AMERICAN UNIVERSITY OF BEIRUT

Virtualization of the LTE EPC using the SDN approach

by
ALI ASSAAD TAWBEH

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science
to the Department of Computer Science
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
December 2015

AMERICAN UNIVERSITY OF BEIRUT

Virtualization of the LTE EPC using the SDN approach

by

ALI ASSAAD TAWBEH

Approved by:

Haidar Safa

Dr. Haidar Safa, Associate Professor — Advisor

Computer Science

Dr. Wassim EL Hajj, Associate Professor — Member of Committee

Computer Science

Dr. Hassan Artail, Professor — Member of Committee

Electrical and Computer Engineering

Date of thesis defense: December 22, 2015

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: ____

Last First Middle

○ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☐ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, **three years after the date of submitting my thesis, dissertation, or project,** to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

____

Signature Date

An Abstract of the Thesis of

Ali Assaad Tawbeh for Master of Computer Science
Major: Computer Science

Title: Virtualization of the LTE EPC using the SDN approach

In the LTE Evolved Packet Core (EPC), many network entities and interfaces have to be maintained and updated regularly. Moreover, to accommodate more users, new hardware must be integrated but rarely used. To address these challenges, the EPC can be moved to the cloud using two modern technologies: Software Defined Networking (SDN) and Network Function Virtualization (NFV). In this thesis, we study the impact of integrating these novel technologies on LTE networks. We propose a hybrid approach for selecting whether to apply NFV or SDN on each gateway at a given time while minimizing the network load taking into consideration some parameters such as the number of active datacenters, the deployment city population, the intensity at a given time, the QoS class identifier (QCI), the generated traffic volume, and the delay budget. We formulated SDN decomposition/NFV virtualization selection, as an optimization problem where the objective is to minimize the network load subject to a set of constraints. The proposed solution is more responsive to the dynamic state of the network such that for a given gateway, at a certain time slot, an SDN decomposition might be the optimal choice while at another time slot with different network state, the NFV architecture might be more suitable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we start by briefly presenting an overview of LTE, SDN and NFV, we then introduce the motivations behind this thesis and define our problem. Thesis objectives and contribution are also summarized.

1.1 Basic Concepts

1.1.1 LTE

3GPP answered the growing need for mobile network services such as voice over IP (VoIP), video calling, video on demand, online gaming and world wide web browsing, by introducing the Long Term Evolution (LTE). In order to satisfy the variety of nowadays applications, LTE was standardized with a set of sophisticated requirements that overtook the features and capabilities of 3rd generation networks (3G) that were originally designed to support basic network services, mainly voice services.

The aim of LTE is to double the spectrum efficiency in comparison with the older systems' generations, increasing the bit rate for cell-edge users in order to provide a wider coverage, increasing data rates and efficiently supporting high user mobility. Also LTE support a wide range of changeable bandwidth which makes it suitable for worldwide market. Moreover LTE introduce an important modernity which is the use of sophisticated Radio Resource Management (RRM) techniques in order to enhance Quality of Service (QoS).

The flat architecture of LTE, depicted in figure 1.1 [1], guarantees supporting seamless mobility in addition to delivering data and signaling at high speed. The architecture is mainly composed of two parts; the Evolved Packet Core (EPC) and the Evolved Universal Terrestrial Radio Access Network (E-UTRAN).

The main entities that are involved in the EPC are: 1) the Mobility Management Entity (MME) which is responsible for mobility, handover, tracking and paging users, 2) the Serving Gateway (SGW) that interfaces with the E-UTRAN

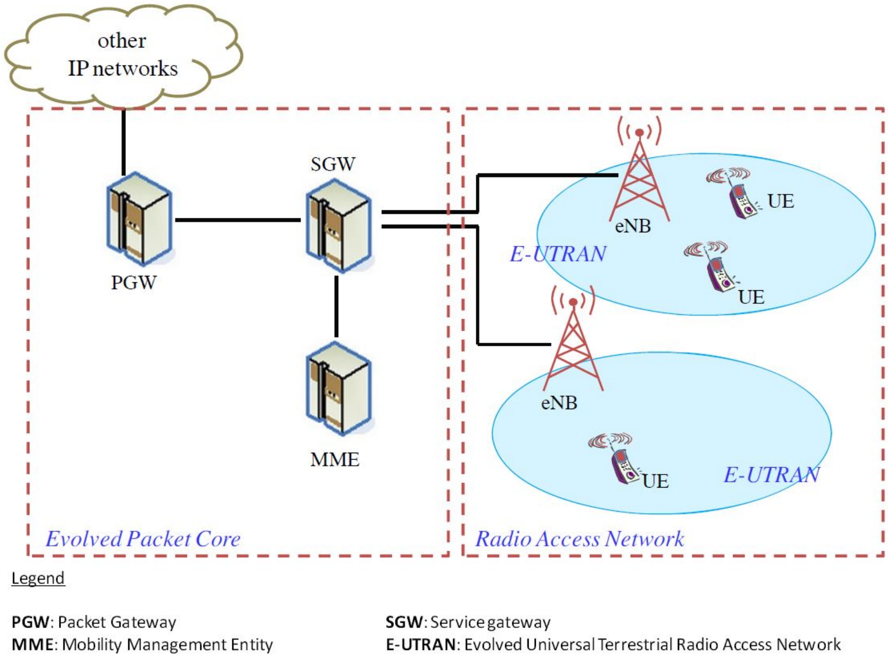

Figure 1.1: LTE Architecture

and routes packet in the EPC and 3) The Packet Data Network Gateway (PGW) which connects the EPC to other Packet Data Networks (PDNs) such as the internet.

The E-UTRAN comprises only two entities: the evolved Node B (eNB) and the User Equipment (UE). Unlike older cellular network generations, the eNB is the only node in the radio access network that is responsible for managing radio resources and controlling procedures.

1.1.2 SDN

Current networking elements have two types of functionalities: control and data flow. Both types are implemented on the same physical devices. To control the network devices, a network administrator needs to program each separately. Software Defined Networking (SDN) was proposed to provide more detailed control plane configuration. Its basic idea is to separate the control plane from the data forwarding plane. In other words, SDN aims to decouple the intelligence of the switch (switching and routing) and move it to a central datacenter, while keeping the switch fabric for data forwarding. This gives the network administrator the power of configuring the network from one central node instead of visiting each element in the network. The main benefits of SDN are 1) Programmability of the network, 2) the rise of virtualization and 3) device configuration and troubleshooting .

