model the OF switch as shown in Figure 2.6-2 and Figure 2.6-3. The flows are defined by header fields specified in the rules column and the incoming packets headers are matched with these rules. The counter in counter column gets updated, if a rule matches and actions from the rule column is applied to the packet. When multiple rules are matched for a packet then the rule with highest priority is applied. Every rule which is matched with the pack header has an exact match specification to be accounted for header fields or wild cards. The actions set by the rule column perform in one of these possible ways: forward the packet to an output port, modify the packet, or send the packet to the next table or to the group table. *Physical ports, logical ports* and *reserved ports* are the three different OF ports supported by OF switch [9].

- *Physical* ports: correspond to a hardware interface of the switch and they map one-to-one to the interfaces.
- *Logical* ports: do not correspond directly to a hardware interface of the switch and have an extra metadata field called Tunnel-ID in the packet associated with the port.
- *Reserved* ports: do not require a switch to support the port which are marked as ‘required’.

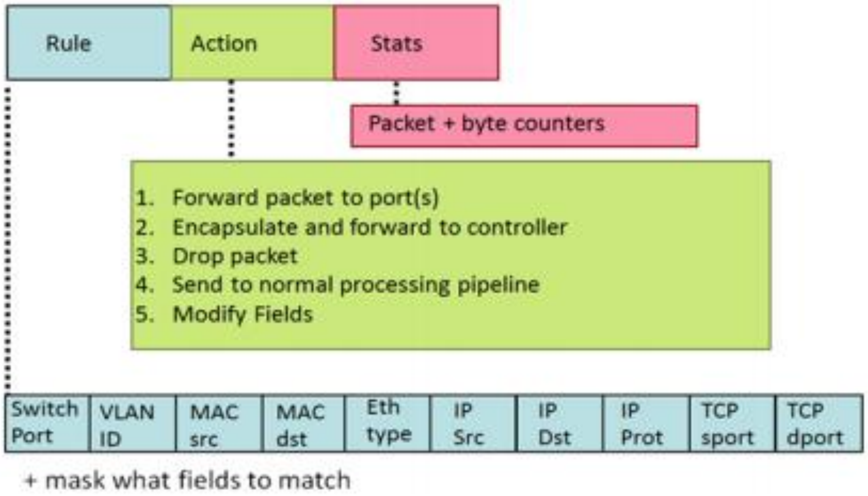


Figure 2.6-3: OF Flow table, source [76]

2.6.2 Open vSwitch

An Open vSwitch is modelled in a fashion so that it can be implemented as a virtual switch in a virtualized server environment. It is designed to perform traffic forwarding between different VMs on the same physical host, as well as between VMS and the physical network. All Linux based virtualised platform can host Open vSwitch such as VMware, Kernel-based Virtual Machine (KVM) and VirtualBox.

Open vSwitch supports standard management interfaces and protocols and several tunneling protocols while supporting distribution across multiple servers. Open vSwitch ports represent connections to physical interfaces and patch cables and vSwitch bridges represent network interfaces to Linux. All the ports in a bridge receive the packets from any given port on that bridge.

2.6.3 Use Cases

One of the notable developments in the carrier network is SDN, as it helps to realise the essential design aspects of 5G network: a) flexible spectrum management; b) network programmability; c) network slicing and d) improved performance in heterogenous and ultra-dense deployments.

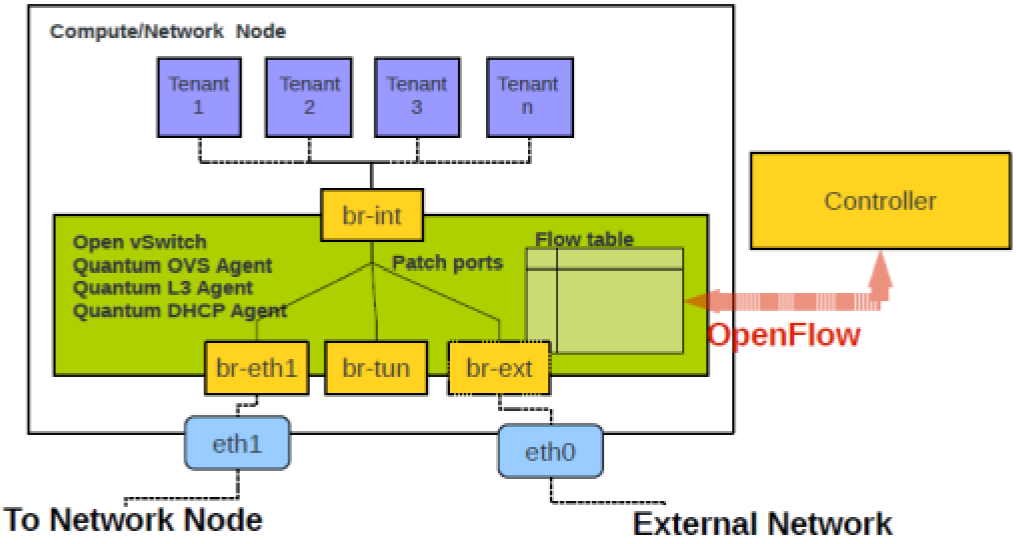


Figure 2.6-4: Open vSwitch, source [77]

2.6.3.1 SDN in Mobile Network

The greed for high performance infrastructure along with excellent management in mobile network (legacy network has distributed network entities across a geographic area) brings in challenges for network providers. Additionally, the transport facilities employed for backhaul vary depending on the network generation, hence it is advisable to migrate most of the packet processing functions to the network entities for better QoS and traffic management. SDN introduction as an external control plane or within the network will enhance the configurability and flexibility of the network. Further in this section how SDN realisation for mobile network can be achieved is discussed.

1. ***S1 Interface***: the end-to end link between cell site and EPC in mobile network knows as the S1 interface need re-configuring to support future virtual network designs and IoT services. Cell site optimisation can be possible by turning on small cells based on usage demands. Further, the S1 activation can be automated and S1 path to virtual gateways and small cells can be optimised depending on network performance, warnings and load statistics provided by the centralised SDN control plane. SDN realisation for MBH network enables autonomic network optimisation to ensure optimal routing across MBH network.
2. ***Resource Management for Improved Network Utilization*:** the current mobile networks are provisioned with excess capacity in protection paths to withstand network outages. However, majority of the time the network remains underutilised as the traffic is limited to 50-70% of the total link capacity to maintain QoE. SDN controller deployment provides the mobile network with a potent control policy which routes traffic bursts in real time to alternate paths based on network load information. Besides SDN also allows the protection paths to be used for unpredicted demand during normal time for better resource utilisation. The centralised Path Computation Element (PCE) enables advanced load balancing and versatile utilisation of primary and protection paths in the network

3. ***Network Reliability and Performance*** – future mobile networks are expected to deliver best-effort data transmission for mobile devices along with support for differentiated low latency services. SDN technology integration in the mobile network permits fine-tuned service options using real time network data retrieved via *Southbound APIs*. On the other hand, *Northbound APIs* functions to optimise entire network performance, coordinate efficiently with the RAN management. *Northbound APIs* correlate end user actions in both RAN and backhaul network to achieve optimal routing, QoS and resource allocation.
4. ***Support Virtual Gateways*** – traffic in future mobile networks will not be directed to a central PDN-GW handoff point through different controllers and GWs, instead on demand virtual servers will become handoff points. SDN based network assists the access and aggregation network to adapt to virtual resources to provide seamless connectivity via fast VNF activations. Further, SDN API calls validate the re-directing of end user traffic to new virtual resources or available capacity for load balancing.
5. ***Self-Organizing, Multi-Layer Networking* –** traffic bursts are presently managed by costly underutilised access networks without consideration for delay or data loss, thereby, reducing user QoE. SDN controlled control plane will be able to forward traffic flows based on network performance and substitute path availability. Multi-layer SDN provides the path information between optical and IP layers for optimal IP routing based on current optical network status.

CHAPTER 3

3 PROPOSED SYSTEM DESIGN

3.1 SOFTWARE DEFINED VIRTUALISED CLOUD RADIO ACCESS NETWORK (SD-vCRAN) AND PROGRAMMABLE EPC FOR 5G

3.1.1 Overview

The concepts and technologies discussed above and the research work published in recent years highlight the considerable challenges in deploying 5G network. The ever growing signalling and strict QoS requirements are placing pressure on current network architecture that was not planned to support the new services. Therefore, to reduce the strain on the backhaul and core network or reconfigure it to support these demands, SDN concept is adopted in the network. The SDN integration in the backhaul and core network involves a complex procedure where the 3GPP standardised protocols are critically assessed. The major challenge is to find a way to preserve the LTE functionalities in the new SDN based backhaul and core network architecture. This research focuses on the use of a novel software defined virtualised CRAN architecture to reduce signalling to the core. This chapter presents a new programmable/configurable backhaul and core network where the main procedures for both control and data planes are based on SDN and NFV technology. Firstly, the challenges associated with the proposed architecture and solutions for the same are presented. Secondly, the proposed architecture is analysed based on flexibility, complexity and performance. The benefits and applications of the proposed system design is discussed.

3.1.2 System Functionality

The proposed architecture will focus chiefly on changing the existing backhaul and core network architecture to integrate the traditional core network entities into the OF network. The control plane and data plane functions within the MME, SGW and PGW are separated, while *S1-MME, S1-U, S6a and Gx* interfaces remain unchanged. The MME functionalities: UE authentication, authorisation and intra-3GPP mobility management are preserved. The control and data plane separation enables the creation of intelligent network components: S/PGW-Controllers (S/PGW-C, where C stands for control plane) which can be hosted as a physical machine or virtual machine in the cloud. The S-GW-data plane (SGW-D) forward traffic flows based on the instructions received from the centralised SDN controller in the core network. The SGW-D and PGW-data plane (PGW-D) entities function as OF switches: OFSW1 and OFSW2 respectively. The SGW/PGW selection mechanism based on DNS is altered. The MME request the OF controller via *Northbound Interface* to install forwarding rules in the OF switches. The control gateway manages the GW-D entities using *Southbound API* (Open Flow) protocol, whereby the instructions are loaded onto OFSW1 and OFSW2 to forward the UE traffic to the destined PDNs. The functionality of the main network entities is presented as follows:

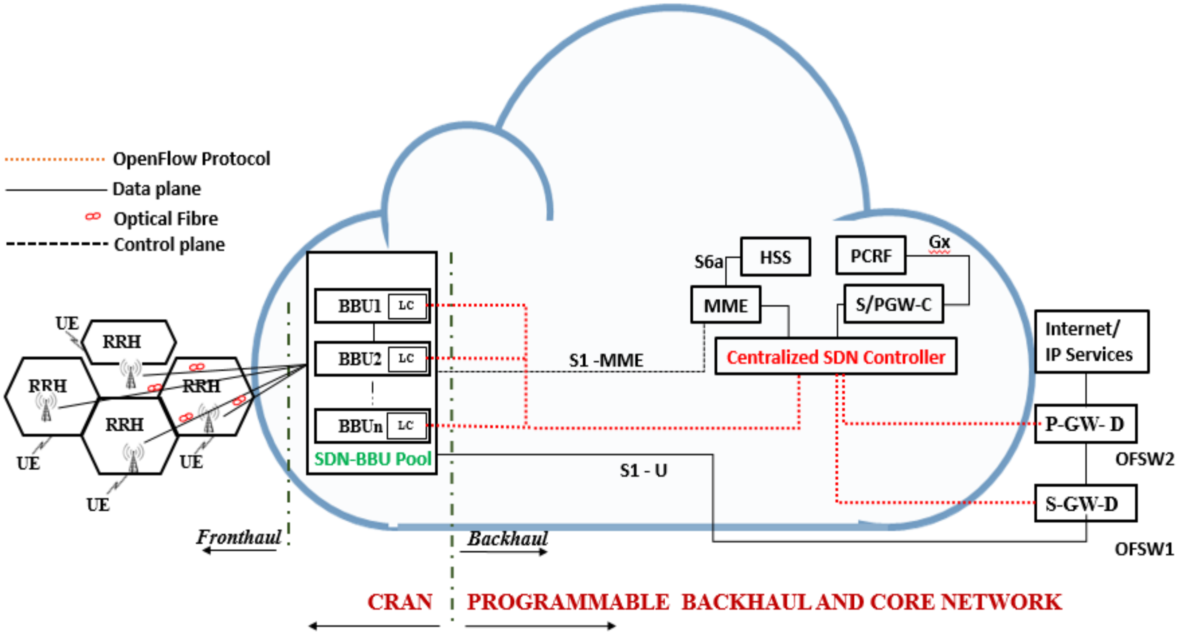


Figure 3.1-1: SD-vCRAN and Programmable EPC architecture for 5G

1. ***SDN-BBU Pool***: The CRAN BBU pool is shifted to the programmable core network which has a SDN controller responsible for load balancing, traffic management and QoS. The controller will route data traffic and any required signalling on to the EPC over the existing 3GPP standardised interfaces. The BBUs can be hosted both ways: OF controlled access switches (SDN-BBU) or standalone baseband processing units (BBU). SDN-BBUs process flow entries received from the controller in their flow tables and forward the user traffic between BBUs and IP services. The BBU preserves the radio functions for the radio interface.
2. ***Local Controller(LC)***: functionalities include packet classification at the SDN-BBU depending on the Location IP (LocIP) which consists the UE IP address and BS prefix. The OF protocol connects the LC with the SDN controller.
3. ***MME***: is a virtualised EPC component which connects with the SDN controller via *Northbound Interface* using OF protocol. Unlike traditional EPC, the new MME function doesn't include S/PGW-D selection. The network operators can choose to use the MME as a standalone component or integrate it into the cloud, thereby adding more elasticity in the core network. Virtualised MME, PCRF and S/PGW controllers can run on top of the controller as VMs and help preserve the 3GPP standardised radio interfaces like S1-MME connecting BBU and MME, S6a connecting MME and HSS, and G*x* connecting PCRF and PGW.
4. ***SGW-C and PGW*-C**: the control functionalities like TEID (Tunnel Endpoint Identidy) assignments are abstracted to the top of controller. The UE IP address after receiving

MME query gets stored in the database, also the database stores the bearer QoS parameters transmitted from PCRF to the SGW-C and the users account. The control plane and GTP-C is preserved to simply the functionalities of the 3GPP standard interfaces.

5. ***SGW-D and PGW*-D**: OF switches will are controlled by the SDN controller via OF protocol and forwards user traffic flows between the BBU and IP services
6. ***SDN Controller***: the most essential component of the architecture accountable for the data plane management and making traffic routing decisions in data plane (between BBU pool and S/PGW-D). It has three major functionalities: 1. maintain current port information, 2. load balancing, and 3. traffic flow routing. The load is balanced in the network by gaining network information (transmitted bytes) periodically from the OF switches. The traffic is routed for a default bearer depending on the bandwidth limitation, UE requirements and commands received in the SDN-BBU pools data plane and S/PGW-D switches. The rules can be installed either reactively or proactively in the OpenFlow switches depending on the classification of bearer. During default bearer establishment, the rules are enforced in the OpenFlow switches flow table in advance (proactive mode) however, for a dedicated bearer the flow tables in the OpenFlow Switches are loaded with rules based on the packet header information (reactive mode).

3.1.3 Proposed System Control Plane

The C-Plane encompassed all 5G network controlling components as presented below:

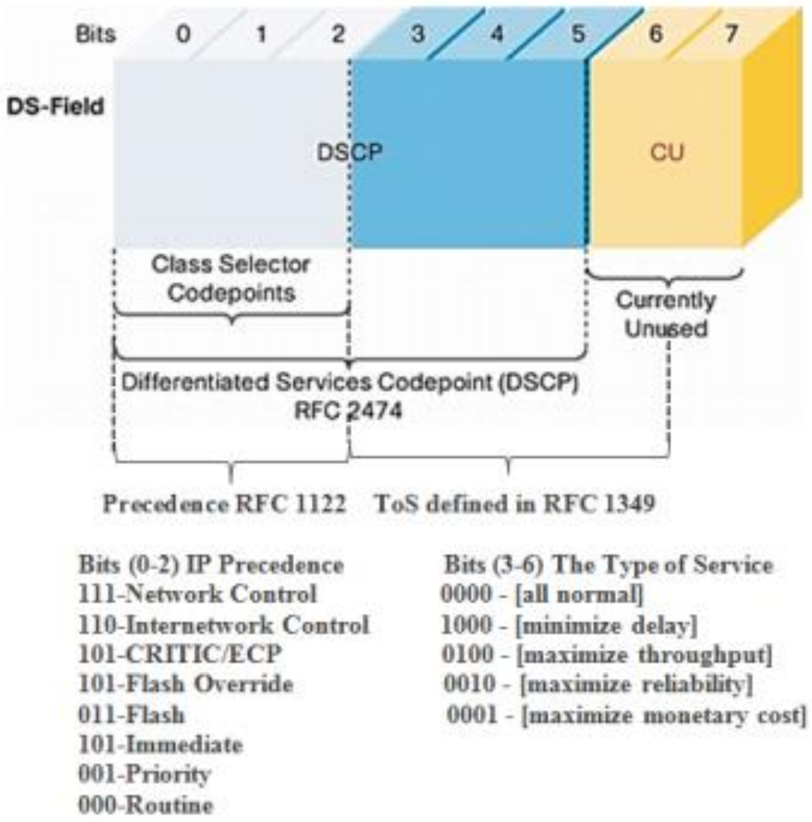
3.1.3.1 Local Controller (LC)

The LC functions are listed below:

- Packet classification of flows and forwarding them to the SDN controller via OF protocol
- Enforce wildcard matching rules during dedicated bearer establishment
- Using LocIP identify users connected to the same BBU
- Handle mobility during user attachment to new BBU
- Reduce the resource processing pressure in the SDN controller. The SDN-BBUs can maintain per-flow state but to improve the core network scalability the flow aggregation is performed.
- Flows with same source IP and destination port are aggregated into a single bundle and routed in same way.
- Equally distribute traffic flows and avoid network blockage at edge switches

3.1.3.2 SDN Controller

The function of the SDN controller is to categorise traffic into different classes and implement QoS parameters to those classes depending on one of these parameters: source and destination IP address, port numbers and protocol in Layer 4 (transport layer), type of service byte value and Layer 2 information. The processed packets based on the above parameters are conditioned and marked. The type of service field is replaced by Different Service (DiffServ) field in the IPv4 to make decisions regarding the classification of packets and traffic monitoring based on Per Hop Behaviour (PHB). The processed packets are categorised into classes based on the type of service bytes in the IP header, Figure 3.1-2, presents the IPv4 type of service as defined by the authors in [78], where the IPv4 type of service field is formed by a four-bit field. In RFC2474 the type of service byte has additional bits (two zero bits) at the end of Differentiated Service Code Point (DSCP) [79].



Bits (0-2) IP Precedence
111-Network Control
110-Internetwork Control
101-CRITIC/ECP
101-Flash Override
011-Flash
101-Immediate
001-Priority
000-Routine

Bits (3-6) The Type of Service
0000 - [all normal]
1000 - [minimize delay]
0100 - [maximize throughput]
0010 - [maximize reliability]
0001 - [maximize monetary cost]

Figure 3.1-2: IPv4 type of Service field, source [80]

3.1.3.3 Classification of Class

The Quality of Service Indicator (QCI) discussed in section 2.3.4.1 is directly mapped to the Differentiated Service Code Point field defined in [79]. According to 3GPP standards, the DiffServ has three classes: default or best effort (BE), assured forwarding (AF) and expedited forwarding (EF). Processed packets can be marked based on the appropriate class type (BE, AF, or EF) with an arbitrary DSCP value. Table 4, presents the mapping between QCI and DSCP.

Class	**Standard**	**DSCP**	**QCI**	**ARP**	**QoS Level**	**Traffic Category**
BE	RFC 2474	0	8,9	1-15	Lowest	Background
AF21	RFC 2597	10	6	1-15	Medium	Interactive
AF22	RFC 2597	12	6	1-15	Medium	Interactive
AF23	RFC 2597	13	6	1-15	Medium	Interactive

AF41	RFC 2597	34	2	1-15	High	Live
AF42	RFC 2597	36	2	1-15	High	Live
AF43	RFC 2597	38	2	1-15	High	Live
EF	RFC 3246	46	1	1-15	Highest/Premium	Conversational

Table 5: Mapping of DSCP to QCI in 3GPP standards, source [78]

1. ***Best Effort (*****BE)**: is the default class as per the PHB specifications in [79]. BE services are assigned to packets marked with DSCP value of ‘000000’ through a DS-complaint node that complies to all the network core DiffServ requirement. For instance, a BE service can be related to a Hypertext Protocol (HTTP) session sent from port 80, to a security system that stops all incoming traffic apart from the traffic sent from port 80. The BE traffic having same destination port ‘80’ can be sent from different BBUs with non-alike traffic tags, which is why it is necessary to optimise the resource utilisation of the GWs. Therefore, BE services need to allow users to have a fair chance to share existing bandwidth.
2. ***Assured Forwarding (AF)***: AF services assures a minimum number of resources to each AF class. AF PHB divides traffic into four classes as specified in [81], further, the packets within each class are divided based on one of the drop preference categories. During network cramming in a DiffServ node on a particular link, the AFx class packets will be dropped. The AFxv class packets will be dropped such that dropping probability (dp) of AFx1 is: dpAFx1 ≤ dpAFx2 ≤ dpAFx3. The DSCP ‘xyzab0’, where ‘xyz’ is 001/010/011/100 and ‘ab’ is drop precedence bits, can be used to denote AFx class. The level of assurance provided by AF services is reliant on the amount of resources reserved for its class, current load of its class, and drop precedence bits.
3. ***Expedited Forwarding (EF*****)**: As defined in [82] EF PHB is the most vital DiffServ to provide low latency, low datagram loss, low jitter, and assured bandwidth services. Application like VoIP and video need premium services to meet the QoS requirements such as low packet loss, guaranteed bandwidth and low packet delay variation. Priority queuing can be used to implement EF along with a rate capping on the class. In a DiffServ network, EF provides premium service which is specifically targeted towards the highest priority application, since most or all traffic can be treated with highest priority in an event of network congestion. The DSCP value for EF is ‘101110’ (46) as shown in Table 4, where the type of service bytes uses all the 8 bits and DSCP use the first 6 digits.

3.1.4 Proposed System Data Plane

3.1.4.1 Packet Date Network Gateway Data (PGW-D)

The PGW-D translates the UE address without querying the SDN controller by caching the packet state sent to the IP services and performing Network Address Translation (NAT) (see Figure 3.1-3). The PGW-D use a large table to cache the related user state, however, for the optimised processing capabilities the OFSW2 doesn’t not necessarily require costly Ternary Content-Addressable Memory (TCAM). The firewalls can be configured or the access list can be configured instead to block malicious attacks from the internet.

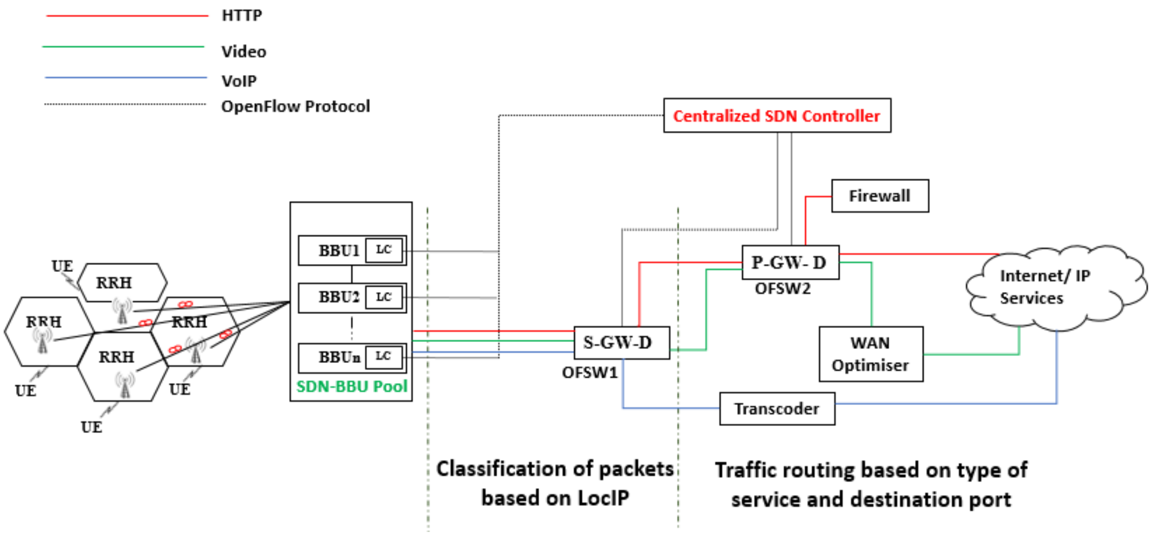


Figure 3.1-3: Classification of packets and routing in proposed system

3.1.4.2 Middleboxes

The system needs middleboxes such as firewalls, transcoders, Wide Area Network (WAN) optimisers, etc., between OFSW2 and IP services to enforce the extra rules to different traffic (e.g. HTTP, video, VoIP) flows. The middleboxes can be hosted on VMs or deployed as physical machine in multiple locations.

3.2 DESIGN CHALLENGES AND PROPOSED SOLUTIONS

The proposed system based on SDN technology has four main challenges as described below. In this section, the design challenges and the possible solutions are presented.

3.2.1 Protocol Support Challenge

1. ***GPRS Tunneling Protocol (GTP) Encapsulation***: GTP-U is identified by pair of TEIDs corresponding to the source and destination nodes, along with the source and destination IP addresses, and User Datagram Protocol (UDP) port numbers. Legacy EPC architecture identifies each bearer by a tunnel and corresponding to multiple bearers a single UE can have multiple sessions. The SDN based proposed system performs entire traffic processing on a OF switch platform, which is controlled via OF protocol. The challenge is OF protocol (1.5.0) version used in this research does not support GTP matching and TEID in GTP header.

   Solution: Couple of solutions are presented for this problem: 1. Extend the OF protocol so that it supports both GTP matching and TEIDs, and 2. Add new entity to perform GTP matching and preserve the original OF protocol. The new entity will act like a decapsulator to remove the TEIDs with an enable easy network management between BBUs and IP core by the SDN controller, as shown in Figure 3.2-1.

2. ***GPRS Tunneling Protocol (GTP) Decapsulation***: In GTP stack the Ethernet header is absent after the UDP header. Therefore, to add GTP protocol in the Open vSwitch after GTP decapsulation the Layer 2 information needs to be retrieved. Hence, the Layer 2 information addition must be considered while extending Open vSwitch to support GTP protocol.
   Solution: A potential solution to this challenge is to make an Ethernet header which will allow the packets to forwarded based on Layer 3 information, as Layer 2 information is not available post decapsulation. possible solution to this problem is to create a simple Ethernet header and the packets will be forwarded based on Layer 3 information, since the Layer 2 information does not exist. A secondary choice may be to route the packets based on information available (input and output ports) and fully utilise the Open vSwitch capabilities. The performance of later solution is better than former solution in terms of throughput. Based on the comparison carried in [83] among these solutions, the current proposed system uses the first solution without implementation of GTP in the Open vSwitch as shown in Figure 3.2-1.

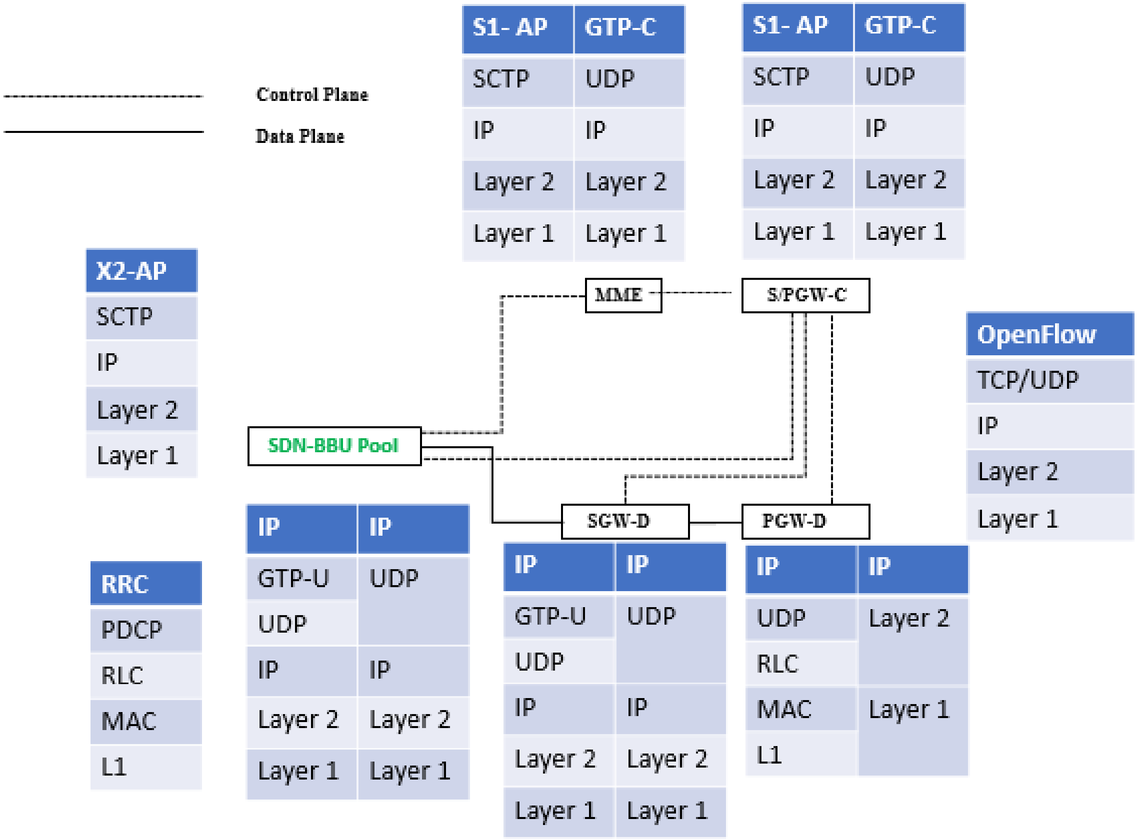


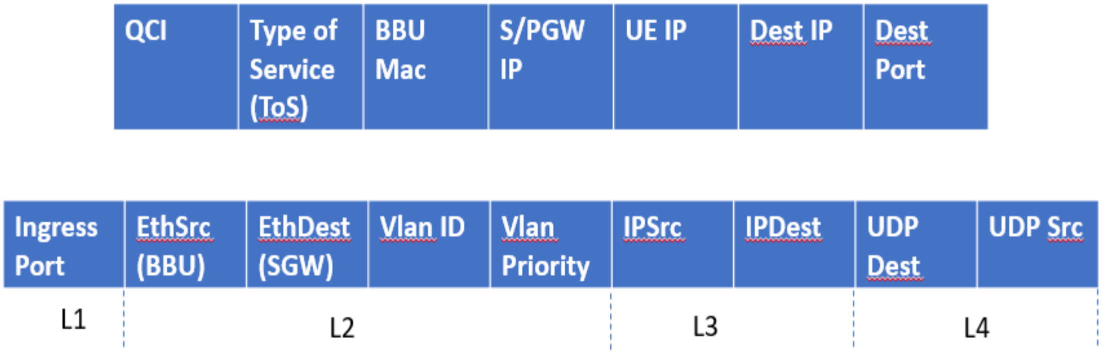
Figure 3.2-1: Proposed system protocol stack

3.2.2 Routing Scheme Challenge

1. ***Bearers***: A set of bearers based on their QoS provide special treatment to a set traffic in the legacy LTE network. The bearers could be either default or dedicated bearer

respectively. When the UE initially attaches to the network, the connection is maintained throughout the connection lifecycle and the default bearer provides BE services. A dedicated bearer acts as extra bearer which provide different treatment to different traffic sessions. These two concepts need to be defined in the proposed system

Solution: 1. **Default Bearer***:* The SDN-BBU access switches and S/PGW-D IP addresses and TEIDS are returned by SDN controller for controlling purposes. For example, depending on the network load information and load balancing mechanism, the SDN controller gets updated about the switches which have more capacity and direct the flow entries to them through data path. Additionally, the default bearer doesn't depend on the SDN controller because the rules are preloaded in the switches. 2. **Dedicated Bearers*:*** The LC in SDN-BBU pool based on LocIP combines the flows into bundles. The processing time and number of flow entries to the SGW-D is reduced by applying wildcard matching. At this stage, the user sessions are grouped in one flow with a unique policy tag or bundle ID, therefore the number of users is proportional to the number of bundles. In the following step, a policy path is established when the SDN controller lookup in its database table for the session QCI value and type of service bit after receiving PACKET_IN message from the SGW-D switch. The traffic can either be routed to the middleboxes or employed one of the service class defined in section 3.1.3.3 along with transmission rate capping. The selection of routing path, installation of policies on the paths and configuration of flow entries in the OF switches is done by the SDN controller. The MME sends *Attach Request* message to the controller once the UE is authenticated and the controller database is loaded with information such as: UE IP address, QCI value corresponding to the user session and form of subscription. The port statistics are periodically updated in the SDN controller by interrogating the LC and S/PGW-D tables as shown in Figure 3.2-2.



QCI	Type of Service (ToS)	BBU Mac	S/PGW IP	UE IP	Dest IP	Dest Port

Ingress Port	EthSrc (BBU)	EthDest (SGW)	Vlan ID	Vlan Priority	IPSrc	IPDest	UDP Dest	UDP Src
L1	L2				L3		L4	

Figure 3.2-2: SDN controller table and SDN-BBU access switch flow table

3.2.3 Flow Aggregation Challenge

1. ***Access Edge***: The access network has to serve millions of users which are attached to the BBUs, and the BBUs are further connected to many GWs. The responsibility to process massive user service requests puts a strain on the access network, hence fine

grained and intelligent polices needs to be installed on the service chain. The access network edge has lower bandwidth than the GW edge, hence the mobile network is characterised by an asymmetric edge.

Solution: To efficiently utilise resources and maintain an optimal processing power consumption, proposed system uses flow granularity to combine OF entries into bundles instead of individual OF flow entry routing. This can be done at the access edge (BBU pool) via flow aggregation, followed by rule application at the GWs for policy path implementation towards the Internet. The classification of the packets is performed at the access edge on traffic arrival and the results are encoded in the packet header. The traffic arriving at the GW edge from the Internet is simply forwarded as it has the classification results stored in the packet header.

3.2.4 Policy Paths Challenge

1. ***Quality of Service (QoS)***: An end-to-end QoS method is needed to handle the ever-growing real-time applications, as well as assure perform traffic treatment based on user profile and sessions. The above is possible by traffic shaping along with policy installation on the path.

   Solution: To add tailor made characteristic to enhance QoS support in the network, the policies can applied be to both individual flows and special application or service bundles. For example, in case of a live streaming/video session, the traffic must benefit from low jitter paths, which can be provided by the proposed system via a dynamic circuit created between source and destination packet switches. The circuit path will have the lowest propagation to maintain the QoS requirement (see Figure 3.1-3). Now for a HTTP traffic which doesn't have any stringent QoS obligations, the traffic is routed through a security system to the IP services. In the same manner, all VoIP traffic need to be routed over the path with shortest propagation delay in the employed topology as shown in Figure 3.6-1. The SDN controller is provided a QoS module which performs functions such as: class of services and DSCP value storage, policy application on paths which will be based on service class or queuing technique (applied to the ques attached to ports on an OF Switch). The module will also perform flow insertion and deletion, packet classification and matching. This module can modify and enqueue the network type of service bits and within the OF. Two types of policies are defined in this module: 1. Queuing policy, and 2. DSCP policy. The former mentioned policies utilise the enqueue action in OF switch to enqueue DiffServ traffic flows in the network and the flows are sent to a queue on an OF switch port, which has actions prelisted in advance. The network administrator handles the configuration of the queues in the OF switches using a OF-config which permits one or more OF D-plane to be realised [84].

3.3 MOBILITY

This section will elaborate the mobility aspects which is considered during proposed system design. The two aspects are: 1. User States – Idle or Connected States, and 2. Call Flows – UE triggered request and Network Triggered Request.

3.3.1 User States

In the EPC, the user is always provided with a default bearer since its initial attachment to the network for as long as the UE stays connected to the network. The IP address allocated to the UE when it gets attached to the network will stay with the UE for as long as the connection exists. The Default bearer is the first and last bearer to be established and realised during the UE connection to the network. Hence, the information corresponding to user applications specifically using IP connection can be transmitted to the IP core and returned to UE at any time. The above scenario defines the UE registered state.

On the other hand, in the core network the Default EPS bearer resources and IP addresses are assigned to a specific UE which is registered and for the radio network the resources are allocated and releases dynamically. The radio network approach is aimed to support the default bearer during transition between two modes (i.e., idle and connected mode) of user registered state.

The signalling between UE and MME during *EPS Connection Management (ECM)- Connected* state is composed of a Radio Resource Control (RRC) connection and S1-MME connection. The RRC connection exists between UE and BBU pool and the S1-MME interface connects the BBU pool and MME. A *Service Request* message is transmitted to MME from UE when it has new traffic to send or finds that the network has intentions to send new traffic. The *Service Request* message is transmitted from MME to *ECM/ RRC-Connected* state. Further, during S1 connection failure due to user inactivity (ECM-IDLE state) the UE has no access to any radio resources even though it is still registered to the network. Therefore, the UE is in *EPS Mobility Management (EMM)-REGISTERED*, while being in *ECM-IDLE* state.

3.3.2 Service Requests

In the proposed system service requests can either be user triggered or network triggered depending on the traffic origin. In the first one the uplink data is sent from UE to the network and in the later the downlink data is sent from network to the UE.

3.3.2.1 Service Triggered Requests

In *Idle* mode, the UE executes *Service Triggered Request* when it needs bearer establishment for data transmission purpose or transmit to MME. The initial user attachment procedure (refer to Figure 3.3-2) and default bearer establishment procedure (refer to Figure 3.3-1) is analysed as follows:

Step 1: The S1-MME and S11 interfaces are preserved in the proposed architecture, which means the Initial Attach Procedure is complaint to the 3GPP architecture. The UE sends *Attach* Request message to the BBU to initiate the *Attachment Procedure.* The message consists International Mobile Subscriber Identity (IMSI), previous Tracking Area Identity (TAI) and attack type. MME receives the *Attach Request* message which is embedded in the initial UE message through the S1AP interface. The *Attach Request* message contains other information and messages such as the TAI, PDN Connectivity Request, E-UTRAN Cell Global Identifier (ECGI).

Step 2: After the BBU identifies the UE, the MME authenticates the UE using its profile in the HSS. The UE subscription data information: IMSI, Access Point Name for user, and subscribed QoS is retrieved by the MME from HSS. UE authentication is performed based on the security vector parameters. When the MME changes after previous UE detach, an Update Location Request (ULR) is sent from the MME to the HSS, which sends a Cancel Location Request to the old MME. Further, the HSS acknowledges the message by sending an Update Location Answer (ULA) to the MME. MME performs UE authentication based on the security vector parameters.

Step 3: In standard 3GPP architecture, the DNS is inquired by the MME for S/PGW selection but in proposed system a Create Session Request is sent from MME to SDN controller over an Open API (OF protocol).

Step 4: The QoS for the initiated user session and maximum uplink and downlink download rate authorised at the OF level is obtained by the S/PGW-C application via PCRF query.

Step 5: In this step the TEID for control, OF switches (with less load) IP address, uplink and downlink download maximum rates are sent to the SDN controller from the S/PGW-C application. Also, in this step a *Create Session Response* message is initiated between the MME and centralised controller.

Step 6: At this stage, the SDN controller configures flows entries in the access (SDN-BBU) switches by communicating with the LC. Further, the controller installs policy rules in the data plane switches (OFSW1 and OFSW2).

Step 7: In this step, the switches send a response to the LC which further communicate with the SDN controller. The OF reply message PACKET_IN is containing the post statistics. The proposed system improves network utilisation by implementing the current step and the previous one (Step 6) in the network to reduce processing time taken in packet forwarding.

Step 8: The IP addresses of data plane OF switches (S/PGW-D), downlink maximum rates, uplink maximum rates are embedded in the *Attach Accept* message sent by MME to BBU. The UE obtains the Non-Access Stratum (NAS) messages forwarded from the BBU. Following this, the default bearer is activated by sending the *RRC Connection Reconfiguration* message in the next step.

Step 9: The *RRC Connection Reconfiguration* message is sent to activate the default radio bearer. The *RRC Connection Reconfiguration* message has *Attach Complete* message embedded in it because NAS Payload and UE communicates with the MME via SDN-BBU.

Step 10: In this step, the UE data packet is sent for first time in the uplink direction towards the Internet from the SDN-BBU. The packet is sent only when the MME receive the *Attach Complete* message.

The *default bearer establishment* is completed *in* steps 1-10.

Step 11: The LC checks the packet for the user source and destination IPs, BBU MAC address and updates the LocIP address in the packet header. Flows are not aggregated for a *default bearer establishment* or the policy rules on the path since Non-GBR is provided by the *default* *bearer*. Further, the flows are combined based on by the LC and forwarded to the data plane switches in a *PACKET_IN Request* message.

Step 12: The type of service bytes, user QoS level and QCI for the particular user session are mapped by the SDN controller after examining the destination port of the user packet. Based on session type, the policy path and flow entries are configured in the data plane switches.

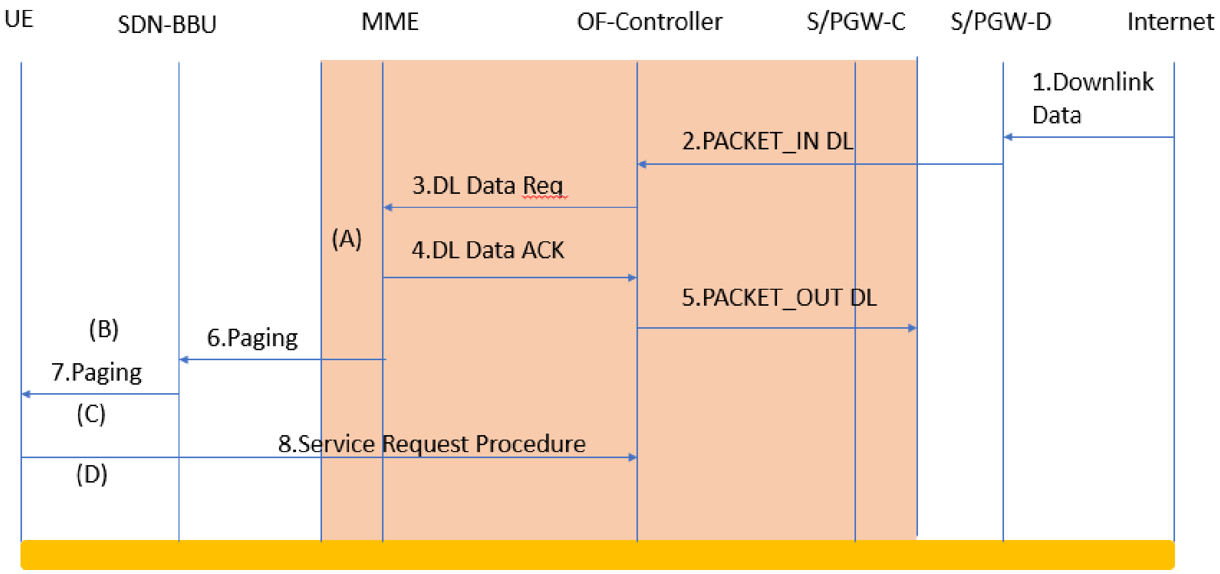


Figure 3.3-1: Proposed System Bearer Establishment Procedure

3.3.2.2 Network Triggered Requests

Network Triggered Service Request (as shown in Figure 3.3-1, (Steps A-D)) is instantiated by the UE (only when UE is in *ECM-Idle* mode) to allow bearer establishment in the network. The bearers are established so that the network can send data or signalling to the UE.

Step A: The SGW-D does a header matching in its flow table when a downlink data packet from an *ECM-Idle* state UE is received. A *Downlink Data Notification* message is sent to the controller from the SGW-D. The controller contains the PACKET_IN message which has the UE profile in it. The MME sends a *Downlink Data Notification Acknowledgment* to the controller, which is forwarded to the data plane switches in a PACKET-OUT message.

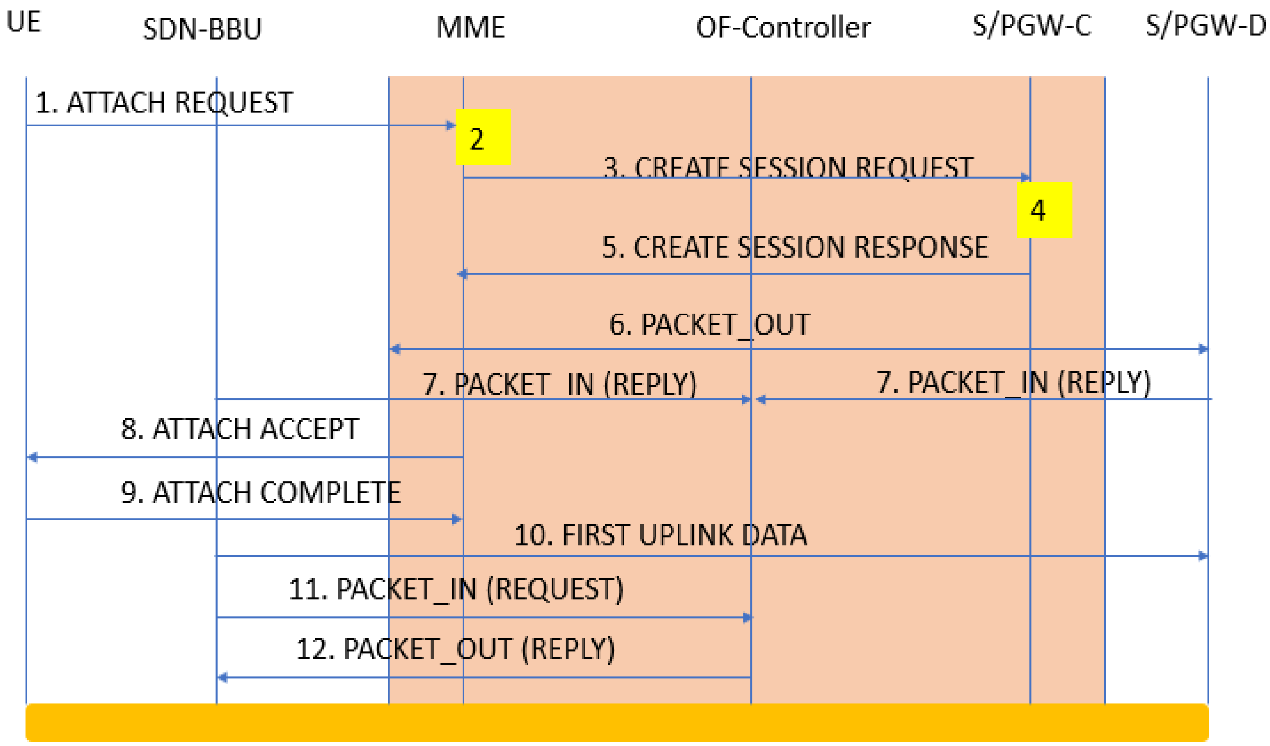


Figure 3.3-2:Proposed System Initial Attachment Procedure

Step B: The MME authorises the UE and registers it in this step. A *Paging* message is sent to every SDN-BBU belonging to same TA as the registered UE. The SDN-BBU after receiving the *Paging* message from MME, pages the UE. The MME has the UE last location information so it can tract the UE even if it is in *Idle* mode. However, the MME executes a TA update if the UE is in a new location that is not listed.

Step C: The UE triggered *Service Request* procedure is executed when the UE receives a paging receipt. The UE attaches to another BBU and establishes the RRC connection so that the TA update can be sent to the network. The TAU may transit the *Idle* state of the UE to *Connected* State in the network.

Step D: In the ultimate step, the UE location gets stored in the LC and flow entries are installed in the SDN-BBU switch. The downlink data is transmitted from SGW-D towards the UE [85].

3.4 SD-VCRAN AND PROGRAMMABLE EPC SYSTEM ANALYSIS

In this section, the proposed system critically examines the data and control plane in terms of: flexibility (services, network upgrades and traffic engineering), complexity (implementation, elasticity, and cost), and performance (QoS, routing and traffic monitoring).