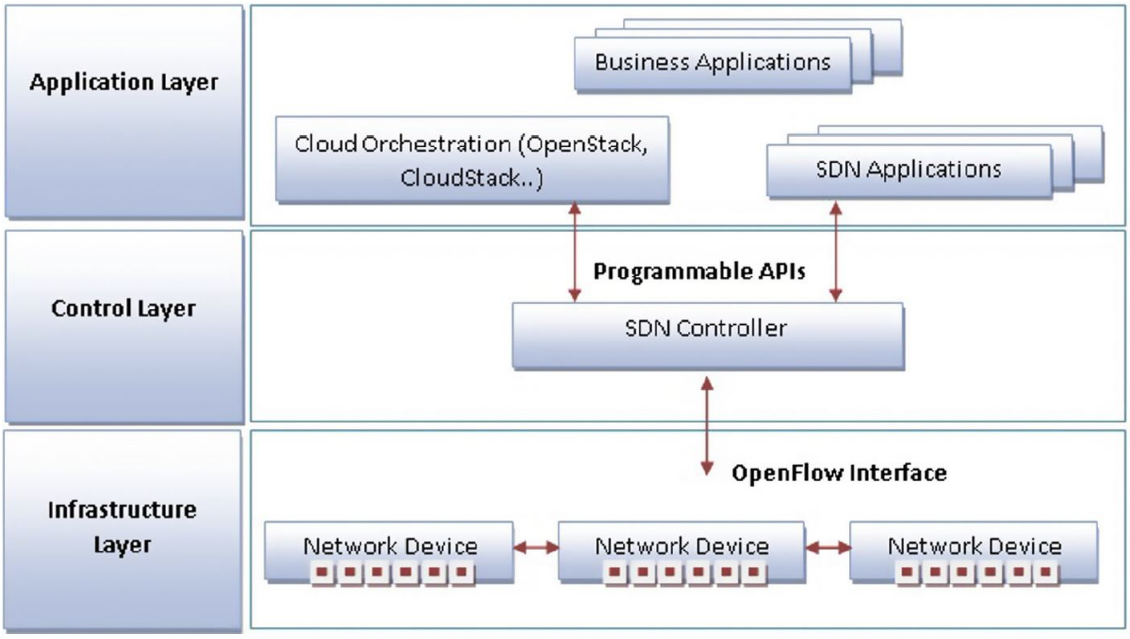

Figure 1.2: SDN Architecture

Figure 1.2 [2] depicts the basic architecture of an SDN network. It is basically composed of three layers: 1) the control layer that abstracts the infrastructure of the network and implements its operating system of the network, 2) the application layer that implements network services and business applications (such as Voip, FTP, routing, etc.) and exploits standard APIs to communicate with the controller, and 3) the infrastructure layer that is composed of simple switches that perform data forwarding.

The OpenFlow protocol is the first standardized protocol between the fabric switches and the controller. This protocol allow the controller to modify the flow tables of each switch in order to instruct them how to process the packets of each flow.

1.1.3 NFV

The sustainable increase of challenges facing network administrators is due to the strong dependence of networks on the underlying hardware devices such as Deep Packet Inspection (DPIs), firewalls, routers, etc. The fast pace of innovation reduced the life cycle of hardware devices which resulted in multiplying CAPEX (Capital Expenditure) and OPEX (Operational Expenditure) of network administrators. The evolution of IT virtualization, lead to the development of NFV. NFV is the concept of shifting from hardware appliances to software instances. It is based on implementing network function on general purpose servers as software instances instead of using dedicated hardware for each function. NFV provides openness of platforms in addition to flexibility and scalability for the telecommunications industry, and most importantly, it allows the integration of new features and applications at a minimum cost. Subsequently, the network operator no longer needs to buy new hardware to replace the existing ones since it can simply run software instances of the new features on the existing hardware.

1.2 Motivation

The increasing number of users and the large income generated by the industry from LTE networks, assure the successfulness and robustness of this technology in the recent years. However, due to the complex nature of traditional cellular network, its ability is questionable whether it can handle the future growth and the entrance of new technologies, such as cloud computing and distributed content, to the mobile operator domain.

LTE EPC is composed of a set of hardware components that each perform fit-for-purpose function with dozens of standardized interfaces that each has a unique definition. For these reasons, the LTE EPC is considered a closed system that lacks flexibility. Evolving the EPC in the same way will have negative results. Operators are going to face increasing capital and operational expenditures (CAPEX/OPEX) while the Average Revenue Per User (ARPU) will decrease. Meanwhile, many types of network entities must be maintained; moreover dozens of unique interfaces should be updated regularly. The high CAPEX/OPEX restrain operators from investing further because of the time gap between standardizing features and actually implementing them. Despite that this process assures obtaining the needed hardware with quality standards, but it relents operators from evolving fast. Each hardware entity is dedicated to perform a specific function. To accommodate more users, new hardware must be integrated but will not be used during off-peak times. This is an inefficient use of resources since hardware equipment are deployed and maintained but not often used. Also adding new network services often comes along with new equipment, making a little re-use of existing ones.

To address these challenges, many researchers thought of moving the EPC to the cloud motivated by the core principles of two modern technologies mentioned earlier: SDN and NFV.

1.3 Problem Definition

NFV and SDN are two promising technologies to innovate current cellular networks with out-of-the-box ideas. Despite their potential in building more robust, flexible, reliable and high performance service delivery networks they may have drawbacks in terms of network load and delay.

NFV requires redirecting traffic to datacenters where network functions are implemented as software instances which adds more load on the network and increases the end to end delay. On the other hand, the integration of SDN to the mobile network require adding a control layer. When SDN is given higher levels of control, the rate of signaling and configuration messages between the controller and the data forwarding elements will result in additional load overhead on the transport network.

In this thesis we study the impact of integrating SDN and NFV on LTE networks. Our aim is to investigate the benefits of applying NFV or SDN on each gateway (PGW and SGW) at a given time in a way that minimizes network load taking into account some parameters such as the number of active datacenters, the population of the city of deployment, the intensity at the given time, the QoS class identifier (QCI) and the volume of generated traffic in addition to the packet delay budget.

1.4 Objectives and Contribution Summary

In this thesis we propose to adopt a hybrid architecture for SGW and PGW gateways, in other words applying both SDN decomposition and NFV concept on each gateways. In other related work either deployment was adopted for a given gateway. However our proposition gives a more granular control over the network. We believe that it is more optimality oriented and responsive to the dynamic state of the network because for a given gateway, at a certain time slot, an SDN decomposition might be the optimal choice for the current network state, however, at another time slot, the state of the network might change and the NFV architecture becomes more suitable. Our contribution can be summarized as following:

1. We surveyed related work found in the literature and identified its limitations.

2. We proposed a hybrid architecture where on each gateway both SDN decomposition and NFV are applied. Depending on several factors and parameters, a deployment is selected.

3. We formulated this selection problem, between SDN decomposition and NFV virtualization, as an optimization problem where the objective is to minimize the network load subject to a set of constraints.

4. We used the QoS Class Identifier (QCI) of each bearer in order to determine the suitable delay budget.

5. We developed a JAVA frame work and we used Gurobi optimization tool in order to implement and evaluate our proposed model.

So in short, for a given gateway, at a given time, a set of QCIs may be operating on one deployment, while the other set will be operating on the other deployment.

1.5 Thesis Organization

The remaining of this thesis is organized as follows: chapter 2 gives an overview on the used technologies, namely we focus on LTE, SDN and NFV and the challenges of virtualizing the EPC. In chapter 3 we survey some of the related work concerning applying SDN and NFV on LTE networks focusing on the work studying the impact of applying both concepts. In Chapter 4 we identify the limitations of the related work and present the proposed approach. In chapter 5, we talk about the implementation details then we present and analyze the results. We conclude in chapter 6.

Chapter 2

Background

In this chapter, we begin by briefly reviewing telecommunication systems and its generations, then we present motivations behind introducing LTE, and describe its high level architecture. We delve into the LTE protocol stack that governs the communication among its layers and entities. We also present data transport mechanism and management of data flows for each user. Then we move to Software Defined Networking (SDN) as we point out the main changes that led researchers to think about a new and innovative network architecture. We tackle the main limitations of current network technologies including cellular network technologies, then we describe the SDN architecture before explaining OpenFlow, a standard SDN protocol. The last section of this chapter is about Network Functions Virtualization (NFV). We start by defining this new technology before explaining its relationship with SDN. Then we expose the use cases of NFV and its benefits and how it can influence the network and how it is enabled. We finish by identifying and listing some of its implementation challenges.

2.1 Long Term Evolution (LTE)

2.1.1 History

When wireless communication systems were first deployed they were used by a relatively small group of individuals because of their high cost. Nowadays, this technology have become an essential part of our daily life, financially more affordable and available at the fingertip of the majority of the world's population. Telecommunication technologies were developed in a sequence of generations: 1G was based on the analogue mobile radio systems, 2G was the first digital mobile system, the 3G main purpose was to handle broadband data. Recently, the Long Term Evolution (LTE) and Long Term Evolution Advanced (LTE-A) are the most promising systems among the 4th and 5th generations.

The First Generation (1G)

When first deployed in the early 1980s, 1G systems were based on analogue communication technologies with Frequency Division Multiple Access (FDMA) as an access method. Cell sizes were relatively large with low capacity since the radio spectrum was not used efficiently. Most of users were business users because mobile phones were large and expensive.

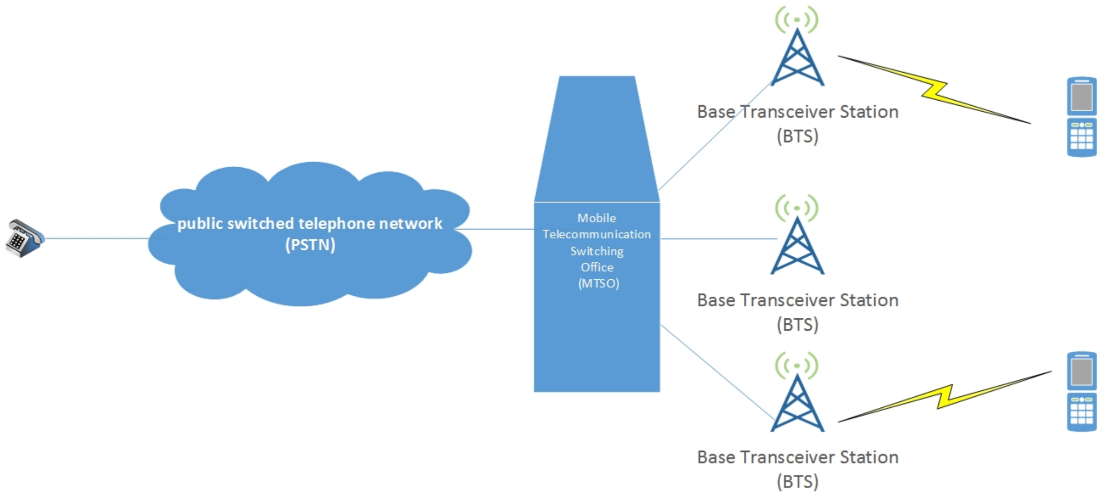

Figure 2.1: First generation system architecture.

The figure above [3] represents the basic architecture of the first generation telecommunication systems. Mobile Telecommunication Office (MTSO) connects Public Switching Telephony Network (PSTN) to base stations. Each geographical area had its own location and equipment databases. Base Transceiver Station (BTS) is a radio component including sender, receiver and antenna.

The Second Generation (2G)

In the 1990s, mobile communication became more commercialized with the introduction of second generation systems (2G). The system capacity increased because of the use of digital technologies and the mobile phones became smaller and cheaper. Short Message Service (SMS) was introduced for the first time. Time Division Multiple Access (TDMA) along with Frequency Division Multiple Access (FDMA) where used as access methods.

2G architecture was mainly composed of two subsystems as shown in Figure 2.2: Radio Subsystem (RSS) and Network and Switching Subsystem (NSS). RSS was composed of BTSes and Base Station Controllers (BSCs) that control BTSes and perform switching between them and manage network resources. The NSS contains Mobile Services Switching Center (MSC) that handle signaling messages and setting-up/tearing-down phone calls. Home Location Register (HLR) is the

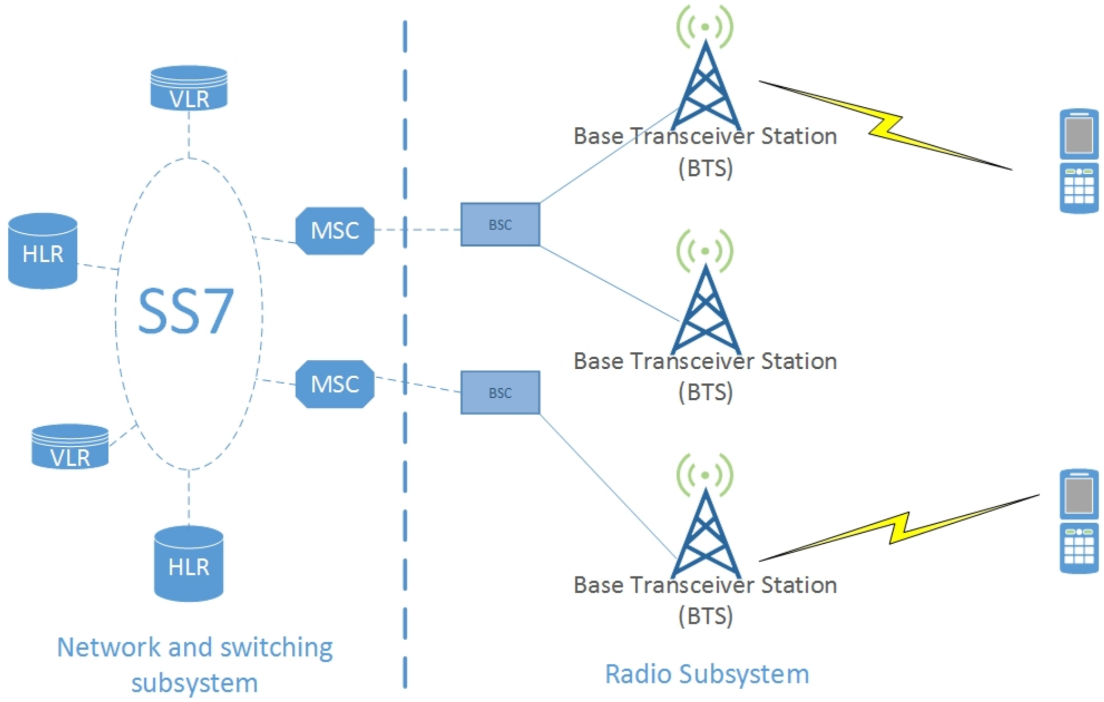

Figure 2.2: Second generation system architecture

master database that contains user data and information about subscribers in its network. Visitor Location Register (VLR) is a local database for a temporal subset of users. It is also worth to mention that the so-called 2.5G systems are based on 2G system with the ability to provide a limited packet switching service to access the internet.