3.4.1 Control Plane

1. ***Flexibility:*** The EPC network overprovisioning is removed by the VNFs flexibility while the 3GPP control interfaces are kept unchanged. The VNFs add intelligence and automation in the network. The SDN controller based on demands performs dynamic load balancing for the VMs, achieving elasticity horizontally and vertically in the control plane. The VNF applications receive efficient traffic management and optimal routing in the proposed network architecture. The network upgrades are simple and fast as they are software based and need no vendor interference. The network providers have the possibility to deploy both production and test framework on the same platform.
2. ***Complexity****:* The proposed system needs both hardware and software change during implementation phase. The VNFs can be hosted as VMs, which will communicate with the controller via REST API. A virtualised server platform is used for implementing the VMS and SDN controller, hence the complexity of the operator network is hugely reduced (less physical machines). The SDN controller has the freedom to attach to many switches and process several requests per second. The system scalability is enhanced as the complexity reduces. In terms of cost, the maintenance cost is massively reduced due to the new programmable EPC.
3. ***Performance:*** In the SD-vCRAN and Programmable EPC architecture the data routing efficiency is increased in comparison to traditional mobile network. The load in the nodes such as SGW and PGW are managed the controller using a load balancing approach, which eliminates the chance or overloading of one of these nodes. Compared to the traditional load balancing mechanism which does not use current load information, the proposed system preserves the GTP-C functionality and applies the load balancing mechanism in the S/PGW-C applications based on traffic flow statistics. The SDN controller which is able to handle several OF switches and LC, it introduces single failure point in the network. The network performance in terms of redundancy can be improved by implementing more than one SDN controller, in which the second will act as backup in case of primary controller failure.

3.4.2 Data Plane

1. ***Flexibility:***

   The OF switch capabilities defines the elasticity in the proposed system data plane. The controller has global view of the proposed architecture, which allows the controller to obtain periodic updates related to received, sent and lost packets, as well as overloaded links. Hence, proposed architecture redundancy is improved and traffic flows are equally distributed in switches with less overload. The QoS for different traffics is supported by the architecture via flow aggregation and prioritising the traffic based on the QCIs. The OF switch does not keep the tunnel establishment, instead it triggers tunnel erase after a predefined time or when the connection between BBU and PGW expires. The controller traffic management manages the traffic.
2. ***Complexity:*** OF hardware switches combined with Open vSwitches form the data plane which simplifies the GTP encapsulation by combining flows into bundles based on type of service and directing Layer 2 forwarding. LC aggregates flows coming from UE to SDN-

BBU into bundles based on LocIP address and differentiates between sessions depending on destination port. For large topologies, the proposed system guarantees improved scalability. In this research a fat tree topology with access, core and aggregation switches is implemented. Additionally, the cost factor is lowered since OF switches are not as expensive as physical hardware.

3. ***Performance:*** The flow tables in the data plane switches are configured and controlled by the SDN controller over OF protocol. The switches are responsible these actions: packet forwarding, packet encapsulation and packet drop. The switch lookup in the flow table after receiving the packet and perform one of the actions mentioned before. In case of no match, the packet is forwarded to the controller. The packets can be forwarded proactively or reactively by the controller. In this proposed system, the flows are routed proactively which means the rules are predefined in the switch. The switch sends PACKET_IN message to the controller, which has the port statistics and number of dropped packets and received or transmitted packets. The controller is able to identify traffic irregularity and enforce monitoring rules to get the load information in the switches. The Programmable EPC uses an adaptive linear prediction based algorithm to reduce the processed data amount and prevent overhead in network due to load information.

3.5 PROPOSED SYSTEM BENEFITS AND APPLICATIONS

The core benefits of these SDN based proposed architecture is listed as follows:

- ***Mobility***: Improved mobility corresponding to the system load. The signalling load to the core node (MME) is reduced. SDN- BBU pool migration to core helps realise CRAN architectures full potential. The LC handles mobility and reduces strain on the controller.
- **User centric system architecture**: the user centric system design provides improved uplink and downlink data rates as they are decoupled and thus optimization is done separately. Also, improved system power usage as less physical hardware and reduced processing time in the network entities.
- ***Enhanced QoS support***: Virtualised system architecture allows operators to differentiate between traffic types based on the QoS demand.
- ***Network Sharing***: Possibility to extend the system for network sharing scenarios by offering controllers for each operator. Each operator can separately implement different mobility standards.
- ***Improve network capabilities***: By integrating CRAN with advanced application services through the SDN controller.
- ***Cost Reduction***: Virtualisation of network components reduce cost which is quite notable compared to proprietary hardware components.
- ***Flexibility***: Freedom to dimension EPC entities and access network component (BBUs), provides better resource utilization in the network. Easy and flexible network upgrades by allocating more resources or adding new functionalities to the network.
- ***Traffic Engineering***: the global network assists the controller to provide better data traffic shaping and load balancing in the network

3.6 SIMULATION TOOLS

The tools and technologies used in the research are presented in this section. Network Function Virtualisation (NFV) and Software Defined Networking (SDN) technology and tools are used for the proposed system and test environment implementation and configuration. The research uses an emulated test environment in a network simulation tool called Mininet. In addition to that, the proposed system performance is analysed and compared for two Open-source SDN controllers. The QoS algorithms and testing scenarios are presented in the section.

3.6.1 Virtualization Tool

Most of the proprietary hardware machines are underutilised in the mobile networks as they are designed to run on a single operating system (OS). Implementing virtualisation (a technique used to abstract resources) in the proposed system, the network resources and components can be accessed in a consistent manner by network providers. The virtualised system provides the benefits of managing pooled resource and increases network responsiveness to dynamic QoS or other user needs. The VMs can support different OS and many applications and the system resources can be distributed between multiple VMs. The hypervisor is used to host multiple VMs on a single physical machine. In this research, VirtualBox (VB) is used to set up the simulation test, as presented in section 3.6.1.1.

3.6.1.1 Oracle VM Virtual Box(VB)

It is a cross-platform virtualization application. VirtualBox is a hypervisor which allows to install and run several VMs in the testing environment. Based on the operational needs the resources are distributed in the VMs. The benefits of VirtualBox in implementing the simulation testbed are outlined in this section. A detailed description of the features and configurations are provided in [86].

A VirtualBox snapshot, allows to save a state of a virtual machine for later use. The VM can return to the previous state at any time using the change log provided by the snapshot, even when the VM has changed later. Any writable data on a VM becomes read-only when the snapshot is taken. VirtualBox administrators can take several snapshots of a VM to create multiple possible point-in-time restore points. When a VM returns to its state from a snapshot, current disk and memory states are deleted and the snapshot becomes the new parent snapshot for that VM [86].

In this thesis the author implemented the proposed system testbed using the VirtualBox 5.1, which permits three network connections modes: Bridged Networking Configuration, Network Address Translation (NAT) Configuration and Host-Only Network Configuration. as discussed ahead. The simulation set up in this proposed system uses Bridged Networking Configuration. In this mode, the VM uses the network adapter on the host system to connect to a network. The VM connects to the Local Area Network in the host system with the host network assistance. The bridged networking is operational with both wired and non-wired host network adapters. Each VM in the bridged network has its own identity.

3.6.2 Network Emulation Test Bed

3.6.2.1 Mininet

Mininet is a software emulator used to prototype a large network composed of virtual hosts, switches, controllers, and links, on a single machine. A SDN prototype can be used by user to simulate network topologies using OF switches via Mininet. Mininet is used as a flexible network testbed and runs on standard Linux network software. Using Python Application Programming Interface (API) complex topologies can be tested without any physical connections by Mininet. The software emulator also includes a topology aware CLI. Multiple hosts and switches can be run on a single OS kernel using process based virtualisation. It can create controllers for network controlling using OF switches, hosts as communication medium in the simulated network, and OF switches. The emulator uses virtual Ethernet pairs to connect the switches and hosts in the simulated network. The main benefit of the emulator is that it allows the emulated network (having its own controller) to connect to other controllers by particularising the IP on which controller is running.

3.6.2.2 Topology

For the proposed system a three-layered mobile topology: core, aggregation, and access, is implemented for testing set up. The ring topology is used for access network and for the core network a mesh configuration is used to obtain maximum flexibility. Figure 3.6-1, shows the proposed topology – a fat tree topology with eight edge switches, four aggregation switches and two core switches. Each edge switch is attached to five hosts. In this topology, every top tier switch connects with the lower-tier switch in a full mesh topology. The number of links in core is equal to number of aggregation switches and the number of links in aggregation switches is equal to number of edge switches. k, k^2, and k^3 represent the number of core, aggregation, and edge switches respectively in the proposed topology. In this topology, a mobile network with 40 RRHs connected to 8 edge switches is simulated. The number of LCs which create the unique LocIP, for each user is equal to the number of edge switches. Each SDN-BBU has a LC which assigs unique LocIP to it. The SGW data plane functions are completely replaced by aggregation switches. The aggregation switches are twice in number of core switches. The configuration code is developed to customise the bandwidth capacity in the links between the switches. The edge and host link has 50Mbps bandwidth limitation and it is 250Mbps for the links between aggregation and edge switches. The link between core and aggregation switches is capped at 2Gbps. In the Mininet, Linux Traffic Control is executed on each link data rate. The Linux Traffic control shaped the traffic using packet schedulers to a configured rate. Virtual Ethernet Interface is provided to each emulated host. The command below is used to implement the custom topology for simulation purpose, where the controller IP is 127.0.0.1:6653 and 10.0.0/8 network is used to define the hosts.

```
sudo mn --custom FatTree.py --topo FatTree --switch ovsk --controller=remote,
ip=127.0.0.1:6653 --ipbase 10.0.0.0/8 --link tc
```

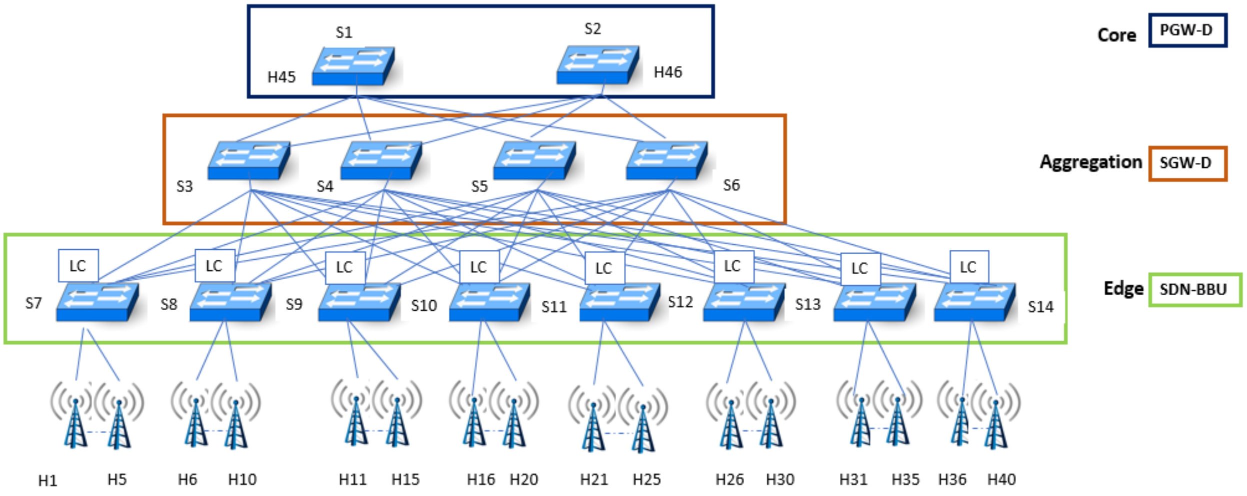


Figure 3.6-1: Proposed Topology (Fat Tree Topology)

3.7 SOFTWARE DEFINED NETWORKING CONTROLLERS

In the proposed network architecture, the Floodlight and OpenDaylight controllers are implemented for the emulation. A detailed comparison between these controller's in terms of architecture, the modules implemented in the proposed system, and the Graphical User Interface is presented in this section. Also, in Chapter 4 the controllers are compared in the simulation environment corresponding to throughput, delay, and datagram loss. Additionally, the Floodlight controller QoS module and the algorithms behind the controller extensions is briefly discussed.

3.7.1 OpenDaylight Controller (ODL-C)

The OpenDaylight is an Open-source collaborative project highly supported by big vendors like Cisco, Big Switch, Brocade, Citrix, Ericsson, Juniper Networks, IBM, VMware, NEC, and other networking companies. The ODL-C is written in Java and can be implemented on any Java supporting hardware. The fourth release of OpenDaylight (Beryllium) is deployed in the proposed system emulation set up. It includes support for SDN, NFV and Network Virtualisation (NV). ODL-C has multiple pluggable modules such as: protocols, interfaces, and applications, which can be utilised to change the ODL-C according to the need of the network administrator. The ODL-C architecture is outlined below and shown in Figure 3.7-1.

- The **top layer** in the architecture consists of network logic applications which monitor and control the network performance. The web based interface DLUX manages the network by gaining information related to network, flow statistics, host locations from ODL topology and host databases. The top layer has a developer toolkit called Network Embedded Experience (NeXt) which provides the network topology user interface elements which will visualise the aggregated nodes, complex network topologies, layout algorithms, traffic path/tunnel/ group and map overlays. The top layer applications use open API interface to communicate with the ODL-C.

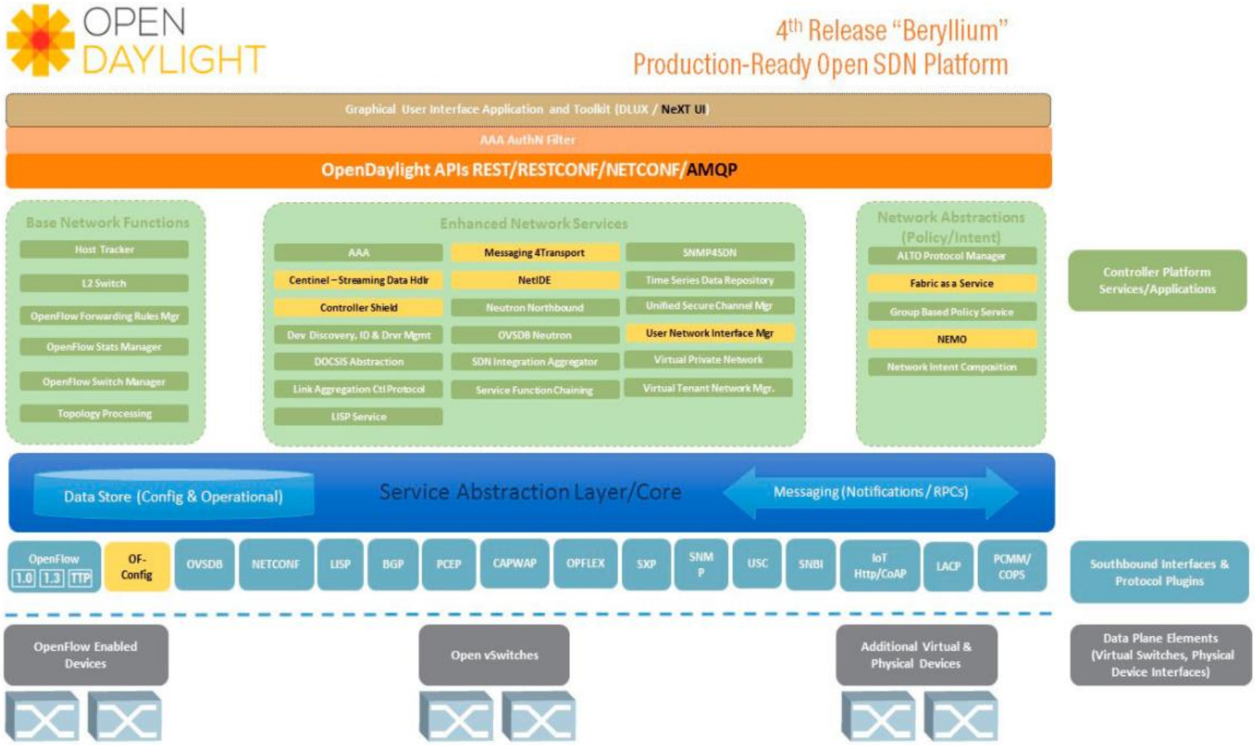


Figure 3.7-1: OpenDaylight Controller, source [87]

- The **middle layer** provides the framework which runs the SDN abstractions and APIs to connect to the top layer (i.e. the *Northbound interface*). The ODL Beryllium provides *Application Layer Traffic Optimization* (ALTO). The *ALTO* is implemented using IETF protocol to provide applications the network information. *ALTO* is liable for enabling simple network views and services. *Northbound* can be more easily mapped onto the physical network through the *Fabric as a Service* (FaaS), which creates a common abstraction layer on top of a physical network. The controller platform uses *NEMO,* a Domain Specific Language (DSL) to enable network applications and users to define their requirements for network resources, services, and logical operation. An application centric policy model called *Group Based Policy (*GBP) in this layer separates network infrastructure information from application connectivity requirements. *NetIDE* functions to provide inside a single network portability and cooperation via a client or server multi-controller architecture. It is responsible to allow OpenDaylight to support and run SDN Applications written for other SDN Controllers. The southbound SNMP plugin called SNMP4SDN provides the legacy switches SDN controller benefits over the SNMP interface. A framework used to collect, store, maintain, query the times series in ODL is called *Time Series Data Repository* (TSDR). Encrypted communication between endpoints obtain a central server by the *Unifies Secure Channel* (USC), where the controller is informed about its encryption capabilities by the client-side agent. The controller gets instructed to encrypt the selected flows based on the business policies. *VPN* executes the infrastructure services corresponding to L3 VPN service. The SDN controller provides multi-tenant virtual network. The MD-SAL datastore saves the user database and credentials. A repository is created- Unified Security Plugin (USecPlugin) to provide northbound applications the controller security information. Other extended and base network functions which provide control of the network elements are implemented in this layer. More detailed description is provided in [88].

- Multiple protocols are supported in the *Southbound Interface* as separate plugins, such as i.e. OpenFlow 1.0, OpenFlow 1.3, BGP, LISP, CAPWAP, SNMP, NETCONF, etc. The MD-SAL is connected to these modules and provides support to the *Southbound Interface* protocols. MD-SAL defines the mapping between the controller and network devices.
- The data plane is composed of the forwarding elements: switches, routers, physical devices, virtual devices, etc. These forwarding elements is responsible for connecting all the endpoints within the network.

3.7.1.1 ODL Web Interface

The forwarding decisions and flow installation on all devices in the OF network is handled by *Simple Forwarding* application. The Address Resolution Protocol (ARP) message issued to find a host and install either destination or thirty-two entries across all the OF switches in the network based on the output ports towards the host. The essential benefit of ODL-C is that it allows L1-L4 flow entries installation in the switches using Web Interface.

3.7.2 Floodlight Controller (FL-C)

Like the ODL-C, Floodlight SDN controller is another open-source SDN project supported by SDN community and vendors like Big Switch. Other vendors such as Arista, Brocade, Dell, Google, IBM, Citrix, Extreme Networks, Fujitsu, HP, Intel, Juniper Networks, and Microsoft have fairly contributed to the FL-C APIs development. The FL-C is written in JAVA and is an Apache-licensed OF controller. Both Open vSwitch and OF switches which connect the OF with non-OF networks are supported by this controller. The FL-C architecture is shown in Figure 3.7-2. The REST API is implemented for communication between controller and applications. The author created and implemented four new modules: Mininet Queues, Add Custom Policies Scenario 1, Add Custom Policies Scenario 1, Load Balancing to the current FL-C implementation to realise the proposed system (see Figure 3.7-3, modules in blue boxes). The load balancing module is implemented in the architecture top layer as an application.

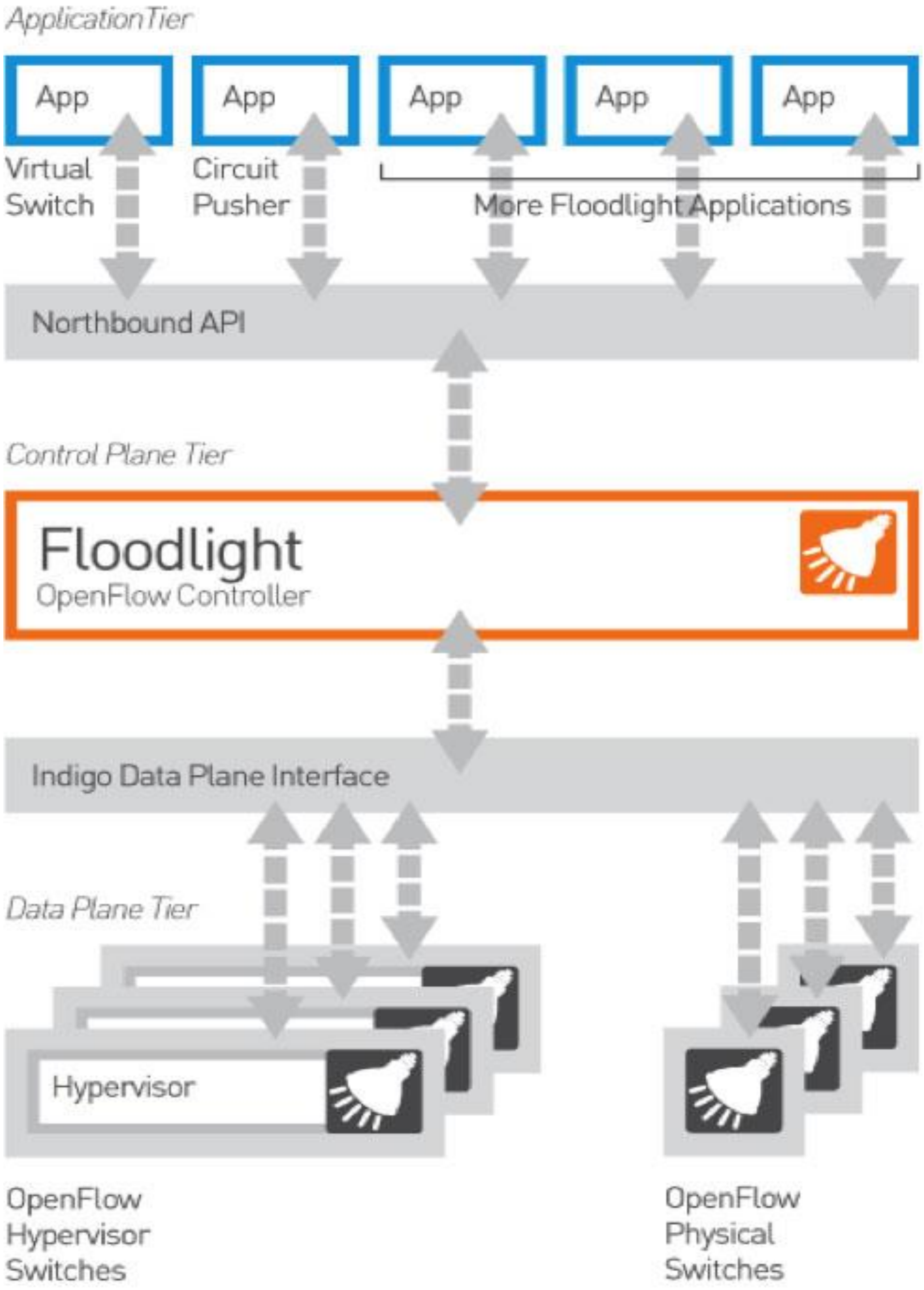


Figure 3.7-2: Floodlight Architecture, Source [89]

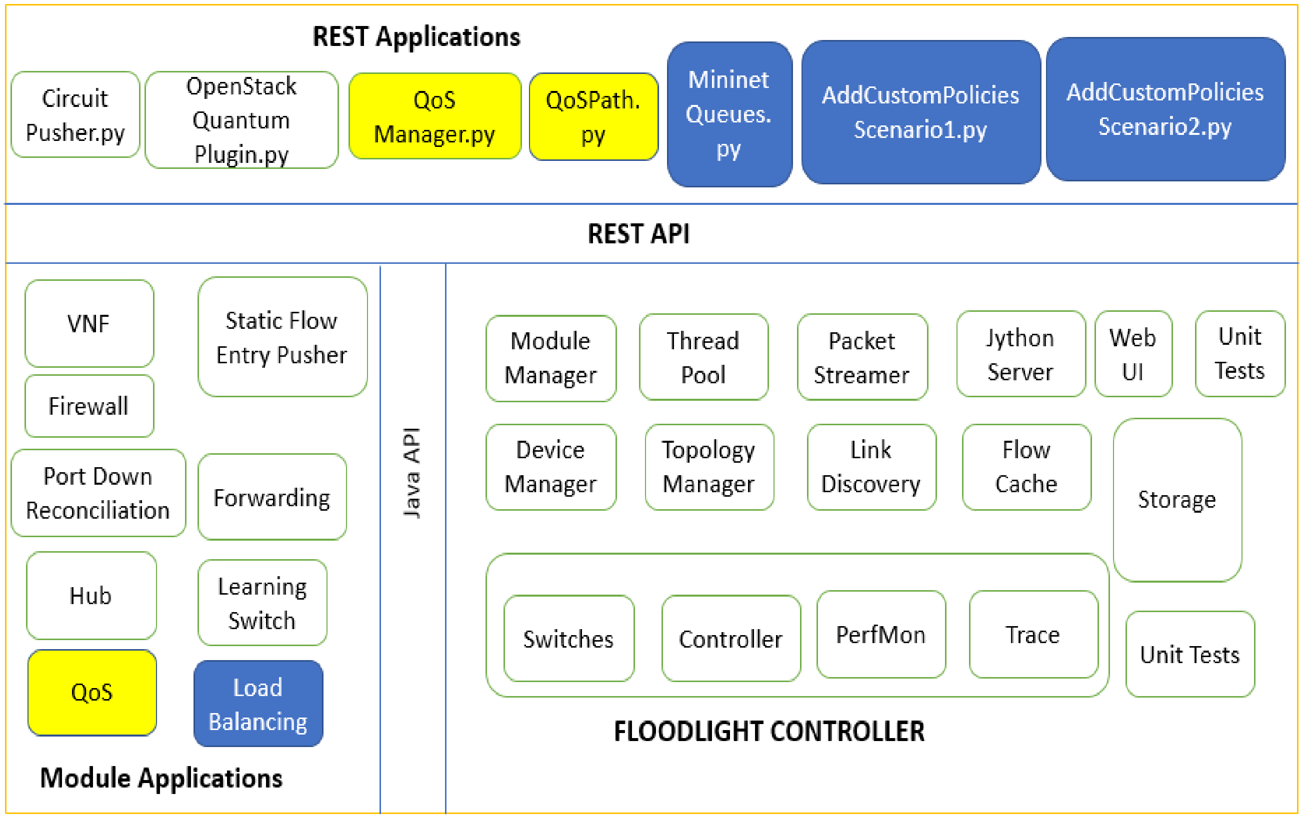


Figure 3.7-3: Proposed System QoS and Load Balancing Modules

3.7.2.1 FL Web Interface

The Floodlight controller on local host 127.0.0.1: 6653 is connected to the topology discussed in Section 3.6.2.2 and is implemented in Mininet. The fat tree topology implemented using FL-C can be viewed in a GUI Web Interface, as shown in Figure 3.7-4. The GUI interface Tool section > Tools display the QoS application output as shown in Figure 3.7-5. The current FL-C doesn't support flow entry configuration; therefore, the flow entry configuration can be done only on OF switches or in the application layer.

3.7.2.2 QoS Module

This module is liable to enforce *QoS* state into the proposed network over the REST API. It performs the following actions: configuration of DiffServ type of service (ToS), maximum rate limit and minimum rate limit. The TOS defines rules based on QoS and the QoS module installs a policy path.

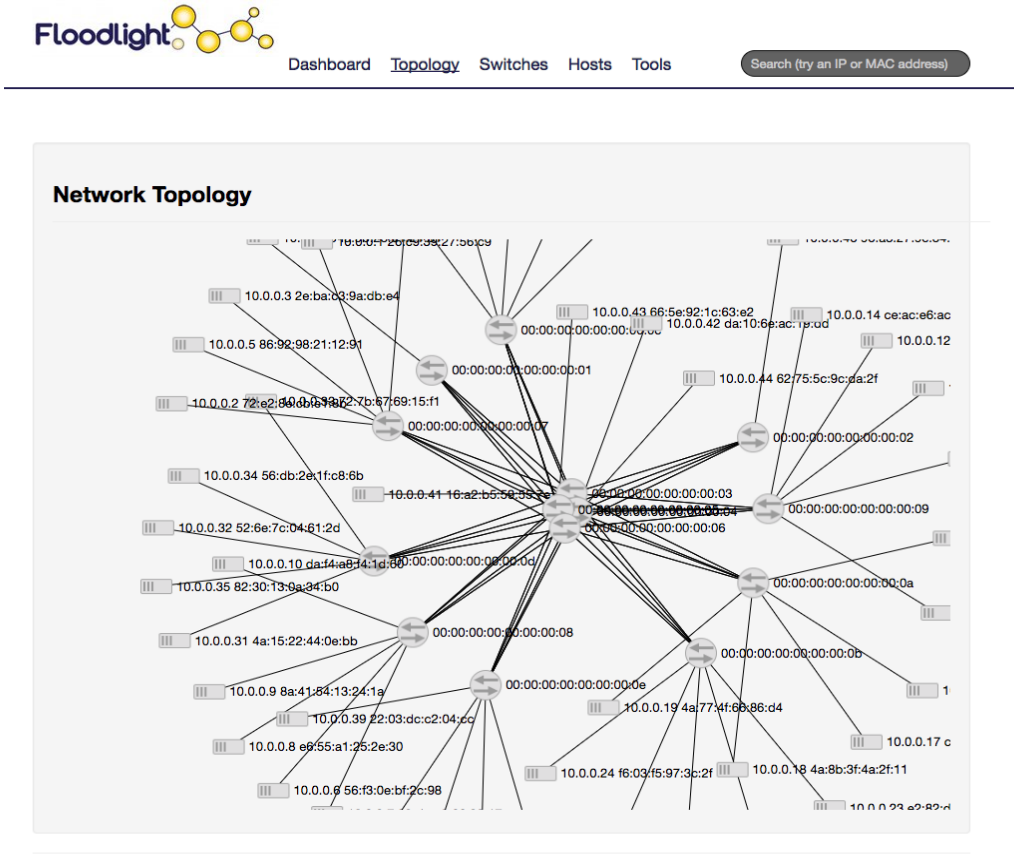


Figure 3.7-4: Proposed Topology in Floodlight

The three application scripts: QoS path, Circuit Pusher.py and QoSManager.py (as shown in Figure 3.7-3) compose the QoS module. New configured scripts in the blue boxes as shown in Figure 3.7-3 are added in the controller.

The current application uses OpenFlow 1.5 specifications, since it allows the extension of QoS module and other extended features like *meter table*s and *set-queue*. The QoS module is programmed in JAVA and uses python script for configuration purpose. The policies are executed on the path depending on each session ToS byte.

Three classes of services and the corresponding ToS values as discussed in Chapter 2 form the *QoS* application along with. In the GUI web interface, it is seen that the shortest path calculated by the switches form the policy path. Figure 3.7-5, present the policy installation in the FL-C where the video session has a ToS value of 34 allocated and priority values set to 3200.

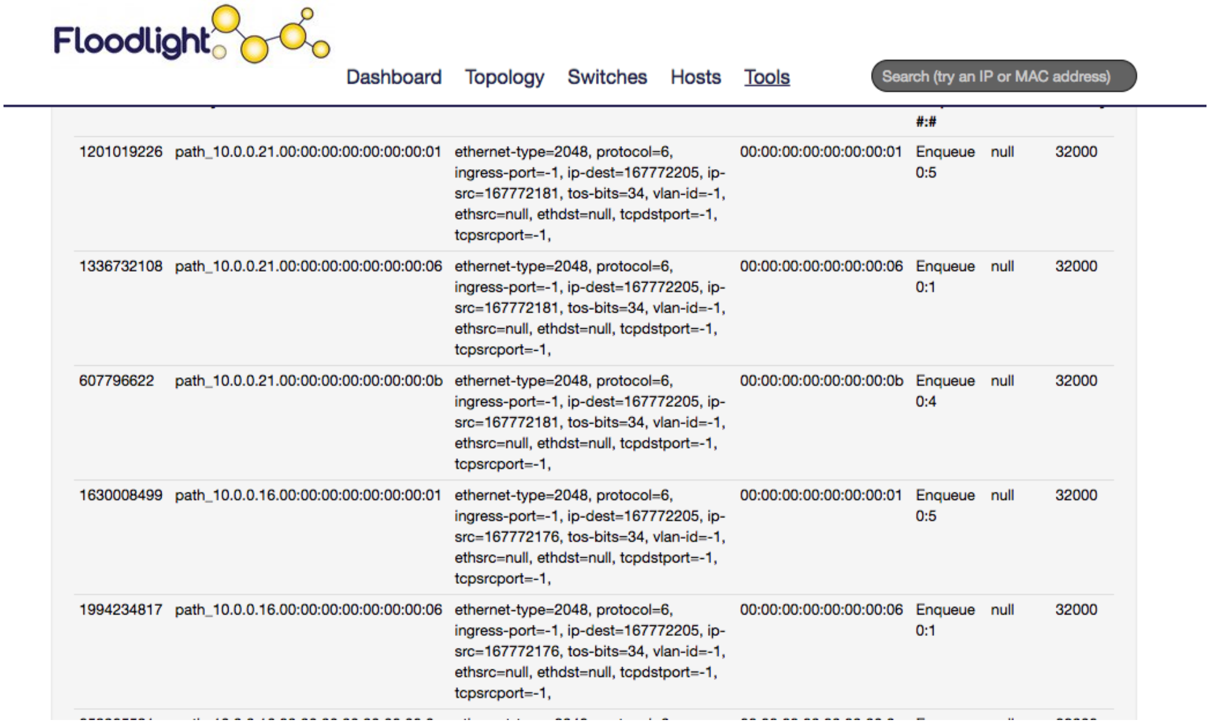
Each switch port on every bridge is configured with three QoS queues by the QoS script, and the queues then get assigned as shown in Table 6. The path between the client and server is chosen by the queueing algorithm and it also executes along the path a pre-defined policy. The shortest path between source and destination is returned by the QoS routing policy.

Queue Number	QoS Queue	Bandwidth (Max)
1	Q_0	20Mbps
2	Q_1	8Mbps
3	Q_2	6Mbps

Table 6: QoS Queues Configuration

The Linux capability of traffic controlling is used for rate-limiting by the Open vSwitch. For a configuration as presented below: the maximum rate is defined to 1Gbps (10^9 bits per second), and there are three QoS queues called $Q_{0,}$ Q_1, and $Q_{2.}$ The first QoS queue (Q_0) has maximum bandwidth, whereas Q_1 has maximum and minimum bandwidth rate of 20Mbps (2 x 10^7 bits per second) and Q_2 is configured to 2Mbps (2 x 10^6 bits per second) as below:

```
queue cmd = "sudo ovs-vsctl \%s -- --id=@defaultqos create qos type=linux -htb other -config:max-rate=1000000000 queues=0=@q0 ,1=@q1 ,2=@q2 -- --id=@q0 create queue other -config:min-rate=20000000 other -config:max-rate=20000000 -- --id=@q1 create queue other -config:max-rate=8000000 other-config:min-rate=8000000 -- --id=@q2 create queue other -config:maxrate=6000000 other -config:min-rate=6000000"
```



				#:#		
1201019226	path_10.0.0.21.00:00:00:00:00:00:00:01	ethernet-type=2048, protocol=6, ingress-port=-1, ip-dest=167772205, ip-src=167772181, tos-bits=34, vlan-id=-1, ethsrc=null, ethdst=null, tcpdstport=-1, tcpsrcport=-1,	00:00:00:00:00:00:00:01	Enqueue 0:5	null	32000
1336732108	path_10.0.0.21.00:00:00:00:00:00:00:06	ethernet-type=2048, protocol=6, ingress-port=-1, ip-dest=167772205, ip-src=167772181, tos-bits=34, vlan-id=-1, ethsrc=null, ethdst=null, tcpdstport=-1, tcpsrcport=-1,	00:00:00:00:00:00:00:06	Enqueue 0:1	null	32000
607796622	path_10.0.0.21.00:00:00:00:00:00:00:0b	ethernet-type=2048, protocol=6, ingress-port=-1, ip-dest=167772205, ip-src=167772181, tos-bits=34, vlan-id=-1, ethsrc=null, ethdst=null, tcpdstport=-1, tcpsrcport=-1,	00:00:00:00:00:00:00:0b	Enqueue 0:4	null	32000
1630008499	path_10.0.0.16.00:00:00:00:00:00:00:01	ethernet-type=2048, protocol=6, ingress-port=-1, ip-dest=167772205, ip-src=167772176, tos-bits=34, vlan-id=-1, ethsrc=null, ethdst=null, tcpdstport=-1, tcpsrcport=-1,	00:00:00:00:00:00:00:01	Enqueue 0:5	null	32000
1994234817	path_10.0.0.16.00:00:00:00:00:00:00:06	ethernet-type=2048, protocol=6, ingress-port=-1, ip-dest=167772205, ip-src=167772176, tos-bits=34, vlan-id=-1, ethsrc=null, ethdst=null, tcpdstport=-1, tcpsrcport=-1,	00:00:00:00:00:00:00:06	Enqueue 0:1	null	32000

Figure 3.7-5:QoS Module in implemented Floodlight Controller

Expedited Forwarding Service is offered to voice sessions since they require low latency and priority treatment to satisfy the QoS needs. In the proposed system for video sessions the data rate limit is set to 20Mbps (Q_1), VoIP sessions traffic limit is set to 8 Mbps and HTTP sessions are provided with 6Mbps data rate. The proposed system is analysed in three scenarios, which is presented in Chapter 4.

In Scenario 1 (see section 4.2.2), the network receives the voice session traffic from users attached to the SDN-BBUs concurrently, where the users are attached to SDN-BBUs through H_{37}, H_{32}, H_{27}, H_{22}, H_{17}, H_{12}, H_7, and H_2. These traffic data rate is limited to 8Mbps once it gets mapped to Q_1. The video session is classified as EF class. The QoS queue Q_2 receives BE data rate services and 6Mbps bandwidth is assigned for each session, which means all the users attached to the rest of the SDN-BBU switches will share the 6Mps bandwidth. In the FL-C QoS module between each user and core server a policy path is established. In Scenario 2, Section 4.2.2, the traffic between server H_{45} and H_{46} is balanced by executing the load balancing algorithm, when two policies defined by host H_1 have same data rate limitation for both H_{45} and H_{46} attached to S_1 and S_2. The policies are installed on the paths when the QoS queues for each port is assigned. After assigning three queues for each port, then the policies can be installed on the path.

3.7.2.3 Traffic Routing

The modules and scripts in the QoS application module are individualistic. The Tarjan algorithm is used to calculate the shortest path by recalling the *TopologyInstance.Java* module. The *Topology Manager* in the FL-C receives query from the *CircuitPusher.py* which is recalled by the *QoSPath.py* script. The *Topology Manager* is responsible to recall the *TopologyInstance.Java* module. The REST API is used for communication with the Floodlight *Topology Manager* module and the script *CircuitPusher.py* obtains the route from source to destination. Based on Depth First Search (DFS), the Tarjan algorithm in the *TopologyInstance.Java* module forms clusters. The algorithm finds strongly connected set of nodes of a directed graph. The set of nodes have minimum 1 oriented path between every 2 nodes. The routing mechanism is outlined below:

- Step 1: Ignoring broadcast domain links form clusters, which are strongly connected set of nodes. The Tarjan algorithm is implemented to compute the strongly connected set of nodes in a network which might have unidirectional links.
- Step 2. Find the shortest path tress for unicast routing in each cluster, where the trees are rooted at the destination. Dijkstra algorithm is enforced in each cluster to calculate the shortest path. Tunnel and direct link cost are same.
- Step 3. In each cluster, the broadcast tree is computed. The tunnel links are discarded by setting the tunnel link cost high. To use minimum number of clusters the cost is set to number of nodes in the cluster +1.
- Step 4. The topology is displayed after the path is returned.

In this routing mechanism, the two-layered mesh proposed topology as shown in Figure 3.6-1, there is no isolated clusters as the mechanism is reliant on the topology. Therefore, the Step 3 is not executed for the proposed topology.

3.7.2.3.1 Depth First Search (DFS)

Based on DFS, the Tarjan algorithm is applied to obtain the list of nodes in the proposed tree topology. DFS algorithm is used to discover the most strongly connected set of nodes in a graph. DFS algorithm finds and groups the strongly connected nodes into clusters taking a directed graph as input. Even if a node has a degree 0 it can still form a strongly connected component. DFS is only applied to all nodes that have not been defined during the search from an arbitrary node, hence the search is performed on each one only once. Any node of among the set of strongly connected nodes can serve as a root which is the first discovered component. An index value acts as a counter for the DFS node number. The sets of undefined nodes *v* in set of vertices *V* has an index value. The node stack (S) acts as a datastore for the history of discovered nodes prior to linking with strongly connected set of nodes. The lowlink is calculated as the mimumum between its own lowlink index and next lowlink index, where lowlink is any node smallest index provided the node is reachable from *v*. When the DFS visits the next node, the current node lowlink index is calculated as minimum between current index value and next nodes index value. The node which has an index equal to its lowlink index, its discarded from the stack. The above described procedure is used to get all the nodes until the first root at the destination.

3.7.2.3.2 Dijkstra Algorithm

The proposed system uses the Dijkstra Algorithm for shortest path calculation used to route the traffic in the network. It makes routing choices based at that moment such that the global optimum is attained in the end. Based on weights and costs the algorithm is implemented in the FL-C. The algorithm is modified to realise the requirements of the proposed network. The random load on each aggregation switch corresponds to the weights on the links, so a positive value is allocated for each weight. All the links are assigned with an infinite value for all nodes and all forerunners during initialisation, as well as during initialisation all nodes unreachable. Followed up, the queue (Q) gets installed with all nodes and the node *u* with shortest weight is extracted while other nodes are in queue. The u: =vertex in Q with minimum distance (*min dist(u)*) looks for the vertex *u* in Q which has smallest distance value. During the last stage, the shortest path distance (*dist[v]*) moves towards the real shortest path weight. Each node is relaxed only once and for each node *v* close to finding the node *u* it is examined if the shortest weight can be reduced moving toward *u*. The forerunner of *v* is then updated to *u* when the shorted path is found [90].

3.7.2.4 Load Balancing

The load balancing procedure is executed to evenly distribute the traffic in the network. In this research, the round robin algorithm is used to balance the traffic between servers which have same policy shared with the clients. It forwards a new connection to the next server in the member list containing the list of servers. The client of this service receives a Virtual IP (VIP) and use it as a destination IP address, where the VIP is composed of Virtual IP, protocol such as TCP and UDP, and port. In the testing environment, multiple assumptions have been made presented as follows:

- The same server pool can have one or multiple VIPs mapped to it. These VIPs must share the same load balancing policy whether random or round-robin.
- Only one VIP is related to each server pool.
- The idle time out is set to 5 seconds for flow rules installation.
- Same input and output switch must be used to enter and leave the OF cluster by the packets.

- The flow rules installed by a server or VIP or server pool is kept even if they are deleted. After 5 second the flow tables get timeout automatically.
- For ODL-C, there is no support for VIP. In ODL-C, load balancer assumes that packets that have VIP IP address as destination are routed to it via external procedures. Adding static entries to the ARP table of source or client host the ARP for VIP can be solved. IN FL-C the ARP is solved automatically.

CHAPTER 4

4 SIMULATION AND RESULTS

In this chapter, the proposed system is validated under different scenarios using the emulation testbed described in Chapter 3. The simulation results are presented and analysed. The chapter has four sections covering: scenario 1, 2,3, and 4. Scenario 1 through 3 presents emulation results. Scenario 4 presents a comparative analysis between two Floodlight and OpenDaylight SDN controllers.

4.1 PROPOSED SYSTEM PERFORMANCE

The SDN based proposed system architecture aims to improve flexibility by virtualising the access and core network function components. The implementation complexity has been presented in section 3.6.2.2. The QoS module and routing procedures have been outlined in section 3.7.2 and section 3.7.2.3 respectively. The Floodlight controller executes the QoS and Load Balancing module. The emulated network performance is analysed in terms of throughput, packet loss and jitter for different traffic session such as: video, VoIP, and HTTP. The expected throughput outcomes should be a close match with the data rate limitation. The throughput outcomes are expected to closely correspond to the data rate limits set on switch port queues (refer to section 3.7.2.2). For VoIP and interactive video, the QoS requirements are set as: latency no more than 150ms, jitter less than 30ms, and packet loss less than 1%. Similarly, the latency should not exceed 4-5 sec and packet loss is capped to 5%. Both OpenDaylight and Floodlight controllers support the chosen load balancing algorithm.

4.2 EMULATION SET-UP

The testing environment consists of five scenarios and all of them are tested for the same topology as shown in Figure 3.6-1. Scenario 1 and Scenario 2 are focused on policy installation and the KPIs for performance analysis are: throughput, jitter, and datagram loss. In Scenario 1 and Scenario 2 the only difference is in the number of policies installed in each host. Scenario 1 has single policy installed in each host and Scenario 2 has more than one policy installed in each host. The servers receive traffic from the hosts in both the scenarios in a network of 40 SDN-BBUs having several users associated with it. The traffic is sent simultaneously from the UE to the network. The SGW-D selection and load balancing mechanism at PGW-D is obtained with the returned path. In Scenario 3, the load balancing is tested at the SDN-BBU level to cancel overload in the SDN-BBU clusters and manage BBU handovers. Both the SDN controllers discussed in Section 3.7 are tested using the proposed load balancing mechanism (see section 3.7.2.4). The clients are served in a Round-Robin pattern by the pool based servers. In Scenario 4, the SDN controller performance is examined.

4.2.1 Scenario 1 – Single Policy Installation

In this scenario, server H_{45} receives traffic through OFSW2 from hosts H_1, H_2, H_3, H_4, and H_5 attached to switch S_7. Policies are installed between servers and clients for UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) packets.

TCP is a connection-oriented protocol which establishes an end to end connection between computers before transferring the data. It ensures packets are not lost by using the three-way handshake, flow, error and congestion control. It makes sure that the data sent from source are received accurately by the destination. It attempts data re-transmission in case, data received is not in the proper format. Following protocols use TCP for transmitting data:

- HTTP(Hypertext Transfer Protocol),
- HTTPs(Hypertext Transfer Protocol Secure)
- FTP(File Transfer Protocol)
- SMTP(Simple Mail Transfer Protocol), etc.

On the other hand, UDP is a connection-less protocol since it does not determine the connection before sending data. UDP is less reliable, however it is used to transfer the data at a faster rate (such as audio and video files).

The data rate is capped to 20Mbps when DSCP H_1 sends video traffic as shown in Table 6. The DSCP AF41 and AF42/AF43 are marked for video session traffic and interactive video session traffic. The VoIP session is marked to EF class and with DSCP value = 46 and served on priority basis at each node.

The single policies installed between hosts H_1, H_2, H_3, H_4, and H_5, and server H_{45} are summarised in Table 7, where HTTP session are treated with lowest priority and VoIP session gets highest priority.

Host	Server	Session	Protocol	ToS	DSCP	Service	Data Rate	Queue	Priority
H_1	H_{45}	Video	UDP	136	34	AF	20Mbps	0	32000
H_2	H_{45}	Voice	UDP	184	46	EF	8Mbps	1	32767
H_3	H_{45}	HTTP	TCP	0	0	BE	6Mbps	2	30000
H_4	H_{45}	HTTP	TCP	0	0	BE	6Mbps	2	30000
H_5	H_{45}	HTTP	TCP	0	0	BE	6Mbps	2	30000

Table 7: Policy Installation in Scenario 1

4.2.1.1 Throughput

All the traffic is dumped into the network simultaneously using *iperf* command in Mininet for both TCP and UDP traffic. The bandwidth limitation is set as shown in Table 6, for single policy installation.