The Third Generation (3G)

In the third generation, the radio access network was innovated; instead of the BTS and BSC, we now have the Node B. Packet switched and circuit switched data transmission were both supported as shown in Figure 2.3 [3]. In the circuit switched domain, MSC has the same role as in 2G systems while the media gateways (M-GWs) are responsible of routing phone calls between different parts of the network. In the packet switched network, the gateway GPRS support nodes (GGSNs) serve as interfaces with external servers and packet data networks (e.g Internet or IP Multimedia Subsystem (IMS)). On the other hand, Serving GPRS Support Nodes (SGSNs) are responsible of routing data between base stations and GGSNs. The home subscriber server (HSS) is a central database that stores information about every subscriber to the network operator. The access method was based on Code Division Multiple Access (CDMA). Further optimization was introduced to radio access network through the use of technologies such as High

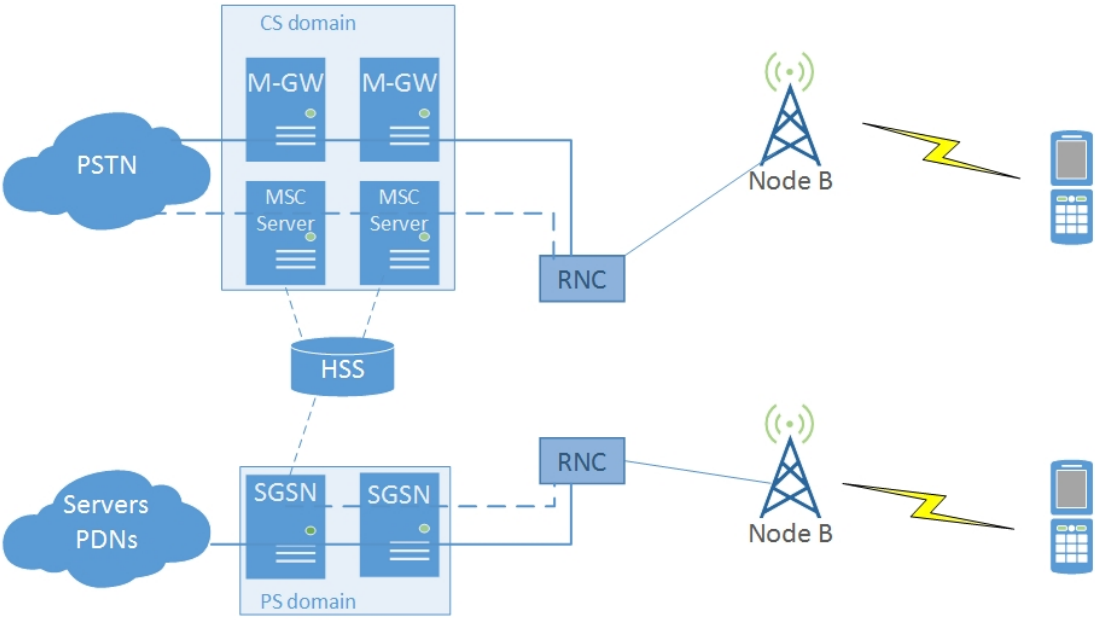

Figure 2.3: Third generation system architecture.

Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA) which led to an increase in the average rate of users upload and download. After these enhancements, the system was known as 3.5G.

2.1.2 The need for LTE and its features

The main motivation of designing LTE systems is the growth of mobile data. Since the deployment of mobile communication systems, the traffic was dominated by voice calls but the use of mobile data is in a dramatic increase since the year 2010 as shown in Figure 2.4 which is taken from Ericssons' mobility report of June 2014 [4]. This increase is due to the spread of 3.5G networks and the introduction of smartphones with user-friendly operating systems and huge number of mobile applications along with the unlimited download/upload of data enabled by operators. This growth had led to a congestion in 2G and 3G networks, as a result increasing network capacity became a must.

Another motivation is the high expenditure of 2G and 3G networks since they support both circuit switched (for voice) and packet switched (for mobile data) core networks while everything can be moved to the packet switched domain by introducing voice over IP (VoIP) techniques. To make such techniques efficient and useful, end-to-end delay must be reduced.

LTE aims at increasing the speed and capacity of wireless mobile networks in addition to reducing system architecture complexity to an IP-based system. Some of its main features are: high peak download and upload rate, reduced

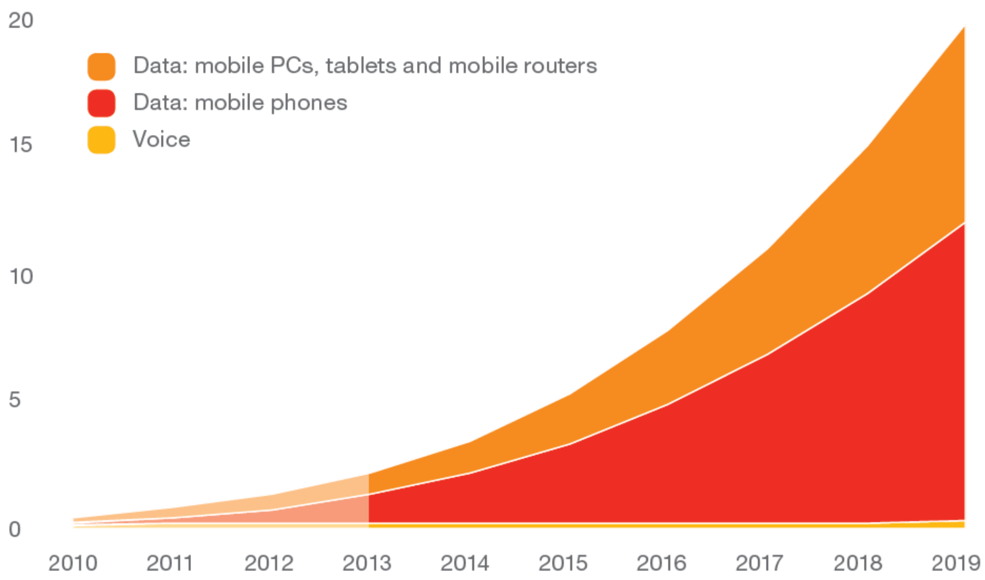

Figure 2.4: Global mobile traffic (monthly ExaBytes)

data transfer latency, using flexible bandwidth ranges, supporting multiple cell sizes with a radius up to 100 km, serving a relatively high number of active users per cell. In addition Orthogonal Frequency Division Multiple Access (OFDMA) is used in the downlink while Single Carrier Frequency Division Multiple Access (SC-FDMA) in the uplink.

2.1.3 Architecture of LTE

The LTE, which is an evolution of the GSM/UMTS standards, is composed of two parts: evolved UMTS terrestrial radio access network (E-UTRAN) and the Evolved Packet Core (EPC) as shown in Figure 2.5 [3]. The EPC interfaces with packet data networks such as the internet, the IP Multimedia Subsystem (IMS) or private corporate networks. E-UTRAN has a single component, which is the evolved Node B (eNB), that handles communications between user equipment and EPC.