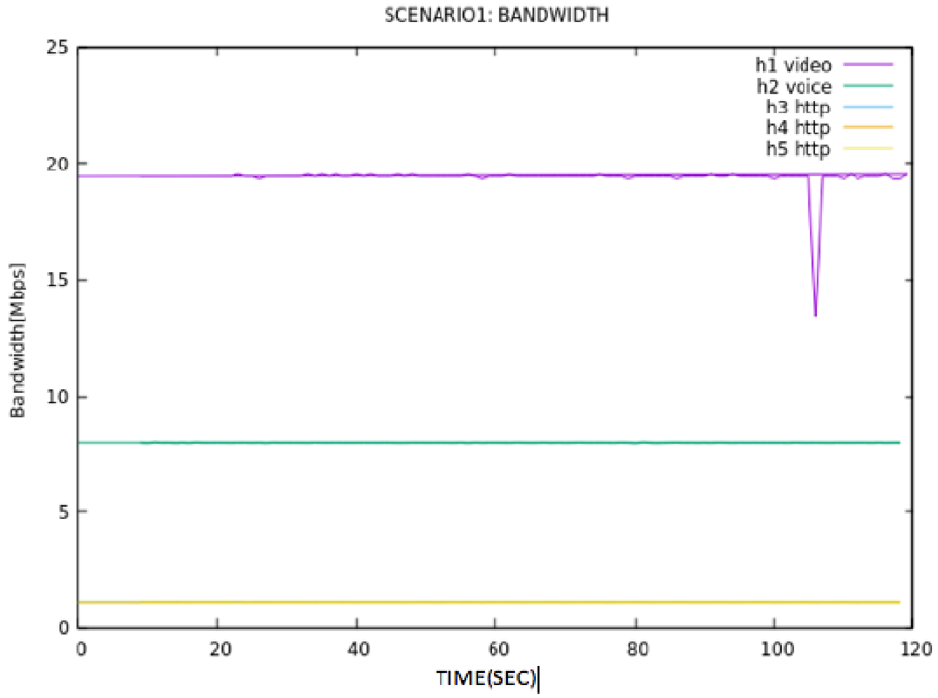


Figure 4.2-1: Throughput for Scenario 1

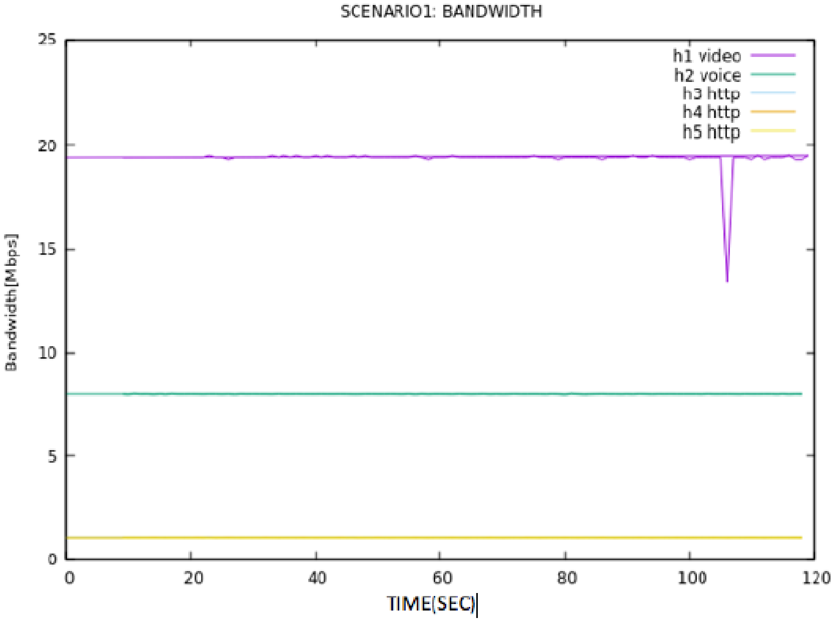
Figure 4.2-1, shows that video session receives 19.2 Mbps, voice session receives 7.7 Mbps and three HTTP sessions receives up to 6 Mbps approximately, which means the HTTP sessions avail the maximum bandwidth assigned. The HTTP connections in the figure above are overlapped as they have most of the time also, the low throughput is due to fact that the bandwidth is approximately same for three HTTP connections. The throughput for traffic sessions is illustrated for a time interval of 120s. The plots were generated using GNU plot.

The *iperf* configurations commands are presented in Table 8. Where S= TOS, the test duration of 120s is given by 1-t and W = window size. For TCP W represents the window size and for UDP it represents the buffer size.

Host	**Commands (W=256)**	**Commands (W=1024)**
H_1	Iperf -c 10.0.0.45 -S 136 -i 1 -t 120 -u -b 20m -w 256k	Iperf -c 10.0.0.45 -S 136 -i 1 -t 120 -u -b 20m -w 1024k – l 250
H_2	Iperf -c 10.0.0.45 -S 184 -i 1 -t 120 -u -b 8m -w 256k	Iperf -c 10.0.0.45 -S 184 -i 1 -t 120 -u -b 8m -w 1024k – l 250
H_3	Iperf -c 10.0.0.45 -S 136 -i1 -t 120 -w 128k	Iperf -c 10.0.0.45 -S 136 -i1 -t 120 -w 256 -l -250
H_4	Iperf -c 10.0.0.45 -S 136 -i 1 -t 120 - w 128k	Iperf -c 10.0.0.45 -S 136 -i1 -t 120 -w 256 -l -250

H_5	Iperf -c 10.0.0.45 -S 136 -i 1 -t 120 - w 128k	Iperf -c 10.0.0.45 -S 136 -i1 -t 120 - w 256 -l -250

Table 8: Iperf Configuration Commands



(i)

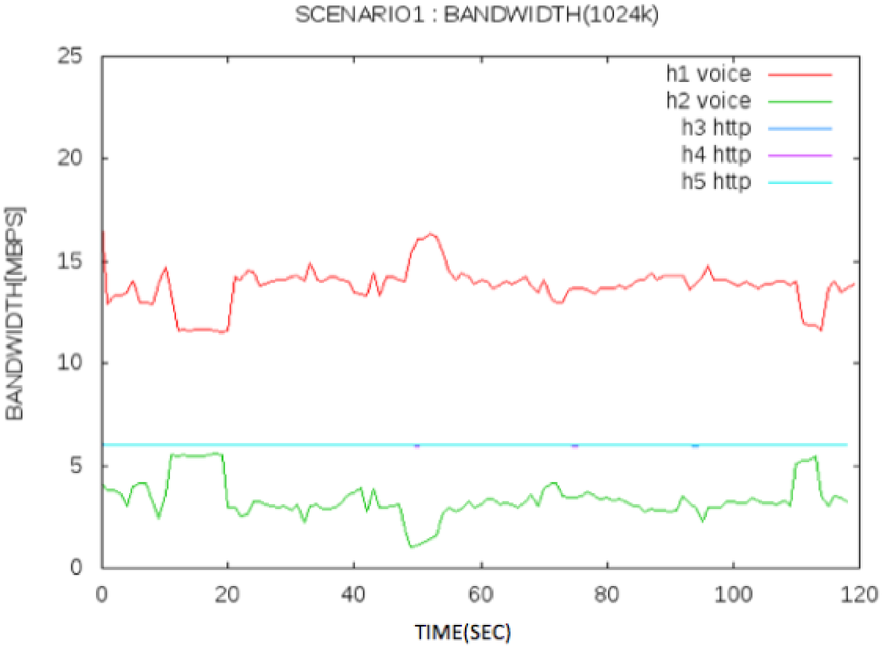
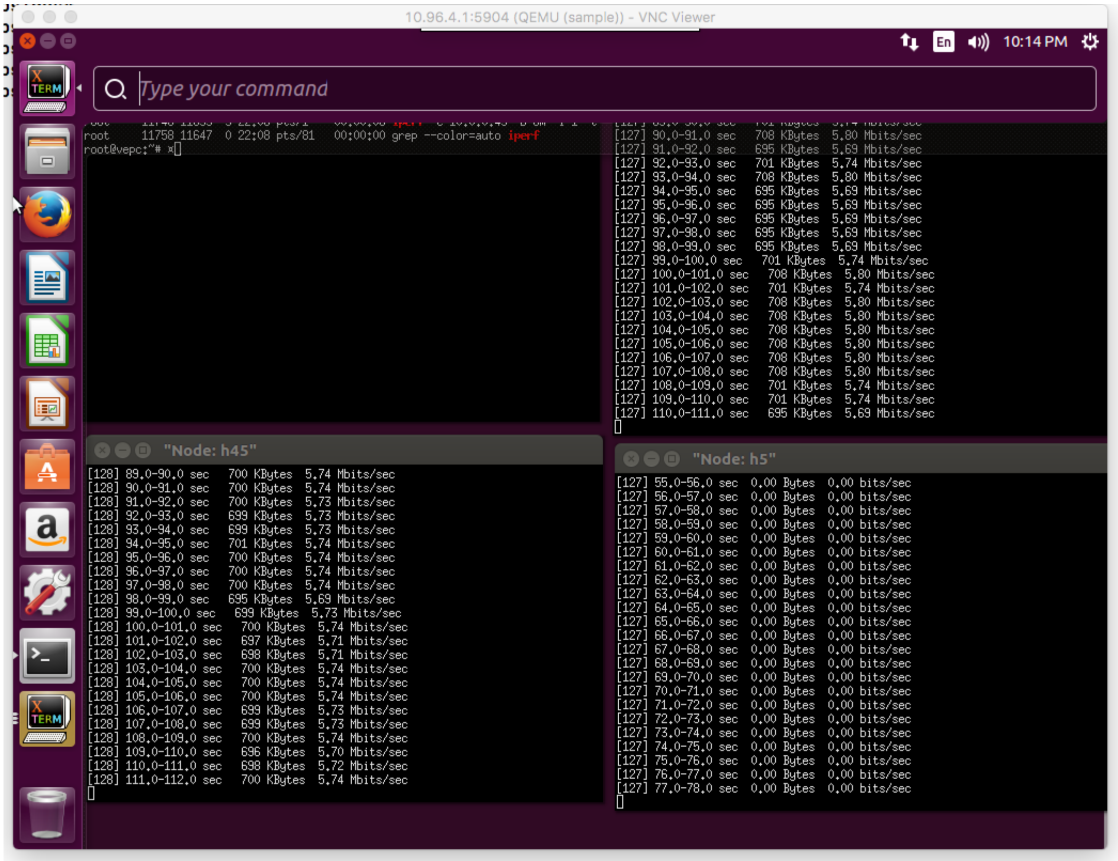


Figure 4.2-2: (i) Throughput without fragmentation, (ii) Throughput after fragmentation



```
root     11758 11647  0 22:08 pts/81    00:00:00 grep --color=auto iperf
root@vepc:~# x
```

```
[127] 90.0-91.0 sec   708 KBytes  5.80 Mbits/sec
[127] 91.0-92.0 sec   695 KBytes  5.69 Mbits/sec
[127] 92.0-93.0 sec   701 KBytes  5.74 Mbits/sec
[127] 93.0-94.0 sec   708 KBytes  5.80 Mbits/sec
[127] 94.0-95.0 sec   695 KBytes  5.69 Mbits/sec
[127] 95.0-96.0 sec   695 KBytes  5.69 Mbits/sec
[127] 96.0-97.0 sec   695 KBytes  5.69 Mbits/sec
[127] 97.0-98.0 sec   695 KBytes  5.69 Mbits/sec
[127] 98.0-99.0 sec   695 KBytes  5.69 Mbits/sec
[127] 99.0-100.0 sec   701 KBytes  5.74 Mbits/sec
[127] 100.0-101.0 sec   708 KBytes  5.80 Mbits/sec
[127] 101.0-102.0 sec   701 KBytes  5.74 Mbits/sec
[127] 102.0-103.0 sec   708 KBytes  5.80 Mbits/sec
[127] 103.0-104.0 sec   708 KBytes  5.80 Mbits/sec
[127] 104.0-105.0 sec   708 KBytes  5.80 Mbits/sec
[127] 105.0-106.0 sec   708 KBytes  5.80 Mbits/sec
[127] 106.0-107.0 sec   708 KBytes  5.80 Mbits/sec
[127] 107.0-108.0 sec   708 KBytes  5.80 Mbits/sec
[127] 108.0-109.0 sec   701 KBytes  5.74 Mbits/sec
[127] 109.0-110.0 sec   701 KBytes  5.74 Mbits/sec
[127] 110.0-111.0 sec   695 KBytes  5.69 Mbits/sec
```

```
[128] 89.0-90.0 sec   700 KBytes  5.74 Mbits/sec
[128] 90.0-91.0 sec   700 KBytes  5.74 Mbits/sec
[128] 91.0-92.0 sec   700 KBytes  5.73 Mbits/sec
[128] 92.0-93.0 sec   699 KBytes  5.73 Mbits/sec
[128] 93.0-94.0 sec   699 KBytes  5.73 Mbits/sec
[128] 94.0-95.0 sec   701 KBytes  5.74 Mbits/sec
[128] 95.0-96.0 sec   700 KBytes  5.74 Mbits/sec
[128] 96.0-97.0 sec   700 KBytes  5.74 Mbits/sec
[128] 97.0-98.0 sec   700 KBytes  5.74 Mbits/sec
[128] 98.0-99.0 sec   695 KBytes  5.69 Mbits/sec
[128] 99.0-100.0 sec   699 KBytes  5.73 Mbits/sec
[128] 100.0-101.0 sec   700 KBytes  5.74 Mbits/sec
[128] 101.0-102.0 sec   697 KBytes  5.71 Mbits/sec
[128] 102.0-103.0 sec   698 KBytes  5.71 Mbits/sec
[128] 103.0-104.0 sec   700 KBytes  5.74 Mbits/sec
[128] 104.0-105.0 sec   700 KBytes  5.74 Mbits/sec
[128] 105.0-106.0 sec   700 KBytes  5.74 Mbits/sec
[128] 106.0-107.0 sec   699 KBytes  5.73 Mbits/sec
[128] 107.0-108.0 sec   699 KBytes  5.73 Mbits/sec
[128] 108.0-109.0 sec   700 KBytes  5.74 Mbits/sec
[128] 109.0-110.0 sec   696 KBytes  5.70 Mbits/sec
[128] 110.0-111.0 sec   698 KBytes  5.72 Mbits/sec
[128] 111.0-112.0 sec   700 KBytes  5.74 Mbits/sec
```

```
[127] 55.0-56.0 sec  0.00 Bytes  0.00 bits/sec
[127] 56.0-57.0 sec  0.00 Bytes  0.00 bits/sec
[127] 57.0-58.0 sec  0.00 Bytes  0.00 bits/sec
[127] 58.0-59.0 sec  0.00 Bytes  0.00 bits/sec
[127] 59.0-60.0 sec  0.00 Bytes  0.00 bits/sec
[127] 60.0-61.0 sec  0.00 Bytes  0.00 bits/sec
[127] 61.0-62.0 sec  0.00 Bytes  0.00 bits/sec
[127] 62.0-63.0 sec  0.00 Bytes  0.00 bits/sec
[127] 63.0-64.0 sec  0.00 Bytes  0.00 bits/sec
[127] 64.0-65.0 sec  0.00 Bytes  0.00 bits/sec
[127] 65.0-66.0 sec  0.00 Bytes  0.00 bits/sec
[127] 66.0-67.0 sec  0.00 Bytes  0.00 bits/sec
[127] 67.0-68.0 sec  0.00 Bytes  0.00 bits/sec
[127] 68.0-69.0 sec  0.00 Bytes  0.00 bits/sec
[127] 69.0-70.0 sec  0.00 Bytes  0.00 bits/sec
[127] 70.0-71.0 sec  0.00 Bytes  0.00 bits/sec
[127] 71.0-72.0 sec  0.00 Bytes  0.00 bits/sec
[127] 72.0-73.0 sec  0.00 Bytes  0.00 bits/sec
[127] 73.0-74.0 sec  0.00 Bytes  0.00 bits/sec
[127] 74.0-75.0 sec  0.00 Bytes  0.00 bits/sec
[127] 75.0-76.0 sec  0.00 Bytes  0.00 bits/sec
[127] 76.0-77.0 sec  0.00 Bytes  0.00 bits/sec
[127] 77.0-78.0 sec  0.00 Bytes  0.00 bits/sec
```

Figure 4.2-3: H5 Connection blocked

In the commands 250 represents the buffer length to read/write. The default datagram size for UDP is 1470 bytes, so it is lowered to 1024 bytes. Figure 4.2-2 shows a comparison of the throughput obtained before and after fragmentation, where the window size is 256K for (i) and 1024K for (ii). 250 bytes is the lowered packet size and one of the HTTP connections (H_5) is blocked at 55 secs (see Figure 4.2-3) and for the remaining time interval the bandwidth available to H_4 and H_3 connections.

4.2.1.2 Packet Loss

Simulation results in Figure 4.2-4, shows that the datagram loss is at all time maintained below 4% video over a duration of 120s. The datagram adds an IP header to the packet when processed by the network layer and transforms into an IP packet. The packet loss before and after fragmentation is shown in Figure 4.2-5. In Figure 4.2-5 (i) the loss is maintained below 4% for both voice and video sessions and for (ii) the loss for video ranges between 10-14% with peak variation of up to 16%. For VoIP traffic sessions, the datagram loss is not much important as compared to video sessions as per the Table 4. This is because the only packet loss for VoIP session is caused by L2 bit error or network disaster. The loss in video happen as TCP synchronisation for a long time repeatedly occupy the node buffer. Therefore, UDP packets are dropped successively even when the node is not fully packet filled.

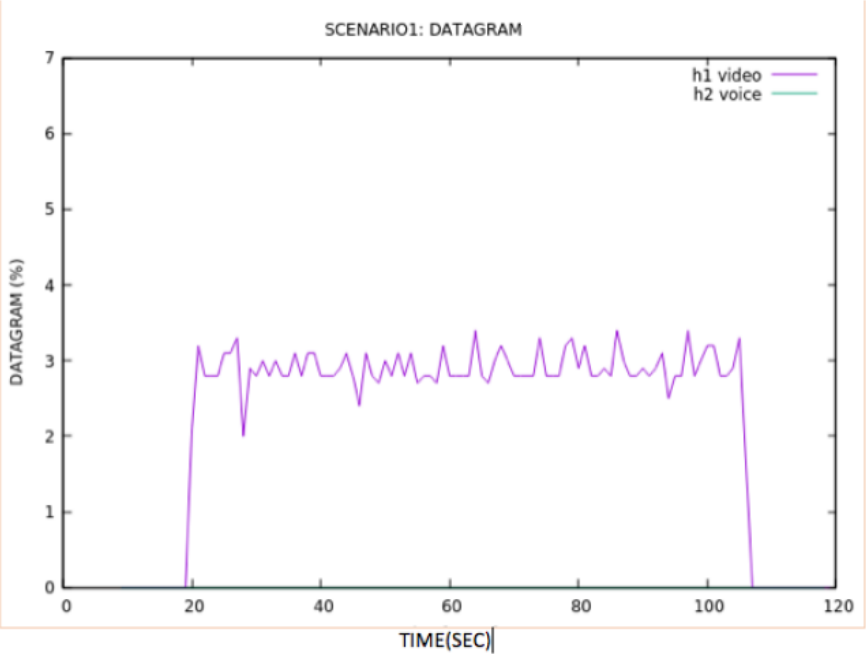
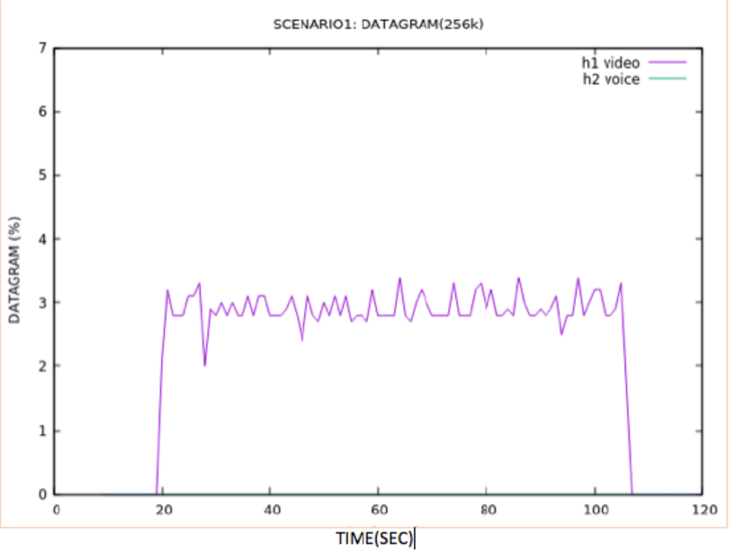
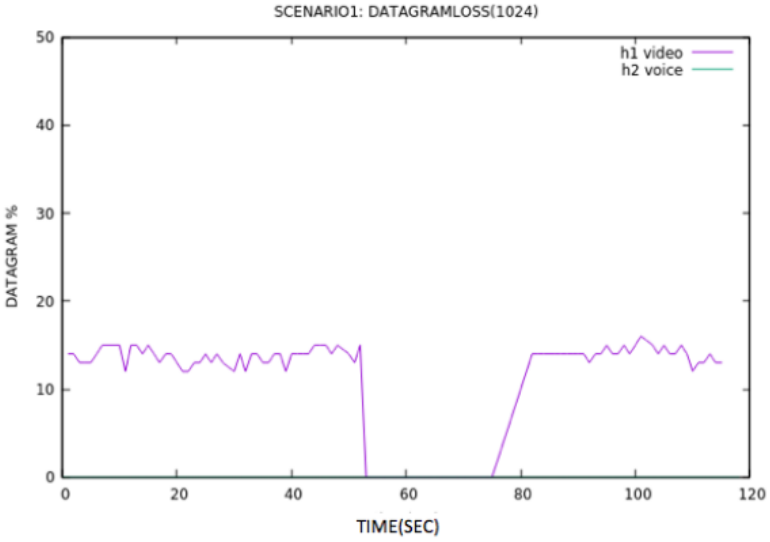


Figure 4.2-4: Packet loss in Scenario 1

Hence, the video session suffers from higher packet loss due to TCP synchronisation, even when the UDP transmission rates are low after fragmentation. TCP does not suffer any loss before and after fragmentation as it can share all the available bandwidth and consume the available bandwidth. Hence, fragmentation will not help in reducing UDP datagram loss.



(i)



(ii)

Figure 4.2-5: (i) Packet loss before fragmentation, (ii) Packet loss after fragmentation.

4.2.1.3 Packet Delay Variation

Packet delay variation is the delay in receiving packets in the network due to network congestion, inefficient routing, and improper queueing. Jitter/Packet delay variation also defines the latency variation in a network. Jitter can be managed using a jitter buffer. Jitter can be defined more accurately as the mean deviation of difference in packet spacing at the receiver for a pair of packets compared to the sender, which means it is equal to the difference between receivers' clock at the time of interval and packets timestamp. *Iperf command* is used to calculate the jitter based on the formula defined in [91].

$$D(i,j) = (Ri - Rj) - (Sj - Si) = (Rj - Sj) - (Ri - Si)$$

Equation 1

In Eq. 1, *D* is the delay for two packets *i* and *j*, S_i is the timestamp from packet *i* and R_i is the time of arrival in the timestamp units for packet i.

$$J(i) = J(i-1) + \frac{|D(i\ -\ 1,i)| - J(i-1)}{16}$$

Equation 2

For each data packet *i* received, the interval arrival jitter $J(i)$ in equation 2 can be calculated using the difference *D* between the packet itself and the previous packet *i*-1 according to the Eq. 2. The $(\frac{1}{16})$ parameter in Eq. 2 while maintaining an excellent rate of convergence provides better nose reduction ratio. The jitter is maintained below 0.2ms on an average for voice and 0.66ms for video. In Figure 4.2-6, the jitter for voice session varies between 0.012 ms and 0.2 ms, whereas for video session it reaches the maximum peak of 0.7 ms and a minimum value of 0.02 ms.

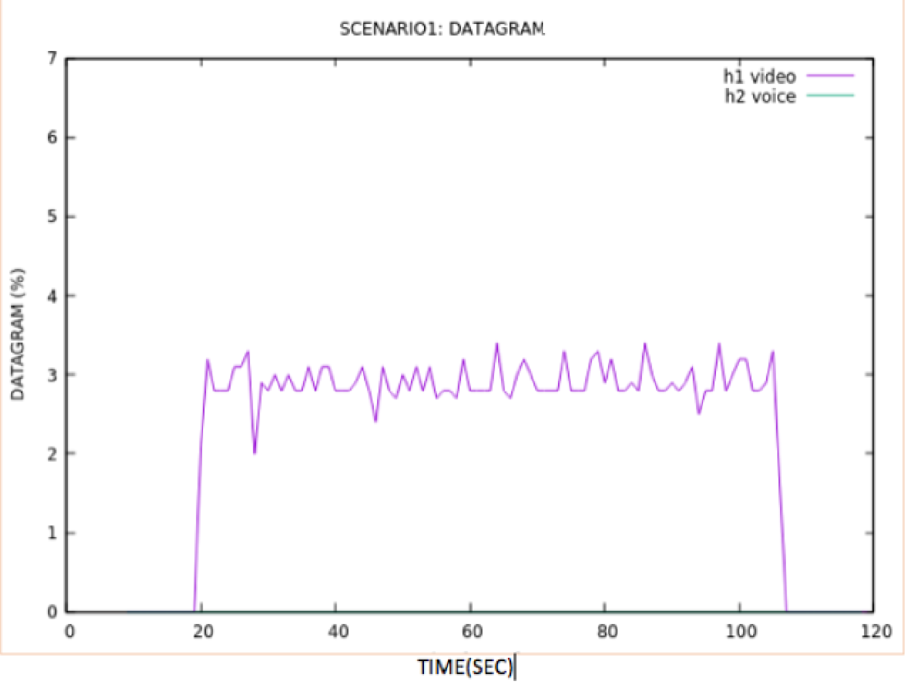
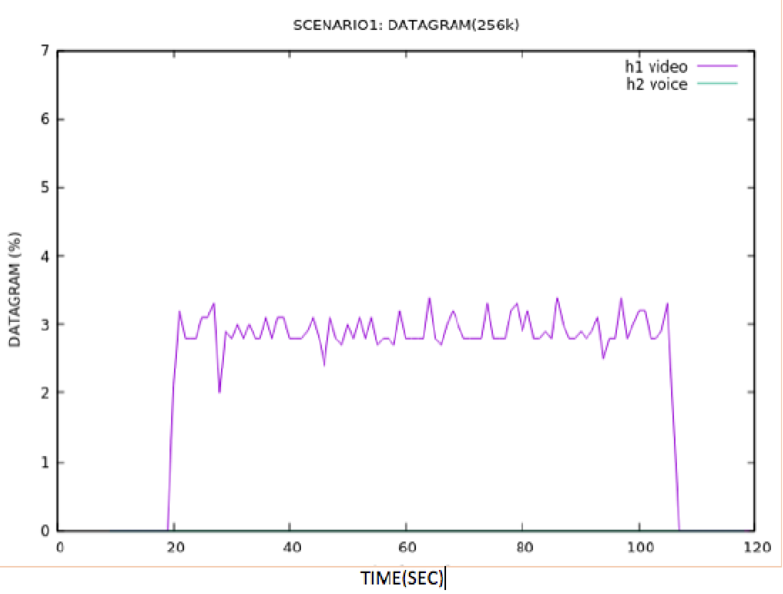
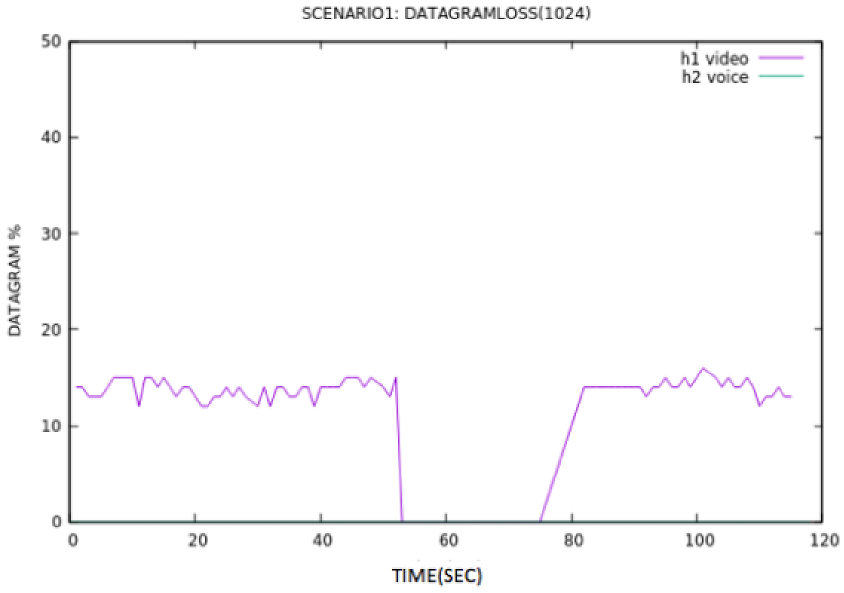


Figure 4.2-6: Jitter for Scenario 1

In Figure 4.2-7 (i) the jitter before fragmentation for voice ranges between 0.0 ms to 0.2 ms and video ranges from 0.1ms to 0.66ms and after fragmentation the packet delay variation varies between 0.0 to 0.2ms for voice and 0.1 to 7ms.



(i)



(ii)

Figure 4.2-7: (i) Jitter before fragmentation (ii) Jitter after fragmentation

4.2.1.4 Round Trip Time (RTT)

The round-trip time is the minimum time between arrival of acknowledgement for a data segment and sending of the same data segment. For a TCP connection, there will be only one data segment per round-trip time as it must wait for ACK message before transmitting the next data segment. Eq. 3 defines the relationship between roundtrip time delay and throughput for TCP session [92].

$$max(Throughput) = \frac{(W.S)}{R}$$

Equation 3

Where *R* represents the round-trip delay, W and S are the window size and maximum segment size respectively. So, the RTT can be defined as shown in Eq. 4 and the one-way delay to the receiver, *D*, is the one-way delay to the receiver and defined as shown in Eq. 5.

$$R = \frac{(W.S)}{\max(Throughput)}$$

Equation 4

$$D = \sqrt{\frac{R}{2}}$$

Equation 5

4.2.1.5 Probability of Loss

The relation between loss probability *p* and W representing window size is established in Eq. 6 [91]

$$W = \sqrt{\frac{8}{3.p}}$$

Equation 6

The average throughput for packets in a TCP connection can be expressed as shown in Eq. 7

$$W = \frac{3.W.S}{4.R}$$

Equation 7

Combining Eq 6 and Eq 7, the average throughput *T* can be derived as shown in Eq. 8

$$T = \frac{\sqrt{\frac{3S}{2}}}{R\sqrt{p}}$$

Equation 8

In Eq. 8, S represents the maximum segment size, K is a constant value $\sqrt{\frac{3}{2}}$, the round-trip time is R and the loss rate is defined by *p*. The loss probability *p* can be calculated from Eq. 9 as follows:

$$p = \frac{\frac{3}{2}S^2}{R^2T^2}$$

Equation 9

The values for round-trip time and loss before and after fragmentation are presented in Table 8. For TCP connection note the window size is *W*=128K before fragmentation and 256K after fragmentation, maximum segment size is *S*=8*KB (kilo bytes)* before fragmentation and 250 bytes after fragmentation.

Hosts	RTT before fragmentation (ms)	RTT after fragmentation (%)	Loss before fragmentation (s)	Loss after fragmentation (%)
H_3	170	1.73	$8.799*10^{-9}$	$8.799*10^{-5}$
H_4	170	1.73	$8.799*10^{-9}$	$8.799*10^{-5}$
H_5	170	1.73	$8.799*10^{-9}$	$8.799*10^{-5}$

Table 9: Round Trip Time and Loss for TCP connection

4.2.1.6 Routing-Shortest Path Calculation

The shortest path is returned by the Dijkstra algorithm presented in Section 3.7.2.3.2. Table 10 presents the paths between core and edge switches. The Dijkstra algorithm in the responsible for S/PGW-U selection. The path calculation is shown when all hosts are connected to server H_{45} during all the traffic sessions. The service class for video, voice and HTTP sessions are Assured Forwarding (ToS is 136), Expedited Forwarding (ToS is 184), and Best Effort (ToS is 0) respectively.

Hosts	**Edge Switch**	**Aggregation Switch**	**Core Switch**	**Session**	**Data Rate**
H_1	S_7	S_4	S_1	Video	20 Mbps
H_1	S_7	S_4	S_1	Voice	8 Mbps
H_3	S_7	S_4	S_1	HTTP	6 Mbps
H_4	S_7	S_4	S_1	HTTP	6 Mbps
H_5	S_7	S_4	S_1	HTTP	6 Mbps
H_6	S_8	S_5	S_1	Video	20 Mbps
H_7	S_8	S_5	S_1	Voice	8 Mbps
H_8	S_8	S_3	S_1	HTTP	6 Mbps
H_9	S_8	S_5	S_1	HTTP	6 Mbps
H_{10}	S_9	S_5	S_1	HTTP	6 Mbps
H_{11}	S_9	S_3	S_1	Video	20 Mbps
H_{12}	S_9	S_3	S_1	Voice	8 Mbps
H_{13}	S_9	S_3	S_1	HTTP	6 Mbps
H_{14}	S_9	S_3	S_1	HTTP	6 Mbps
H_{15}	S_9	S_3	S_1	HTTP	6 Mbps
H_{16}	S_{10}	S_6	S_1	Video	20 Mbps
H_{17}	S_{10}	S_6	S_1	Voice	8 Mbps
H_{18}	S_{10}	S_4	S_1	HTTP	6 Mbps
H_{19}	S_{10}	S_4	S_1	HTTP	6 Mbps
H_{20}	S_{10}	S_4	S_1	HTTP	6 Mbps
H_{21}	S_{11}	S_6	S_1	Video	20 Mbps
H_{22}	S_{11}	S_5	S_1	Voice	8 Mbps
H_{23}	S_{11}	S_6	S_1	HTTP	6 Mbps
H_{24}	S_{11}	S_6	S_1	HTTP	6 Mbps
H_{25}	S_{11}	S_6	S_1	HTTP	6 Mbps
H_{26}	S_{12}	S_4	S_1	Video	20 Mbps
H_{27}	S_{12}	S_5	S_1	Voice	8 Mbps
H_{28}	S_{12}	S_4	S_1	HTTP	6 Mbps
H_{29}	S_{12}	S_4	S_1	HTTP	6 Mbps
H_{30}	S_{12}	S_4	S_1	HTTP	6 Mbps
H_{31}	S_{13}	S_6	S_1	Video	20 Mbps
H_{32}	S_{13}	S_5	S_1	Voice	8 Mbps

H_{33}	S_{13}	S_6	S_1	HTTP	6 Mbps
H_{34}	S_{13}	S_6	S_1	HTTP	6 Mbps
H_{35}	S_{13}	S_6	S_1	HTTP	6 Mbps
H_{36}	S_{14}	S_5	S_1	Video	20 Mbps
H_{37}	S_{14}	S_3	S_1	Voice	8 Mbps
H_{38}	S_{14}	S_5	S_1	HTTP	6 Mbps
H_{39}	S_{14}	S_5	S_1	HTTP	6 Mbps
H_{40}	S_{14}	S_5	S_1	HTTP	6 Mbps

Table 10: Routing in between core and edge switches

4.2.1.7 Summary

In Scenario 1- Single Policy Installation section 4.2.1, the results obtained are maintained as per the predefined rate limit for each traffic session such as video has 20Mbps, voice has 8Mbps and 6Mbps for 3 HTTP connections. The 6Mbps is utilised by 3 HTTP connections during the lifecycle of the session using the flow control procedure. Therefore, the bandwidth consumption is majorly taken up by HTTP connections and effects the video and voice connections (UDP connections). Even, if the transmission rate is lowered for the video and voice connections or the traffic streams are fragmented, the HTTP connections still rapidly consume the bandwidth as the TCP connections share a bottleneck node. This is the reason the packet loss for UDP connections cannot be improved even after fragmentation. The QoE for users for video and voice streaming applications are affected as the UDP packet loss determines the end-to-end user experience. The jitter and delay in UDP streaming is also dependant on the loss percentage. The voice applications need to be transmitted in real time, so the packets cannot be retransmitted. The UDP applications are transmitted irrespective of the packet loss, which means the client transmits over any bandwidth without datagram loss concern. At receiver, for UDP no multicasting, sequence numbers or resynchronisation is used for UDP sessions, also it does not require any acknowledgement message. On the other hand, for TCP sessions the packets are re-transmitted if there is any loss. From the results, it is evident that the jitter and percentage of loss are within the prespecified parameters in Section 4.1.

4.2.2 Scenario 2 – Multiple Policy Installation

Multiple policies are sent to the servers from the corresponding host in Scenario 2. Table 11 presents the session corresponding to five hosts: H_1, H_2, H_3, H_4, and H_5. The Round-Robin mechanism is used for policy installation in the paths, based on Session or Host type. For both Host type and Session type the window size is 1024 K for TCP sessions and 1047 bytes for UDP buffer size.

The Host type flow characteristics are shown in Table 12. H_{45} and H_{46} have the traffic equally distributed among themselves. The Session type characteristic of each flow is shown in Table 13, where the session type determines the traffic distribution. H_{45} receives the flows for the first session initiation and H_{46} receives the second traffic. For both Host type and Session type, the service class for video, voice and HTTP sessions are AF, E, and BE. The ToS value is 136, 0 and 184 for video, HTTP, and voice sessions.

Hosts	Session
H_1	1 video 1 voice 1 HTTP
H_2	1 video 1 HTTP
H_3, H_4, H_5	2 HTTP

Table 11: Scenario 2 sessions

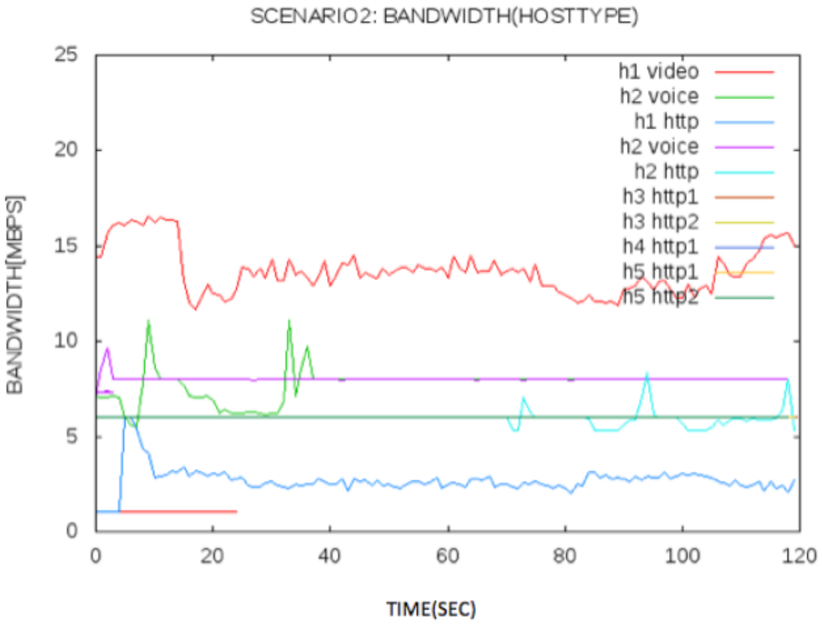
Host	Server	Session	Protocol	DSCP	Queue	Data Rate
H_1	H_{45}	1 Video	UDP	34	0	20 Mbps
H_1	H_{46}	1 Voice	UDP	46	1	8 Mbps
H_1	H_{45}	1 HTTP	TCP	0	2	6 Mbps
H_2	H_{46}	1 Voice	UDP	46	1	8 Mbps
H_2	H_{45}	1 HTTP	UDP	0	2	6 Mbps
H_3	H_{46}	1 HTTP	TCP	0	2	6 Mbps
H_3	H_{45}	1 HTTP	TCP	0	2	6 Mbps
H_4	H_{46}	1 HTTP	TCP	0	2	6 Mbps
H_4	H_{45}	1 HTTP	TCP	0	2	6 Mbps
H_5	H_{46}	1 HTTP	TCP	0	2	6 Mbps
H_5	H_{45}	1 HTTP	TCP	0	2	6 Mbps

Table 12: Scenario 2 Host-Type

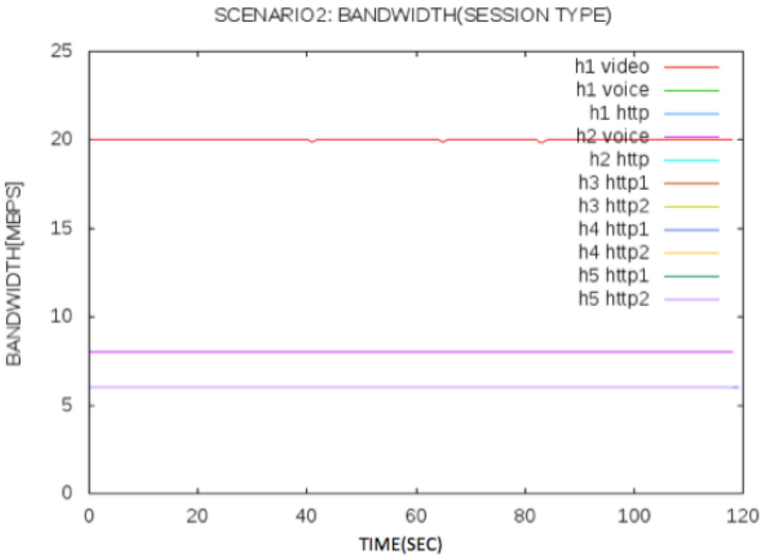
Host	Server	Session	Protocol	DSCP	Queue	Data Rate
H_1	H_{45}	1 Video	UDP	34	0	20 Mbps
H_1	H_{45}	1 Voice	UDP	46	1	8 Mbps
H_1	H_{45}	1 HTTP	TCP	0	2	6 Mbps
H_2	H_{46}	1 Voice	UDP	46	1	8 Mbps
H_2	H_{45}	1 HTTP	UDP	0	2	6 Mbps
H_3	H_{45}	1 HTTP	TCP	0	2	6 Mbps
H_3	H_{46}	1 HTTP	TCP	0	2	6 Mbps
H_4	H_{46}	1 HTTP	TCP	0	2	6 Mbps
H_4	H_{45}	1 HTTP	TCP	0	2	6 Mbps
H_5	H_{45}	1 HTTP	TCP	0	2	6 Mbps
H_5	H_{46}	1 HTTP	TCP	0	2	6 Mbps

Table 13: Scenario 2 Session-Type

In Figure 4.2-8 (i), there are 8 HTTP connections and each of the connections takes approximately 6 Mbps from the server. The bandwidth for video is capped to 17 Mbps and for two voice the session starts at 8 Mbps. Note that the window size for TCP is set to 1024 K and the buffer size is not configured for UDP in Scenario 2. In (ii), 20 Mbps is bandwidth assigned to the video traffic, one policy is executing one at a time so there is no overhead and the maximum bandwidth is attained, and the voice session is limited to 8Mbps.



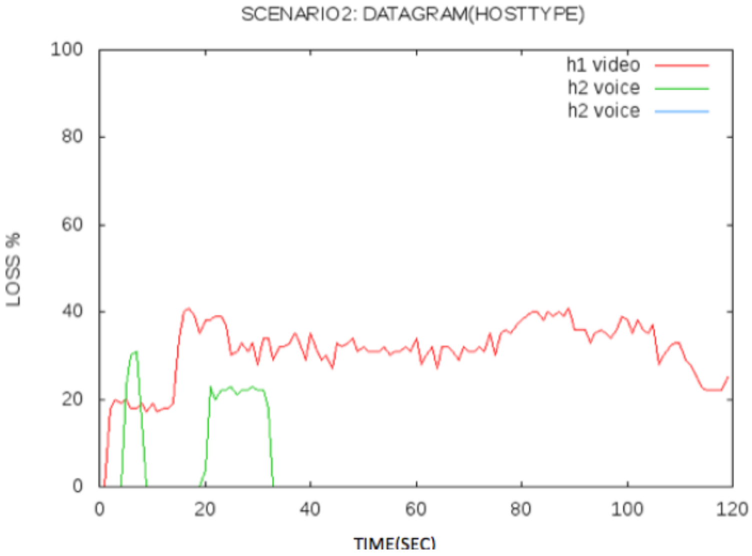
(i)



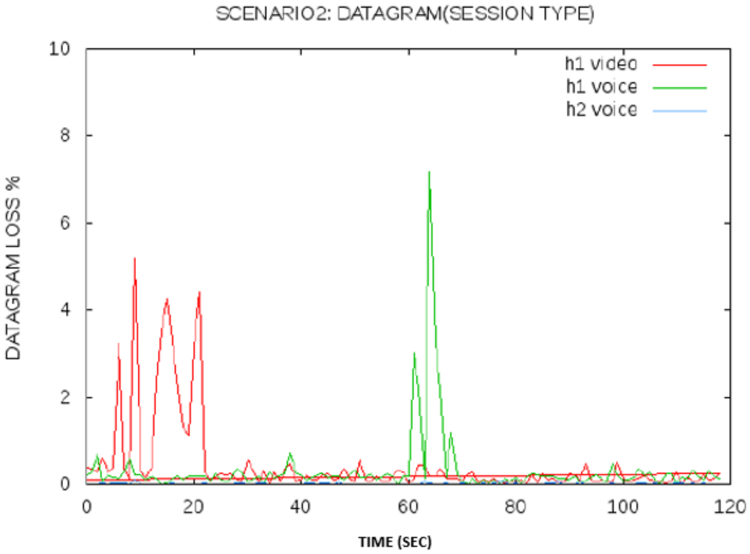
(ii)

Figure 4.2-8: (i) Round-Robin Host Type Throughput for Scenario 2, (ii) Round-Robin Session Type Throughput for Scenario 2

In Figure 4.2-9, the average loss for the two voice sessions suffer less loss in Host type compared to Session type. However, in Host type there are some peak variations and represent the data rate changes at the same moment. At 15ms the video sessions compared to other sessions experience more loss. On the contrary, for Session type the H_1 voice connection loss is higher than the voice traffic sent from H_2. The loss is a result of the TCP synchronisation due to the presence of TCP stream. Secondly the high data rate demand for video session is a reason for the loss. From the plots 4.2-9 (i) and (ii) for Round-Robin Host type the load balancing is achieved better compared to Session type.



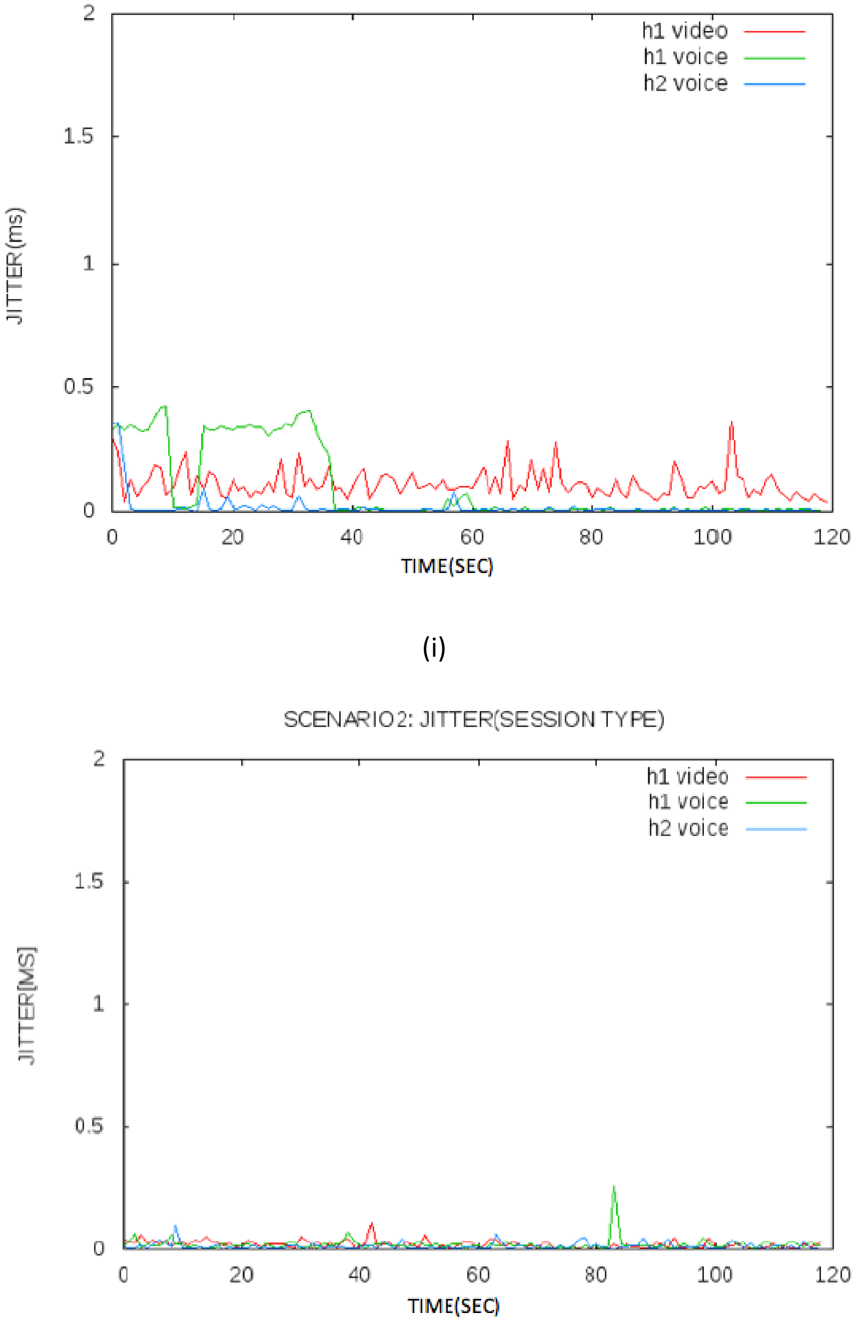
(i)



(ii)

Figure 4.2-9: (i)Packet Loss for Host type in Scenario 2, (ii) Pack loss for Session type in Scenario 2

In Figure 4.2-10 (i), the jitter value is minimum for both voice and video connections and always maintained below 0.5ms with few variations which is within the specifications mentioned in [93]. The jitter value in Session type is much smaller (maintained below 0.2ms) than host type, with higher spikes shown at 85ms.