Evolved UMTS Terrestrial Radio Access Network (E-UTRAN)

The radio access network, known as evolved UMTS terrestrial radio access network (E-UTRAN) connects the User Equipment (UE) to the evolved packet core (EPC) through the evolved Node B (eNB). The eNB sends data, as radio transmissions, to its UEs on the downlink and receives data from them, always as radio transmissions, on the uplink. The eNB handles its UE's low-level operations, such as handover, by sending them signalling messages. The eNB is connected to the UEs via the Uu interface, to the EPC via the S1 interface and to other eNBs via

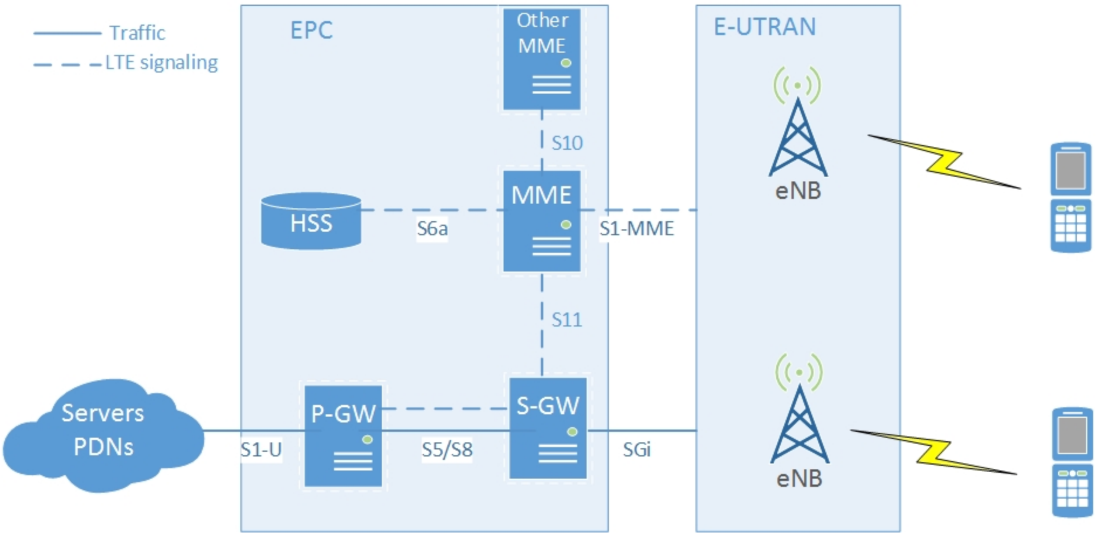

Figure 2.5: Main components of the EPC and the E-UTRAN.

the X2 interface.

Evolved Packet Core (EPC)

The evolved packet core (EPC) is mainly composed of the home subscriber server (HSS), the packet data network gateway (PGW), the service gateway (SGW) and the mobility management entity (MME). The HSS, which remained unchanged from UMTS and GSM, is basically a data base containing the profiles of all network operators subscribers. PGW handles the communication between the EPC and one or more packet data network, through the SGi interface, such as the operator's server, the IP multimedia subsystem (IMS) or the Internet. SGW plays the role of a router; it forwards data from the eNB to the correspondent PGW. The MME deals with high level operations, it is a control entity that manages data streams, security issues and other network elements by sending signaling messages.

2.1.4 Communication Protocols

The protocol stack is composed of two planes and layers as shown in Figure 2.6 [3]. The user plane protocols deal with users' originated or terminated data while the control plane protocols are used by the network elements only. The upper layer is LTE-specific while the lower layer transport data from one network element to another. The user plane implements protocols that ensure reliable transfer of data between UEs and PDNs taking into account UEs' mobility. The Most used protocol in this plane is the GPRS tunneling protocol user part (GTP-U). The control plane contains a handful of signaling protocols that define how information are exchanged between different network components. For example, on the air interface, an eNB communicates with a UE using the radio resource control (RRC). In the fixed network, the MME, the SG-W and the PG-W communicate via the GPRS tunneling protocol control part (GTP-C).

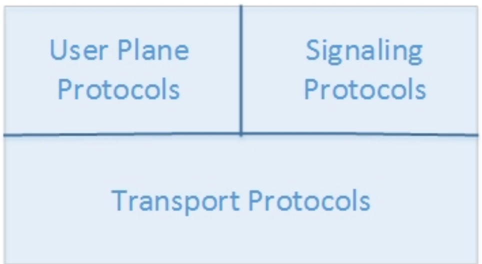

Figure 2.6: LTE's high level protocol architecture

As we mentioned earlier, the MME controls some of the UE's high level behavior but there is no direct link between the MME and the UEs, this is why the air interface is divided into two levels: the non-access stratum (NAS) that handles high-level signaling messages and the access stratum (AS) that transports these messages on the S1 and Uu interfaces, as shown in Figure 2.7 [3].

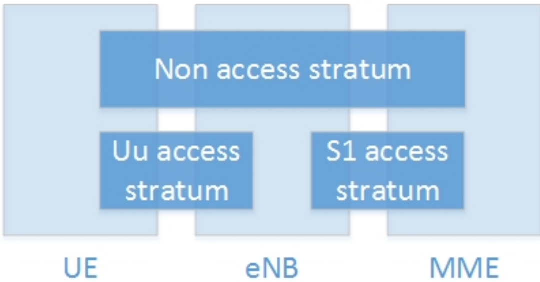

Figure 2.7: Air interface levels

2.1.5 Bearer Management

In LTE systems, data is transported from one part of the system to the other using bearers which are implemented according to the protocol used on the S5/S8 interface. Most implementations use the GTP protocol. The Evolved Packet System bearer (EPS bearer), which is the most important, is a bi-directional pipe of data composed of one or more service data flows that carry information from the user equipment to the PDN gateway according to a determined quality of service (QOS). A service data flow is associated with a certain service or application and it embodies several data flows constituting the service. All the service data flows that are included in the same bearer are assigned the same

QOS. Setting up, tearing down, and modifying bearers are managed by GTP-U and GTP-C protocols.

There are two types of bearers; Guaranteed Bit Rate bearers (GBR bearers) and non-GBR bearers. A GBR bearer is characterized by a guaranteed bit rate for data that makes it suitable for real time applications while a non-GBR bearer does not provide such guarantees and thus can be used for non-real time applications and services.

A default bearer is a non-GBR bearer that is set up, with an IP address, when a mobile is attached to the network in order to provide it with always-on connectivity to a default PDN. For each PDN that the UE desires to connect to, other than the default one, it receives an additional IP address with a new default bearer. One or more dedicated bearers can be received after establishing default bearer in the same network. A dedicated bearer uses the IP address of its parent default bearer and it can have higher QOS and guaranteed bit rate.

GTP-U handles matching between S1 and S5/S8 bearers and transport layer protocols using bi-directional tunnels. For each bearer, the GTP-C gives two tunnel endpoint identifiers (TEIDs), via signaling messages , one for the downlink tunnel and the other for the uplink tunnel.

When a PGW receives a packet from the PDN, it has to assign it to the correspondent bearer. The correspondent bearer is identified using its associated traffic flow template (TFT) that is composed of as many packet filters as the number of packet flows that constitute the bearer. Packet filters contain basically the destination and source IPs in addition to the correspondent TCP or UDP port numbers. After identifying the GTP-U tunnel, the PDN adds to the packet TEID and the SGW's IP address of the mobile. When the packet arrives to the SGW, it repeats the same process in order to deliver the packet to the UE.[3]

2.2 Software Defined Networking (SDN)

2.2.1 Motivations

The need to change

The exponential increase of mobile devices and data, always-on connectivity demand, and cloud computing services motivate researchers to think about changing the static traditional network architecture into a more dynamic architecture able to meet market requirements more efficiently. Some of the main reasons are: 1) **changing traffic patterns** where in current applications the connection occurs between a client and multiple servers and databases exchanging information, in contrast to the traditional connection that occurs between one server and one client. Also clients are accessing networks using any device at anytime from anywhere, 2) **consumerization of IT**, where the IT departments responsible for corporate networks are being under pressure in order to accommodate access of clients using personnel devices, 3) **the rise of cloud services** where currently enterprises are highly leveraging cloud services for an easy and agile access to services and application, 4) **security, scalability and storage** should be taken into account when designing cloud services, 5) **the emerging of big data** where enterprises are now dealing with huge datasets processed on thousands of servers that need direct connection between each other, therefore 6) **robust scaling abilities and greater network bandwidth capacity** are needed [5, 6].