(i)

(ii)

Figure 4.2-10: (i) Jitter for Host-type in Scenario 2, (ii) Jitter for Session type in Scenario 2

4.2.2.1 Path Calculation

All aggregation switches are directed to server H_{45} and H_{46} in Scenario 1, where for 40 hosts single policy installation is done. In Scenario 2, the path returned for multiple policy installation on 5 hosts in Host type and Session type is shown in Table 14 and Table 15. At the aggregation switch level, the path returned for Host type and Session type load balancing varies between switch S_3 and switch S_5. The load is distributed between H_{45} and H_{46} in Round-Robin Host type. The return path at aggregation level alternates between S_3 and S_5.

Table 15 confirms that for Session type, the return path for all H_1 sessions follow same path (i.e, S_7, S_3 and S_1) and for all H_2 sessions the traffic is sent via S_2, S_5 and S_7. Unlike the Host type, the path is reliant on server chosen, so S_3 is selected to route traffic sent to H_{45} and S_5 is selected to route traffic sent to H_{46}. The difference between Host and Session type is the traffic distribution pattern such as similar traffic type and server selected.

Host	**Server**	**Edge Switch**	**Aggregation Switch**	**Core Switch**	**Session**	**Bandwidth**
H_1	H_{45}	S_7	S_3	S_1	1 Video	20 Mbps
H_1	H_{46}	S_7	S_5	S_2	1 Voice	8 Mbps
H_1	H_{45}	S_7	S_3	S_1	1 HTTP	6 Mbps
H_2	H_{46}	S_7	S_5	S_2	1 Voice	6 Mbps
H_2	H_{45}	S_7	S_5	S_1	1 HTTP	6 Mbps
H_3	H_{46}	S_7	S_3	S_2	1 HTTP	6 Mbps
H_3	H_{45}	S_7	S_5	S_1	1 HTTP	6 Mbps
H_4	H_{46}	S_7	S_3	S_2	1 HTTP	6 Mbps
H_4	H_{45}	S_7	S_5	S_1	1 HTTP	6 Mbps
H_5	H_{46}	S_7	S_3	S_2	1 HTTP	6 Mbps
H_5	H_{45}	S_7	S_5	S_1	1 HTTP	6 Mbps

Table 14: Routing in between edge and core switches (Host Type)

Host	**Server**	**Edge Switch**	**Aggregation Switch**	**Core Switch**	**Session**	**Bandwidth**
H_1	H_{45}	S_7	S_3	S_1	1 Video	20 Mbps
H_1	H_{46}	S_7	S_5	S_2	1 Voice	8 Mbps
H_1	H_{45}	S_7	S_3	S_1	1 HTTP	6 Mbps
H_2	H_{46}	S_7	S_5	S_2	1 Voice	6 Mbps
H_2	H_{45}	S_7	S_5	S_1	1 HTTP	6 Mbps
H_3	H_{46}	S_7	S_3	S_2	1 HTTP	6 Mbps
H_3	H_{45}	S_7	S_5	S_1	1 HTTP	6 Mbps
H_4	H_{46}	S_7	S_3	S_2	1 HTTP	6 Mbps
H_4	H_{45}	S_7	S_5	S_1	1 HTTP	6 Mbps

H_5	H_{46}	S_7	S_3	S_2	1 HTTP	6 Mbps
H_5	H_{45}	S_7	S_5	S_1	1 HTTP	6 Mbps

Table 15: Routing in between edge and core switched (Session Type)

4.2.2.1 Summary

In Scenario 2, either Host type or Session type instantiation based on Round-Robin fashion is used for policy enforcement on paths. In Host-type instantiation, client H_1 sends 1 video session and 1 VoIP session to server H_{45} and $_{46}$. Server H_{45} receives all the UDP traffic in Session type. For example, in Host type H_{45} gets the first VoIP session transmitted from client H_1 and second VoIP session from client H_2 is sent to server H_{46}. The bandwidth during second session is divided between video and voice sessions, even when it is more robust. The packet loss performance is better in first setting. TCP streams are responsible for packet loss in both configurations- Host type and Session type due to TCP flow control mechanism. In case of UDP application better interface to the network is achievable since UDP has no connection set up delays, retransmission, and flow controls. The packet loss is extremely low for VoIP connections connected to server H_{46} in Figure 4.2-9 (ii). On the other hand, for VoIP connection connected to serve H_{45} are subjected to more packet loss as the bandwidth is shared with video connections leading to more number of packets dropped. The throughput variations also contribute to packet loss. In Figure 4.2-10 (ii), the jitter value is very small owing to the throughput stability. During the lifecycle of one session in a Session type same type of traffic is never consequently sent to the same server. The load balancing alternates between two switches (S_3, S_5) at aggregation level, in terms of path returned.

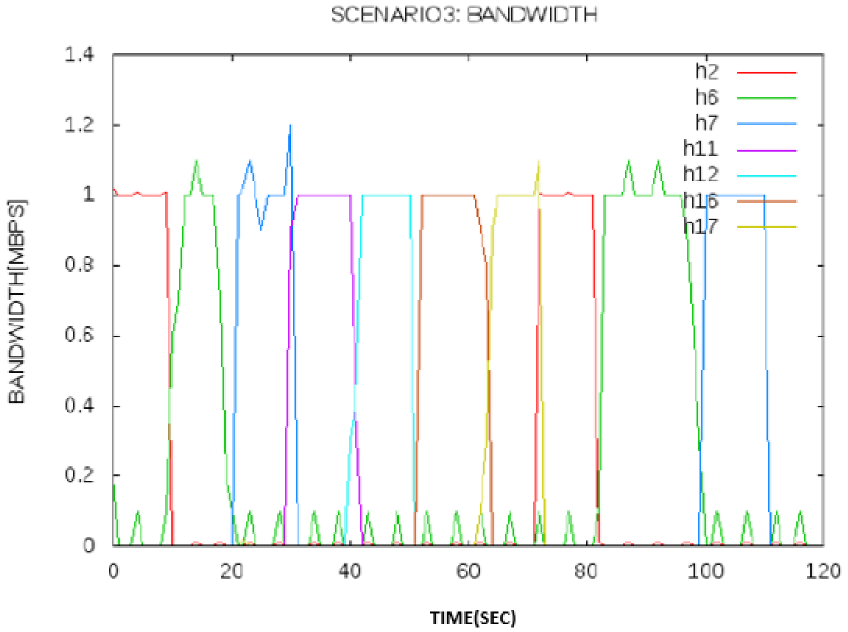
4.2.3 Scenario 3- Load Balancing

Two SDN controllers namely Floodlight and OpenDaylight is used for simulation in Scenario 3. Both the controller's architecture and functionalities have been discussed in section 3.7, Chapter 3. Table 16, presents the configuration for both controllers in Figure 4.2-11. The bandwidth is set to 1 Mbps for both Floodlight and OpenDaylight controller.

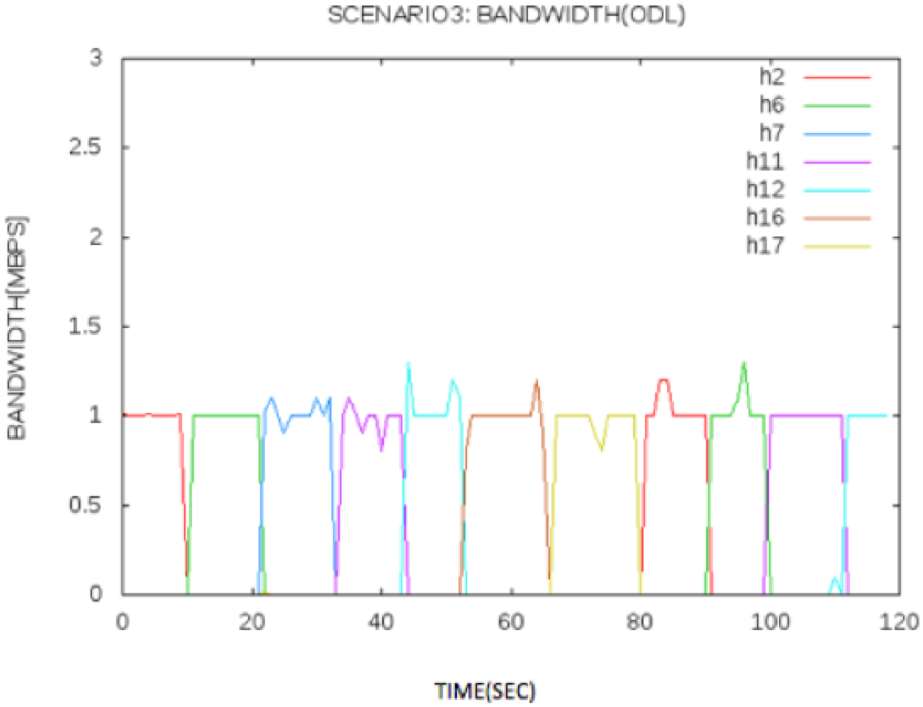
Host	**TCP VIP**	**UDP VIP**	**Bandwidth**
H_1	10.0.0.200	10.0.0.100	1 Mbps

Table 16: Scenario 3 Configuration

The Virtual IP (VIP) contains the client (H_1) along with other pool of servers (H_{17}, H_{16}, H_{12}, H_{11}, H_7, H_6, H_2). Both the client and server cannot run this at the same time, it will connect after skipping connections repeatedly. The client connects in a chain pattern to each server in a network using either Floodlight or OpenDaylight controller during 1s. Figure 4.2-11 (i) and (ii) validate the round robin mechanism and for OpenDaylight controller the throughput performance is higher as compared to Floodlight controller because the throughput is more stable at 1Mbps. However, the Floodlight controller throughput performance undergoes more variations compared to former.

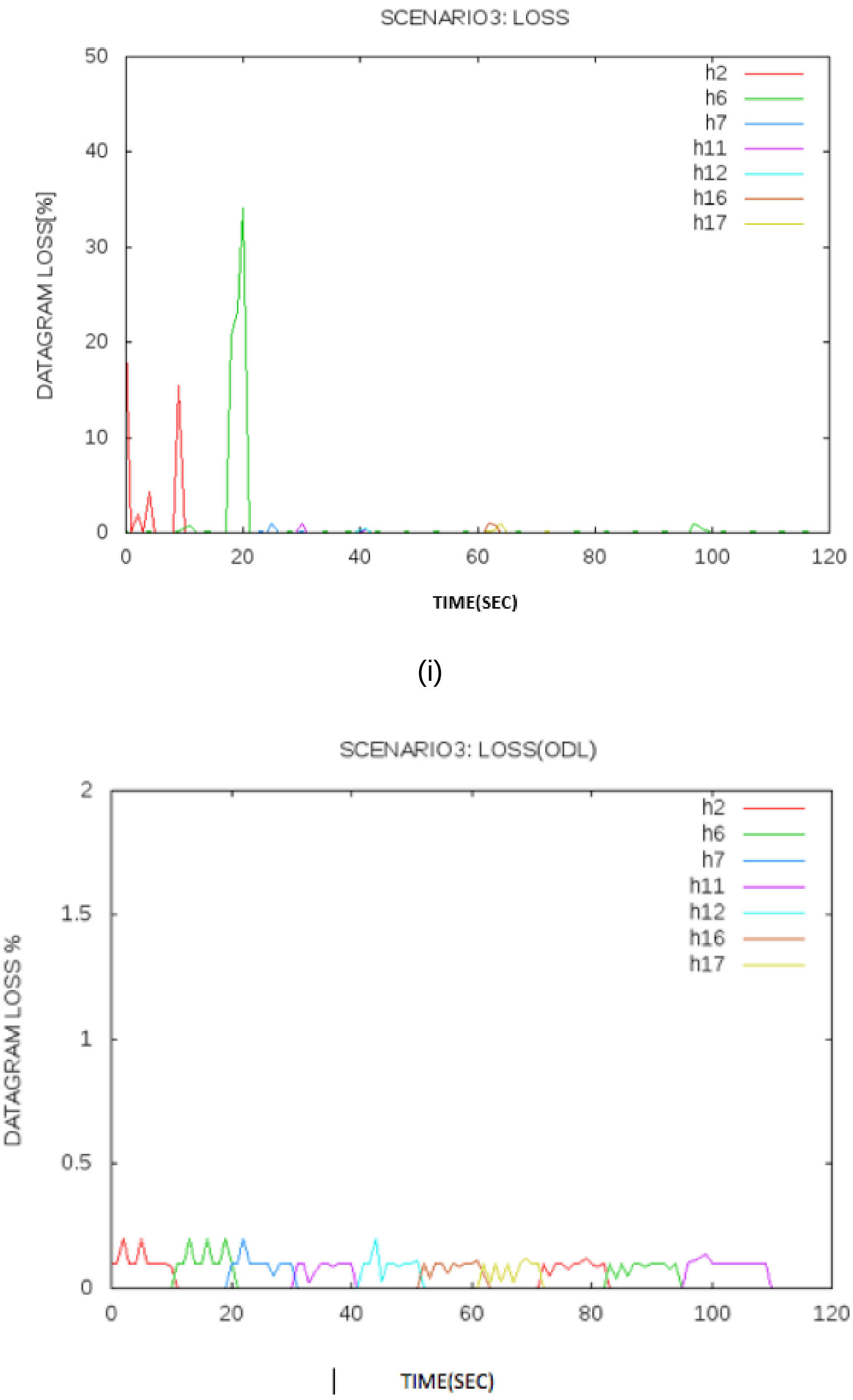


(i)



(ii)

Figure 4.2-11:(i) Throughput using Floodlight Controller, (ii) Throughput using OpenDaylight Controller

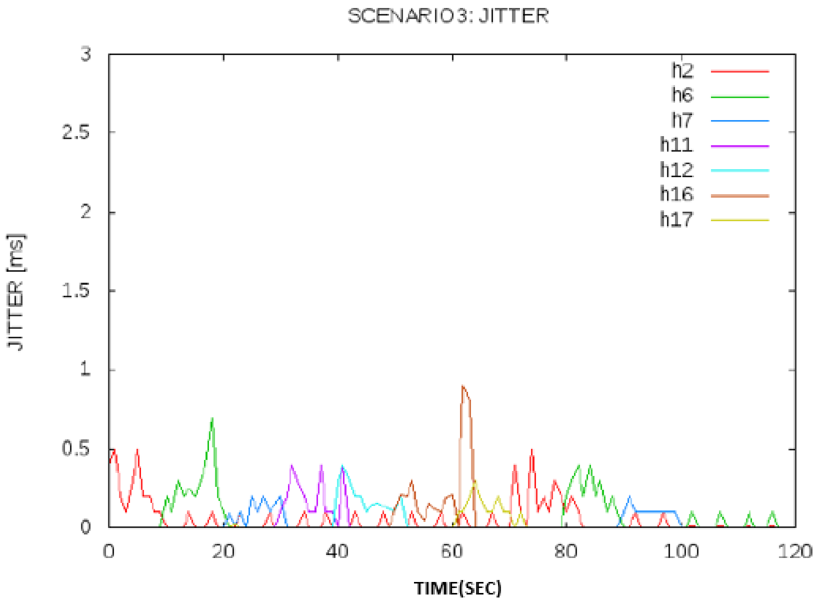


(i)

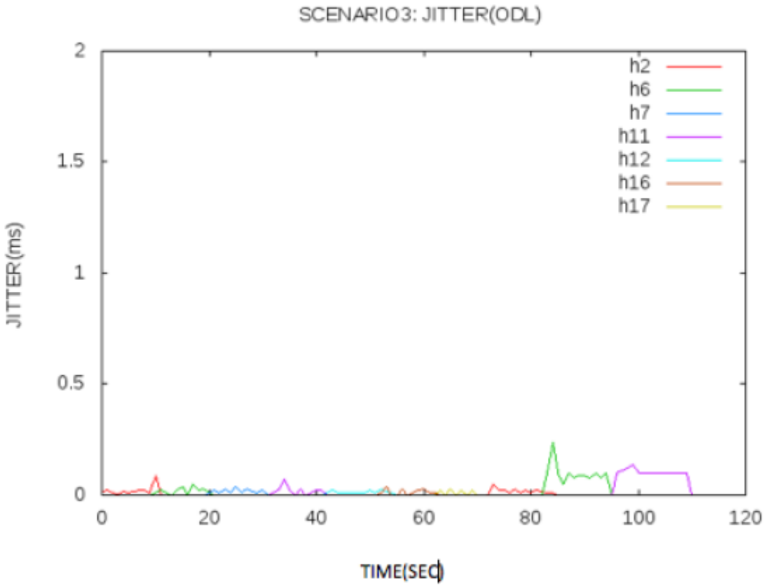
(ii)

Figure 4.2-12: (i) Datagram Loss for Floodlight Controller, (ii) Datagram Loss for OpenDaylight Controller

In Figure 4.2-12 (i, ii), the data loss for the UDP traffic (VoIP and voice sessions) which is sent to server in proposed network using a Floodlight or OpenDaylight controller. The datagram loss for Floodlight controller is much higher than OpenDaylight controller. At the early stage of each server connection, the data loss for Floodlight load balancing is as high as 15%, whereas for OpenDaylight controller the loss is maintained below 0.3%.



(i)



(ii)

Figure 4.2-13: (i) Jitter for Floodlight Controller, (ii) Jitter for OpenDaylight Controller

The jitter value for OpenDaylight is below 0.2ms and for the Floodlight Controller the packet delay variation is greater than 0.4 ms. For connection order as defined by the VIP pool sequence, the round robin chain is followed by server connection, where each server connection duration is 10s.

4.2.3.1 Summary

In Scenario 3, the two controllers: Floodlight and OpenDaylight are collated in terms of throughput, datagram loss and packet delay variation. The QoS module in Floodlight controller is not combined with the Round-Robin load balancing module since the address for load destination changes every time. Moreover, the source and VIP have no policy paths between them. The maximum queue value is set for all traffic when the QoS module is activated to install policies on the paths. Each test session for both OpenDaylight and Floodlight controllers lasts for 10s, where ODL-C test run for 10s when one server passed via the chain. The first server is excluded from the FL-C pool, whereas in case of ODL-C only if the host is identified in the VIP pool it can connect to itself. However, when the host connects to itself it does not receive any traffic from the server. The ODL version (Beryllium) used in the proposed network does not support loops in topologies, so the classic fat tree topology is only used for simulation purpose.

4.2.4 Scenario 4- Floodlight and OpenDaylight Controller Comparison

The Floodlight and OpenDaylight controller performance can be summarised in Table 16, derived from the simulation results in scenario 2 for the proposed topology in Figure 3.6-1, having 8 edge switches and more than one policy is installed for each of the edge switches.

Controller	Throughput	Packet Loss	Packet Delay Variation	Mode of Performance
Floodlight	Lower with more variations	Higher	Higher	1. Latency mode 2. Throughput mode
OpenDaylight	Higher	Lower	Lower	1. Latency mode

Table 17: Performance Comparison

The request processing time for OF controller under minimum load conditions is measured in latency mode, whereas the buffers receive as many requests from the switches depending on their capacity in throughput mode. ODL-C cannot perform in throughput mode as the CPU is very high and fails to process incoming messages in an efficient and stable manner. On the other hand, for latency mode the ODL-C performance is adaptive to the environment. The processing capacity of ODL-C is limited after a point of time as it functions only until a point of time. However, FL-C processing capacity is higher in comparison to ODL-C.

4.2.4.1 Performance Comparison

The performance of Floodlight and OpenDaylight Controller is outlined as follows depending on their performance and characteristic:

1. ***QoS Module***: FL-C has QoS module which is based class of service and policy paths. ODL-c has no QoS module.
2. ***Load Balancing***: Both FL-C and ODL-C support load balancing.
3. ***Bandwidth***: The bandwidth is limited to 1Mbps for FL-C, whereas ODL-C has no bandwidth restriction.
4. ***Topology***: FL-C supports topologies with loop configuration. ODL-C does not support any loop configuration.
5. ***Program***: Both are written in Java
6. ***User Interface***: For FL-C there is two supported UI: Web and Java. In ODL-C it is only Web based UI.
7. ***Processing***: FL-C can process 1M of flows per sec and it ranges up to 100000 flows per sec for ODL-C.

Both pros and cons are associated with both controllers (Floodlight and OpenDaylight) based on the application and use case environment. The load balancer performance is higher in FL-C for a maximum data rate of 1Mbps. On the other hand, the ODL-C processing stops when subjected to process millions of flows per second. FL-C is suitable for UDNs and ODL-C can be used for load balancing purposes for networks having thousands of switches with many hosts (100).

4.3 EMULATED NETWORK SUMMARY

The simulation environment corresponds to results generated in Scenario 1, 2 and 3 respectively. The only drawback of Mininet based emulated network is the restriction on CPU and available bandwidth for a single server. In the current research, the network resources are shared and need to be balanced among virtual switches and virtual hosts because the measurements are run on a single virtual machine. Hence, the data rate on the links are low with lower performance as compared to dedicated switching hardware.

CHAPTER 5

5 CONCLUSION AND FUTURE WORK

For the future mobile network architecture Cloud Radio Access Networks (CRAN), Software Defined Networking (SDN) and Network Function Virtualisation are the most promising enablers. The current mobile networks face issues related to network provisioning, expenditure, deployment complexity which limit the evolution towards 5G networks. On the other hand, SDN and NFV based networks can bring flexibility in programming and managing network functions as per network administrator requirements. New capabilities such as intelligent network resource utilisation, traffic steering and reduced cost pressure are added to the network so that it can provide services to the exponentially increasing number of users in the network.

In this research, the work focuses on investigating which parts of the mobile network (both access and core network) can be virtualised and made programmable under SDN control, as well how can SDN technology be integrated with legacy network. The CRAN architecture is considered for this work as it is the principal part of the next 5G mobile communication networks owing to its features discussed in Section 2.3. At the core site, Evolved Packet Core is a suitable candidate for NFV and SDN technology implementation. The author successfully realised a programmable/configurable mobile network system by adopting SDN and NFV features to provide improved traffic management and optimisation of Quality of Service (QoS) for 5G. The novel SDN based virtualized mobile network is implemented by reconfiguring SDN controller, CRAN and EPC network:

- Reconfigure the SDN controller: created and implemented four new modules - Mininet Queues, Add Custom Policies Scenario 1, Add Custom Policies Scenario 1, and Load Balancing to the current FL-C implementation to realise the proposed system (see Figure 3.7-3, modules in blue boxes). The load balancing module is implemented in the architecture top layer as an application. Added customised QoS module (QoS path, Circuit Pusher.py and QoSManager.py compose the QoS module) to enforce *QoS* state into the proposed network over the REST API. It performs the following actions: configuration of DiffServ type of service (ToS), maximum rate limit and minimum rate limit.
- Reconfigure CRAN - migrated BBU pool to EPC and deployed SDN controlled local controllers in BBU pool. The LC which is a SDN agent handles the aggregation of flows, also it is responsible to handle mobility when user attaches to another BBU from current one.
- Reconfigure EPC - add programmability feature by separating the control and data plane in the EPC via virtualisation. The EPC components Mobility Management Entity (MME) and Serving/Packet Data Network Gateway Control Plane (S/PGW-C) are controlled by the centrally placed SDN controller. The MME functionalities and the S1-MME interface is preserved.

The GPRS Tunneling Protocol User Plane is discarded in the proposed system so that OpenFlow (OF) switches to replace data plane, where the OF switches are controlled by a centralised SDN controller. The infrastructure needed to run NFV is offered by SDN. The routing of flows is by installing flows in the switches proactively by the SDN controller and the load balancing is also handled by the controller.

The proposed system preserved the 3GPP standard control plane functionalities in a way that the complexity in the data plane due to Tunnel Endpoint Identifiers (TEID) is reduced. Although there are many approaches which allow the GTP-U in the Open virtual switches, in this research the protocol is eliminated to decrease processing time in OF switches and overhead in the packet header.

Majority of the existing research approaches targeted the GTP-U encapsulation or use other protocols in both data and control plane. In the proposed mobile network architecture, the S1-MME interface which connects the BBU pool to the EPC is preserved as it includes heavy exchange of messages typically related to user authentication which is a very sensitive mechanism. The access network and EPC incorporates programming and smart managing capabilities of SDN technology to gain improved traffic engineering and support for QoS policies. The most attractive benefits of the SDN and NFV is the flexibility in load balancing to manage the cramming in the network nodes and routing of traffic flows in the network, which is not possible in current standard mobile networks and require long standardisation process before being realised.

The traffic is managed at the LC for a group of BBUs and RRHs or it can be managed by the centralised controller. The policy installation on the paths is done based on the service class which means traffic steering is based on the session type such as VoIP, video and HTTP having different data rate demands. The network is emulated using Mininet to validate the QoS procedure and load balancing mechanism. The QoS module and load balancing is implemented at the application layer on top of the SDN controller, which communicates with the modules using *Northbound* interface. The comparison between two SDN controllers is presented in Chapter 4. The simulation results obtained in terms of throughput, datagram loss and jitter are maintained with the requirements of session types. The emulation tool used (Mininet) add latency in controller and might affect the performance of the network. The comparison between two controllers show that the Floodlight is better in terms of performance compared to OpenDaylight for Ultra Dense Networks (UDN) having many users connected to the network. The OpenDaylight controller gets blocked after a point if it needs to process too many messages.

The proposed system can be improved to address the responsiveness to users depending on the application requirements and extend the OF protocol to the access network fronthaul (link between BBU and RRH) for application aware traffic steering. Therefore, due to the software programmability and elasticity any improvement in the network is possible, thus it can be concluded that SDN, NFV and CRAN are the principal options for future mobile networks (like 5G). Adopting the proposed system requires discarding tradition hardware with a fine defined strategy and follow multiple transition phases in the deployment.

5.1.1 Future Work

The research focuses on the EPC and CRAN virtualisation as well as QoS and load distribution use cases validation. However, the project can be extended at the access network part in the fronthaul and cell site. The algorithms used for QoS and load balancing can be improved in terms of scheduling queues dynamically based on the flow size. The shortest path selection can be changed in way that the weights on links are assigned based on the load information the switch receives in real time. The possibility to replace S1-MME interface with OpenFlow protocol needs to be investigated. Moreover, the benefits of moving all EPC elements to a control plane running on top of the controller needs to be analysed.

Since SDN technology supports Internet Protocol version 6 (IPv6) each flow can be identified easily and aggregated into bundles, which in turn simplify forwarding procedure at the switch level. A first step towards the improvement of the proposed architecture is done by extending the OF control to the fronthaul and cell site in CRAN architecture. The extension project is inspired by the benefits of SDN principle and has a *programmable fronthaul* (PF) and SDN enabled Remote Radio Heads. The PF will allow to program the fronthaul, thus changing the traffic steering in the network at large timescale, unlock new potentials of the network in terms of application awareness, data sensitivity and enhance QoS performance of the network. A novel Self-Adaptive Application Aware (SAAA) Algorithm will be introduced which will consider all types of applications and categorize them based on the network requirements. Depending on the category of applications the packets will be forwarded under specific constrained network metrics (path cost, link load, delay, or delay variation).

6 REFERENCES

1. ‘‘SK Telecom’s view of 5G vision, architecture, technology, and spectrum,’’ SK-Telecom, Seoul, South Korea, White Paper, 2014
2. Hicham, Magri, Noreddine Abghour, and Mohammed Ouzzif. "CLOUD RADIO ACCESS NETWORK TECHNOLOGY FOR THE NEXT FIFTH GENERATION MOBILE NETWORKS." *Journal of Theoretical and Applied Information Technology* 93.2 (2016): 375.
3. M. Jaber *et al.*: 5G Backhaul Challenges and Emerging Research Directions
4. P. Rost *et al.*, ‘‘Cloud technologies for flexible 5G radio access networks,’’ *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 68–76, May 2014.
5. White Paper, “Network Functions Virtualisation, An Introduction, Benefits, Enablers, Challenges and Call for Action.”
6. Open Networking Foundation (Apr. 2012), Software-Defined Networking: The New Norm of Networks. https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf
7. 3GPP. (2015). *An Interview with Philippe Reininger—RAN3 Chairman*. http://www.3gpp.org/news-events/3gpp-news/1684- ran4
8. 5G network architecture Huawei
9. ONF, “Open Networking Foundation, Wireless and Mobile Working Group,” https://www.opennetworking.org/images/stories/downloads/working-groups/ charter-wireless-mobile.pdf, 2013
10. R. Chávez-Santiago et al., ‘‘5G: The convergence of wireless communications,’’ Wireless Pers. Commun., vol. 83, no. 3, pp. 1617–1642, Mar. 2015.
11. *Working Party 5D (WP 5D)—IMT Systems, ITU Towards IMT for 2020 and Beyond.* http://www.itu.int/en/ITU-R/
12. J. Skold. (Feb. 2014). *Research Trends and IMT Beyond 2020*. https://www.itu.int/en/ITU-D/RegionalPresence/AsiaPacific/Documents/
13. ‘‘IMT Vision—Framework and overall objectives of the future development of IMT for 2020 and beyond,’’ document Rec. ITU-R M.2083-0, International Telecommunication Union, Nov. 2015.
14. NGMN Alliance. (Feb. 2015). *5G White Paper*. http://www.3gpp.org/technologies/presentations-white-papers
15. Rost, Peter, et al. "Cloud technologies for flexible 5G radio access networks." *IEEE Communications Magazine* 52.5 (2014): 68-76.
16. 5G America. http://www.5gamericas.org/en/
17. ITU towards ‘‘IMT for 2020 and beyond’’ http://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Pages/default.aspx
18. 5MGF activities for 2020 and beyond in Japan. http://5gmf.jp/wp/wp-content/uploads/2015/11/CEATEC2015_5GWS_5GMF.pdf
19. 5G Forum. https://www.5gforum.org/
20. 5GPPP. https://5g-ppp.eu/

21. J. F. Monserrat, G. Mange, V. Braun, H. Tullberg, G. Zimmermann, and Ö. Bulakci, ‘‘METIS research advances towards the 5G mobile and wireless system definition,’’ EURASIP J. Wireless Commun. Netw., vol. 2015, p. 53, Mar. 2015.
22. Alexander William Dawson, Mahesh K. Marina, Francisco J. Garcia "On the Benefits of RAN Virtualisation in C-RAN Based Mobile Networks". Pages 103-108 IEEE Computer Society Washington, DC, USA ©2014
23. ETSI GS NFV 001, "Network Functions Virtualisation (NFV); Use Cases," 2013.
24. J. Kempf, B. Johansson, S. Pettersson, H. Luning, and T. Nilsson, "Moving the mobile evolved packet core to the cloud," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*. IEEE, 2012, pp. 784–791.
25. A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt, "A Virtual SDN enabled LTE EPC Architecture: a case study for S-/P-Gateways functions," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*. IEEE, 2013, pp. 1–7.
26. P. Gurusanthosh, A. Rostami, and R. Manivasakan, "SDMA: A semi-distributed mobility anchoring in LTE networks," in *Mobile and Wireless Networking (MoWNeT), 2013 International Conference on Selected Topics in*. IEEE, 2013, pp. 133–139.
27. S. B. H. Said, M. R. Sama, K. Guillouard, L. Suciu, G. Simon, X. Lagrange, and J.-M. Bonnin, "New control plane in 3GPP LTE/EPC architecture for on-demand connectivity service," in *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*. IEEE, 2013, pp. 205–209.
28. X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: scalable and flexible cellular core network architecture," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 163–174.
29. M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic Flow Scheduling for Data Center Networks," in *NSDI*, vol. 10, 2010, pp. 19–19.
30. Y. Zhang, "An adaptive flow counting method for anomaly detection in SDN," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 25–30.
31. Qilin, Mao, and Shen Weikang. "A load balancing method based on SDN." *Measuring Technology and Mechatronics Automation (ICMTMA), 2015 Seventh International Conference on*. IEEE, 2015.
32. Kaur, Sukhveer, et al. "Round-robin based load balancing in Software Defined Networking." *Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*. IEEE, 2015.
33. Gau, Rung-Hung, and Pei-Kan Tsai. "SDN-based optimal traffic engineering for cellular networks with service chaining." *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*. IEEE, 2016.
34. Zhang, Junjie, et al. "Load balancing for multiple traffic matrices using SDN hybrid routing." *High Performance Switching and Routing (HPSR), 2014 IEEE 15th International Conference on*. IEEE, 2014.

35. Fizi, Faaiz S., and Shavan Askar. "A novel load balancing algorithm for software defined network based data centers." *Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom), International Conference on*. IEEE, 2016.
36. Bhandarkar, Smriti, and Kotla Amjath Khan. "Load Balancing in Software-defined Network (SDN) Based on Traffic Volume."
37. Ranjani, S. "Dynamic Load Balancing using Software Defined Networks."
38. Adalian, Nareg, et al. "Load balancing in LTE core networks using SDN." *Multidisciplinary Conference on Engineering Technology (IMCET), IEEE International*. IEEE, 2016.
39. Wang, Gang, et al. "Efficient traffic engineering for 5g core and backhaul networks." *Journal of Communications and Networks* 19.1 (2017): 80-92.
40. Ksentini, Adlen, Miloud Bagaa, and Tarik Taleb. "On using SDN in 5G: the controller placement problem." *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016.
41. Koryachko, Vyacheslav, Dmitry Perepelkin, and Vladimir Byshov. "Approach of dynamic load balancing in software defined networks with QoS." *Embedded Computing (MECO), 2017 6th Mediterranean Conference on*. IEEE, 2017.
42. A. Osseiran et al., “Scenarios for 5G mobile and wireless communications: The vision of the METIS project,” IEEE Commun. Mag., vol. 52, no. 5, pp. 26–35, onf.
43. A. Bradai, K. Singh, T. Ahmed, and T. Rasheed, “Cellular software defined networking: A framework,” IEEE Commun. Mag., vol. 53, no. 6, pp. 36– 43, Jun 2015.
44. C. J. Bernardos, et al., “An architecture for software defined wireless networking,” IEEE Wireless Commun., vol. 21, no. 3, pp. 52–61, 2014.
45. H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, “NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC),” IEEE Netw., vol. 28, no. 6, pp. 18–26, 2014.
46. K. Pentikousis, Y. Wang, and W. Hu, “MobileFlow: Toward software defined mobile networks,” IEEE Commun. Mag., vol. 51, no. 7, pp. 44–53, July 2013.
47. W. Tan, J. Zhang, C. Peng, B. Xia, and Y. Kou, “SDN-enabled converged networks,” IEEE Wireless. Commun., vol. 21, no. 6, pp. 79–85, 2014.
48. I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, “A roadmap for traffic engineering in SDN-OpenFlow networks,” Comput. Netw., vol. 71, pp. 1–30, 2014.
49. H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, “OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software Defined Networks,” in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, 2012, pp. 1–8.
50. R. Wallner and R. Cannistra, “An SDN Approach: Quality of Service using Big Switch’s Floodlight Open-source Controller,” *Proceedings of the Asia-Pacific Advanced Network*, vol. 35, pp. 14–19, 2013.
51. L. E. Li, Z. M. Mao, and J. Rexford, “CellSDN: Software-defined cellular networks,” 2012.
52. Cisco, “Global Mobile Data Traffic Forecast Update,” 2010–2015 White Paper, Feb. 2011.
53. Nokia Siemens Networks 2011, “2020: Beyond 4G Radio Evolution for the Gigabit Experience,” White Paper, Feb. 2011.

54. Ericsson, “More than 50 Billion Connected Devices,” White Paper, Feb. 2011, http://www.ericsson.com/res/ docs/whitepapers/wp-50-billions.pdf
55. Minimum requirements related to technical performance for IMT-2020 radio interface(s). https://www.itu.int/md/R15-SG05-C-0040/en
56. Coriant, “SDN PROGRAMMABLE CONTROL ON THE MOBILE EDGE,” White Paper, 2015. http://www.coriant.com/misc/downloads/Coriant_WP_SDN_on_the_Mobile_Edge.pdf
57. Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi. “Wireless network cloud: Architecture and system requirements”. In: *IBM Journal of Research and Development* 54.1 (Jan. 2010).
58. Santhi Kumaran “A Perspective of the Cellular Network of the Future: Cloud-RAN” Proceedings of the First International Afro European Conference for Industrial Advancement AECIA 2014
59. I. Hwang, B. Song, and S. Soliman. “A Holistic View on Hyper-Dense Heterogeneous and Small Cell Networks”. In: *Communications Magazine, IEEE* 51.6 (2013).
60. J. Hoydis, S. ten Brink, and M. Debbah. “Massive MIMO: How many antennas do we need?” In: *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*. 2011.
61. *Light Radio Network. Alcatel-Lucent*. https:// www.alcatel-lucent.com/solutions/lightradio
62. H. Holma and A. Toskala. *"LTE-Advanced: 3GPP Solution for IMT-Advanced"*. John Wiley and Sons, Ltd, 2012.
63. S. Thomas. *Poll: Savings Drive CRAN Deployments*. Light Reading, July 2015
64. GPP, TS. "23.401: General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access." *V12* 5 (2014).
65. “3GPP Evolved Packet System (EPC); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control Plane (GTPv2-C).” 3GPP TS 29.274 version 10.9.0 Release 10, 2013.
66. “General Packet Radio System (GPRS) Tunneling Protocol User Plane (GTOv1-U).” 3GPP TS 29.281 version 11.6.0 Release 11, 2013
67. Quality of Service (QoS) in LTE, Bec Technologies. www.bectechnologies.net/main/newsletter_images/QoS.pdf
68. Policy and Charging Rules Function (PCRF) in LTE EPC Core Network Technology. Netmanias. http://www.netmanias.com/en/?m=view&id=techdocs&no=10997
69. https://www.ijsr.net/archive/v4i5/SUB154155.pdf
70. “TR 23.843 V12.0.0 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on Core Network Overload (CNO) solutions (Release 12),” December 2013.
71. NFV White Paper, “Network Functions Virtualization: An Introduction, Benefits, Challenges and Call for Action,” 2012.
72. Hawilo, Hassan, et al. "NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)." *IEEE Network* 28.6 (2014): 18-26.
73. ONF, “OpenFlow-enabled SDN and Network Function Virtualisation (NFV),” 2014

74. Chowdhury, N. and R. Boutaba, "Network Virtualization: State of the Art and Research Challenges", Communications Magazine, IEEE, Volume 47, Issue 7, pp. 20-26, 2009.
75. Greenberg, A., Hjalmtysson, G., Maltz, D., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., and H. Zhang, "A Clean Slate 4D Approach to Network Control and Management", ACM SIGCOMM Computer Communication Review, Volume 35, Issue 5, pp. 41-54, 2005.
76. N. McKeown, OpenFlow (Or: “Why can’t I innovate in my wiring closet?”), http://www.openflow.org/documents/OpenFlow.ppt
77. T. Graf, “Underneath OpenStack Quantum: Software Defined Networking with Open vSwitch.”
78. IETF RFC 1349, “Type of Service in the Internet Protocol Suite,” 1992.
79. RFC 2474, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,”
80. Cisco, “Quality of Service - The Differentiated Services Model.
81. IETF, RFC2597, “Assured Forwarding PHB Group,” 1999.
82. IETF, RFC2598, “An Expedited Forwarding PHB,” 1999.
83. B. Liu, “Software Defined Networking and Tunneling for Mobile Networks,” 2013.
84. ONF, “OpenFlow Configuration and Management Protocol,” https://www.opennetworking.org/standards/of-config, 2013
85. C. Mulligan, “EPC and 4G Packet Networks: Driving the Mobile Broadband Revolution.” *Academic Press*, 2012
86. About VirtualBox. Oracle. https://www.virtualbox.org/manual/ch01.html
87. OPENDAYLIGHT: BERYLLIUM RELEASED. https://rojerfarre.files.wordpress.com/2016/02/odl-beryllium-diagram.jpg
88. OpenDaylight Documentation, Release Beryllium. https://media.readthedocs.org/pdf/opendaylight/1.0.0/opendaylight.pdf
89. What is a Floodlight Controller? SdxCentral. https://www.sdxcentral.com/sdn/definitions/sdn-controllers/open-source-sdn-controllers/what-is-floodlight-controller/
90. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to Algorithms. Section 24.3: Dijkstra’s algorithm.” MIT Press and McGraw–Hill. pp. 595-601. 2001
91. RFC 3550, “RTP: A Transport Protocol for Real-Time Applications,” http://www.rfc-base.org/txt/rfc-3550.txt
92. P. van Mieghem, “Data Communications Networking.” *Techne Press*, 2011.
93. T. Szigeti, C. Hattingh, R. Barton, and K. Briley, “End-To-End QoS Network Design: Quality of Service for Rich-Media and Cloud Networks.” *Pearson Education*, 2013
94. T. R.E, “Depth-first Search and Linear Graph Algorithms,” 1972
95. T. Pfeiffer, “Next generation mobile fronthaul and midhaul architectures [invited],” IEEE/OSA J. Opt. Commun. Netw., vol. 7, no. 11, pp. B38–B45, Nov. 2015
96. X. Wang et al., “Energy-efficient virtual base station formation in optical-access-enabled cloud-RAN,” IEEE J. Sel. Areas Commun., vol. 34, no. 5, pp. 1130–1139, May 2016.
97. Q. Chen, G. Yu, A. Maaref, G. Y. Li, and A. Huang, “Rethinking mobile data offloading for LTE in unlicensed spectrum,” IEEE Trans. Wireless Commun., vol. 15, no. 7, pp. 4987–5000, Jul. 2016.

98. G. Kalfas, N. Pleros, L. Alonso, and C. Verikoukis, “Network planning for 802.11ad and MT-MAC 60 GHz fiber-wireless gigabit wireless local area networks over passive optical networks,” IEEE/OSA J. Opt. Commun. Netw., vol. 8, no. 4, pp. 206–220, Apr. 2016.
99. H. Zhang, Y. Dong, J. Cheng, M. J. Hossain, and V. C. M. Leung, “Fronthauling for 5G LTE-U ultra-dense cloud small cell networks,” IEEE Wireless Commun., vol. 23, no. 6, pp. 48–53, Dec. 2016

7 GLOSSARY

7.1 ACRONYMS

- **5G** Fifth Generation Mobile Network
- **AF** Assured Forwarding
- **ALRP** Allocation and Retention Priority
- **AMBR** Aggregate Maximum Bit Rate
- **API** Application Programming Interface
- **ARP** Address Resolution Protocol
- **BBU** Base Band Unit
- **BBU Pool** Base Band Unit Pool
- **BE** Best Effort
- **CAPEX** Capital Expenditure
- **CLI** Command Line Interface
- **CPU** Central Processing Unit
- **CPRI** Common Public Radio Interface
- **CRAN** Cloud Radio Access Network
- **CWDM- PON** Coarse-Wavelength Division Multiplexing Passive Optical Network
- **DWDM-PON** Dense Wavelength Division Multiplexing Passive Optical Network
- **DFS** Depth First Search
- **DiffServ** Differentiated Services
- **eMBB** Enhanced Mobile Broad Band
- **ECM** EPS Connection Management
- **EMM** EPS Mobility Management
- **FL-C** Floodlight Controller
- **FTP** File Transfer Protocol
- **GUI** Graphical User Interface
- **GUTI** Globally Unique Temporary ID
- **HSPA** High Speed Packet Access
- **HSS** Home Subscriber Server
- **HTTP** Hypertext Transfer Protocol
- **HTTPs** Hypertext Transfer Protocol Secure

- **IMS** IP Multimedia System
- **IMSI** International Mobile Subscriber Identity
- **IP** Internet Protocol
- **IPv6** Internet Protocol version 6
- **LAN** Local Area Network
- **LC** Local Controller
- **LI** Lawful Interception
- **LISP** Locator / ID Separation Protocol
- **LocIP** Location IP
- **LTE** Long-Term Evolution
- **MAC** Media Access Control Address
- **MBB** Mobile Broad Band
- **MBR** Maximum Bit Rate
- **MME** Mobility Management Entity
- **MM-Wave** Microwave
- **NAS** Non-Access Stratum
- **NaaS** Network as a Service
- **DNS** Domain Name System
- **DSCP** Differentiated Services Code Point
- **ECGI** E-UTRAN Cell Global Identifier
- **ECMP** Equal Cost Multipath
- **ECM** EPS Connection Management
- **EF** Expedited Forwarding
- **EMM** EPS Mobility Management
- **EPC** Evolved Packet Core
- **E-UTRAN** Evolved UMTS Terrestrial Radio Access Network
- **FE** Forwarding Element
- **4G** Fourth Generation
- **GBR** Guaranteed Bit Rate
- **3GPP** 3rd Generation Partnership Project
- **GGSN** Gateway GPRS Support Node
- **GLM** Gateway Load Manager
- **GRE** Generic Routing Encapsulation
- **GSM** Global System for Mobile Communications
- **GPRS** General Packet Radio Service
- **GTP** GPRS Tunneling Protocol
- **GTP-C** GPRS Tunneling Protocol Control Plane
- **GTP** GPRS Tunneling Protocol Prime
- **GTP-U** GPRS Tunneling Protocol User Plane
- **GTPv2** GPRS tunneling protocol version 2
- **2G** Second Generation

- **NAT** Network Address Translation
- **NE** Networking Element
- **NFV** Network Function Virtualisation
- **NFVI** NFV Infrastructure
- **NGMN** Next Generation Mobile Network
- **OAM** Operations, Administration, and Management
- **OF** OpenFlow
- **ODL-C** OpenDaylight Controller
- **ONF** Open Networking Foundation
- **OPEX** Operating Expenditure
- **OS** Operating System
- **OVS** Open vSwitch
- **PCEF** Policy Control Enforcement Function
- **PCRF** Policy and Charging Rules Function
- **PDN** Packet Data Networks
- **PGW** Packet Data Network Gateway
- **PGW-D** PDN Gateway Data
- **PHB** Per-Hop Behaviour
- **QCI** QoS Class Indicator
- **QoE** Quality of Experience
- **QoS** Quality of Service
- **REST** Representational State Transfer
- **RRH** Remote Radio Heads
- **RRC** Radio Resource Control
- **RTP** Real-time Transport Protocol
- **SAL** Service Abstraction Layer
- **SaaS** Software as a Service
- **SDN** Software Defined Networking
- **SDN-BBU** Software Network Defined Base Band Unit
- **SGSN** Serving GPRS Support Node
- **SGW** Serving Gateway
- **SGW-D** SGW User Plane
- **SIM** Subscriber Identity Module
- **SIP** Session Initiation Protocol
- **SIR** Signal to Interface
- **SNMP** Simple Network Management Protocol
- **SMTP** Simple Mail Transfer Protocol
- **S/PGW** Serving/Packet Data Network Gateway
- **S/PGW-D** S/PGW Data Plane
- **S/PGW-C** S/PGW Control Plane
- **SPI** Security Parameter Index

- **SSL** Secure Sockets Layer
- **SSH** Secure Shell Protocol
- **TA** Tracking Area
- **TAI** Tracking Area Identify
- **TAU** Tracking Area Update
- **TCAM** Ternary Content-Addressable Memory
- **TCP** Transport Control Protocol
- **TEID** Tunnel Endpoint Identity
- **UE** User Equipment
- **UDP** User Datagram Protocol
- **URLLC** Ultra Reliable and Low Latency Communications
- **Wi-Fi** Wireless Fidelity
- **WDM-PON** Wavelength Division Multiplexing Passive Optical Network