Current network limitations

The main limitations of current networking technologies are:

1. **Complexity that leads to stasis**: networks are becoming more and more complex because of the big sets of protocols that connect hosts to network's nodes. Each of these protocols is dedicated to solve a particular problem with no mean of abstraction, so any update on the network or configuration of resources is highly manual and must touch every node in the network. This is why networks are relatively static in contrast to the dynamic nature of today's server that uses virtualization to accommodate a higher number of users. So when VMs (Virtual Machines) migrate from a physical server to another, the physical end points change and this mechanism challenges the current network architecture.

2. **Inconsistent policies**: every time a new virtual machine is instantiated for example, IT must configure all ACLs (Access Control Lists) of every node which make take hours to days.

3. **Inability to scale**: in order to serve the growing number of users, more network devices are added to the network which makes it more and more complex. Also big companies that deal with huge datasets use large scale parallel processing algorithms and so the number of computing node increases as well as the data exchanged which may affect the performance seriously.

4. **Multi-tenancy** makes things even more complex.

5. **Vendor dependence**: a mismatch between vendor's equipment and market need, the equipment lifecycle can range to more than three years which make it unable to respond to the dynamic market needs.

2.2.2 SDN architecture

SDN is a network architecture where the control plane is separated from forwarding plane and is directly programmable. Figure 2.8 illustrates the SDN architecture. In this figure, the infrastructure layer, also called forwarding layer, is composed of programmable switches, routers and other physical devices that perform data forwarding. The intermediate layer is the control layer, that represents the intelligence of the network and contains the controller that runs the SDN control software (i.e. operating system for the network) in order to administer the infrastructure layer and manage its resources via an open protocol interface such as OpenFlow [10]. The control layer provides a set of APIs (Application Programming Interfaces) to communicate with the application layer which implements customized business applications and network services such as security, QoS, routing, bandwidth management, etc. This separation abstracts the infrastructure for network services and applications so the network appears as a single logical switch maintained by centralized software-based SDN controllers. The nodes no longer need to process bunches of protocols, instead they are instructed by the SDN controllers. Just like how a CPU has an instruction set to program a computer system, the protocol defined between the controller and the forwarding layer specifies some basic primitives that can be used by an application running on top of the controller in order to program the infrastructure layer. By centralizing network's intelligence, network operators and administrators can create new services and configurations that manage resources instead of hundreds of configuration for each device or waiting for features to be embedded in vendor's products [6] lines.

2.2.3 Benefits

In this section, we are going to explain some of the main benefits of applying the SDN concept.

1. **Network programmability:** SDN can overcome the inflexibility and complexity of the current networks by implementing the control plane. The abstraction of the data and control plane reduces the complexity of the infrastructure so the nodes do not run bunches of protocols anymore. SDN also provides visibility to the applications and services simplifying the network management. Operators can easily define network flows with specific requirements to serve users having specific needs. Adding new or custom policies or new devices to the network does not require visiting and reconfiguring every node. This can be done programmatically on the control plane which is separated from physical devices.

2. **Enhancing virtualization techniques:** In current network architectures, migrating virtual machines from one datacenter to another may cause service interruption when updating MAC address, while when adopting SDN's architecture MAC addresses can be abstracted by using tunnels by SDN. SDN enables multi-tenancy such that each customer has its own virtual slice of the network. SDN has the potential to offer network as a service

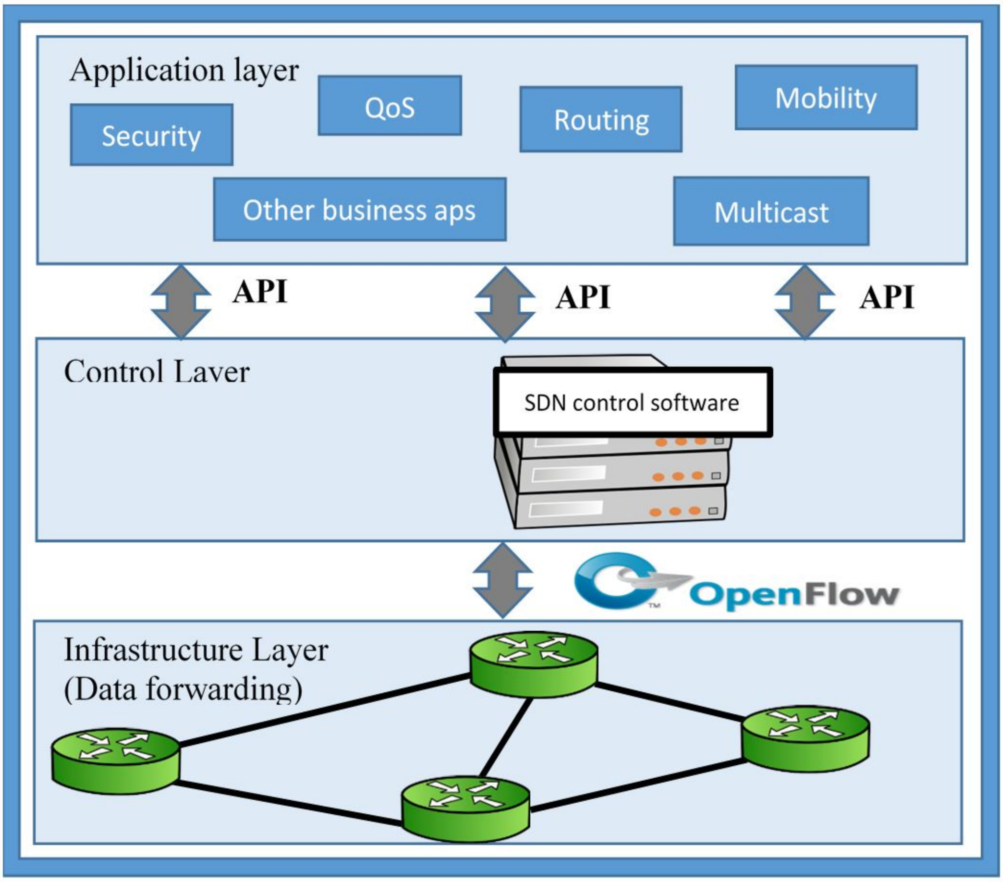

Figure 2.8: SDN's architecture

(NAAS), which enables virtual operators and gives a more flexible service models with the ability to control their traffic.

3. **Troubleshooting and configuration of devices:** SDN makes networks more dynamic and adaptable to the requirements due to the ability of configuring and troubleshooting a device from a single point. Also SDN permits rapid innovation in networks since it can offer testing platform on the same network. Therefore, testing new policies and protocols can use production traffic without causing user experience interruption.

This separation between control and switching is an important step toward intelligent networks [2].

2.2.4 OpenFlow

OpenFlow is the first standard protocol defined between the control layer and forwarding layer and is implemented on both sides. OpenFlow also enables the network to respond to real-time changes at the application, user and session levels. The main benefits of OpenFlow are: centralized control of multi-vendor environments, reduced complexity through automation, higher rate of innovation, increased network reliability and security, more granular network control and better user experience[7].

The idea of OpenFlow exploits the fact that the majority of modern switches and routers support a flow-tables to implement a set of common functions like firewalls, statistics collection, Network Address Translation (NAT), QOS filtering, etc. OpenFlow enables programming the flow-tables via an open protocol.

Figure 2.9 [7] depicts a basic OpenFlow switch. To classify a switch as an OpenFlow switch, it must consists of at least three parts: 1) *A flow table* where each entry is associated with an action that decides how to process the incoming flow, 2) *a secure channel* that connects the switch to the datacenter where the controller resides, 3) *the OpenFlow protocol* which is implemented on the switch's side and the controller side to standardize and control the communication between both sides.

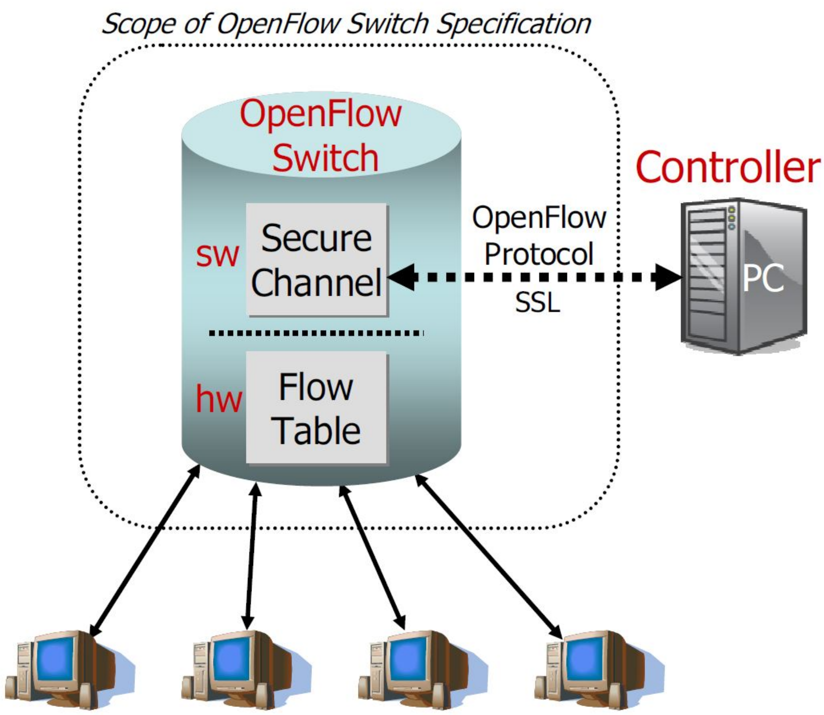

Figure 2.9: Main components of an OpenFlow switch

The standardization of the interface (OpenFlow protocol) that communicates with the flow table allows the definition of each entry externally and thus avoiding the need to program the switch.

The possible actions that a switch might perform when it receives flow packets are:

1. Forward this flow's packets to a given port: this action allows routing of packets through the network.

2. Encapsulate and forward this flow's packet to the controller: this action is usually taken for the first packet of a new flow. The controller determines how this flow should be treated and whether to install a new rule (flow entry) in the flow table.

3. Drop this flow's packet: this action is used for security reasons such as protecting services from attacks. It also can be used to reduce broadcast discovery traffic.

4. Process this flow's packet by the switch's normal processing pipeline.

The first three actions must be supported by any *dedicated OpenFlow switch* which is a dumb data path entity that forwards packets according to the flow table that is manipulated by a remote controller. The *OpenFlow-enabled switches* are commercial switches that run the OpenFlow protocol and enhanced by a secure channel to communicate with the controller. The flow table is built on the ternary content-addressable memory (TCAM). This type of switches can isolate non-OpenFlow traffic by adding the fourth action.

A flow table is composed of three columns, as shown in figure 2.10. The first column is a packet header that matches the packets belonging to the same flow, each header field can be used as wild card to aggregate flows. If a packet matches many flow entries then the action of the first match is used. If there is no match, the packets are either dropped or forwarded to the controller depending on the switch's configuration. The second column specifies the action that determines how to process the matching packets of a flow (listed previously). The third column holds statistics related to the count and size of packets of a flow, it also saves the time of the last matching packet.

To demonstrate the operation of an SDN network with the OpenFlow protocol, consider a simple network consisting of two switches, two hosts and on controller.

Indeed, assume that, in figure 2.10, host A is sending a packet to host B through switch 1, when the packet arrives to switch 1, the latter tries to match it with one of the flow entries, however if it did not find any match, the packet is forwarded to the controller by default as shown in figure 2.11. Upon receiving the packet, the controller instructs the switch to install a new rule that matches all packets with destination field host B and with the action *forward packet to switch 2* as shown in figure 2.12.

OpenFlow 1.2 and later versions might comprise more than *one flow table*, in addition to a *group table* as illustrated in figure 2.13 [8]. Each flow entry in the

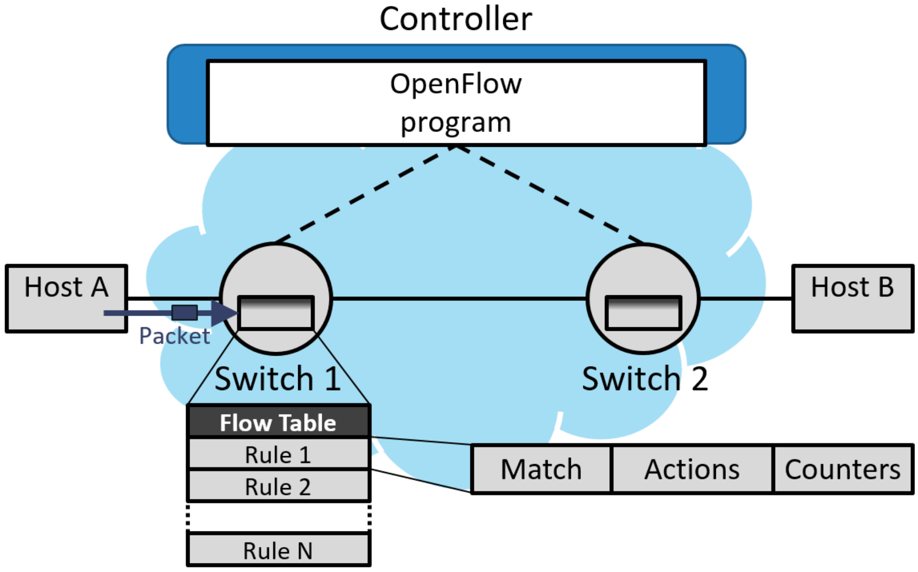

Figure 2.10: Host A sends data to Host B

flow tables is associated with instructions that are composed of either a set of actions or pipeline processing instructions which extends processing packets in a sequence of flow tables. Packets move from on table to another until the actions of the matching entry does not specify a next table.

Some actions may forward packets to group table which is composed of group entries and each group is associated with a list of action buckets. Grouping flows allows additional processing and complex forwarding such as flooding, rerouting and link aggregation. It allows efficiently changing common output across flows (IP forwarding to the same next hope).

2.3 Network Functions Virtualization (NFV)

2.3.1 Definition

NFV changes the architecture of current operators' networks. Current networks are composed of a large collection of proprietary hardware where each is dedicated for a specific function. Adding a new function or a new service will require buying new dedicated hardware-based appliances and providing the needed costly resources, such as: space, power and energy, in order to accommodate these machines, compounded by finding skilled employees to manage the complex configuration of integration of such boxes. Furthermore the hardware life cycle is becoming shorter due to the rapid technology innovation.

As shown in figure 2.14 [9], NFV suggests implementing network functions as

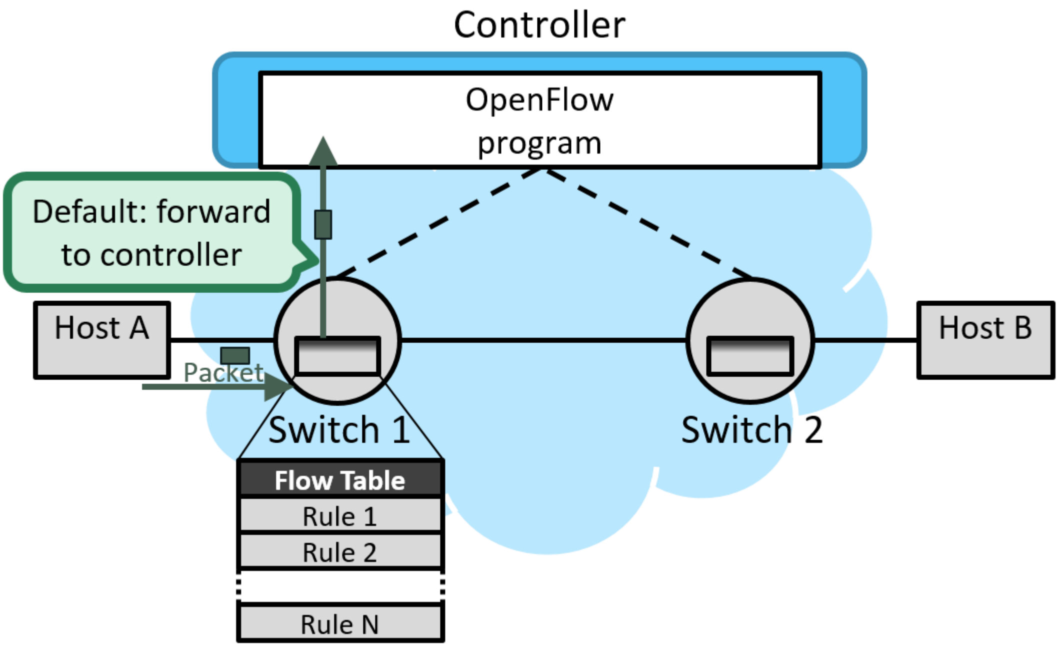

Figure 2.11: Sending data to the controller

software that can be run on powerful industry general purpose servers with the ability of instantiating and moving instances of these functions between different datacenters at different locations as required, without the need of installing new hardware. For example, the architecture and functions of routers, DPIs (Deep Packet Inspections) and routers can be emulated by pieces of software. These virtualized network functions can be instantiated on physical servers or standard high volume IT infrastructure. The orchestrator is the software that monitors and manages allocated physical resources to software instances [10].

NFV and SND share the objective of using industry standard servers and switches. However, they are not dependent technologies, and each can be implemented separately. NFV is a complementary to SDN and both can be combined together to achieve greater value since the SDN idea which is separating control and data planes can enhance performance, introduce flexibility and simplicity when it comes to resolving compatibility issues and maintenance through programmability and centralized control. Also SDN connects the different virtualized functions resulting from applying the NFV concept . In turn, NFV can provide the infrastructure,where the SDN controller can be virtualized and the network functions are implemented as software instances , on which SDN can operate. [9].

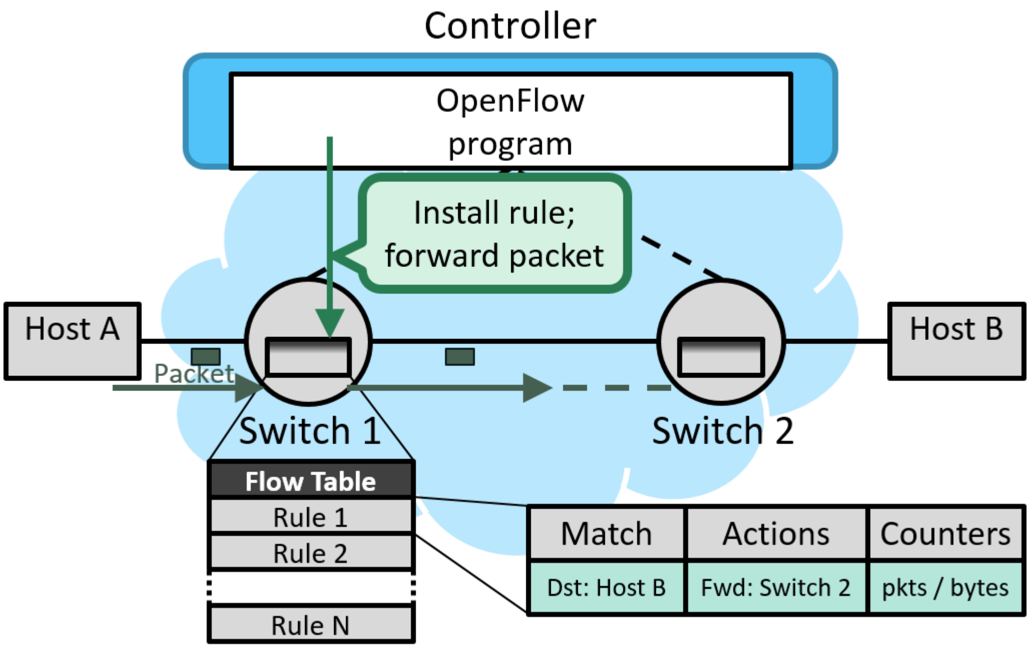

Figure 2.12: Installing a new rule

2.3.2 Fields of application and use cases

NFV can be applied to the data/control plane functions in mobile or fixed networks. There are many examples of NFV use cases that include but not limited to:

- Tunneling gateways.
- Security functions: intrusion detection systems, virus scanners, firewalls, etc.
- Switching nodes.
- Mobile network entities.

2.3.3 Benefits

Adopting NFV brings many benefits to network operators, such as:

- Reducing hardware equipment cost.
- Reducing power and space used.
- Shortening network operator innovation life cycle, and changing evolution mode from hardware-based to software-based.

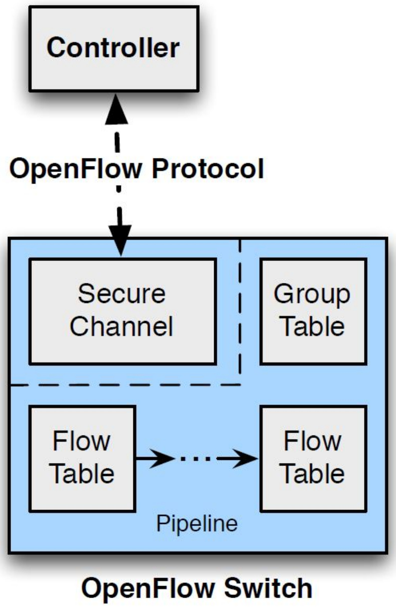

Figure 2.13: OpenFlow switch enhanced with multiple flow tables and a group table

- Improving testing new protocols and policies because NFV enables using the same infrastructure to run production and testing which helps achieving better testing results in shorter time, facilitates integration, and leads to cost saving development.
- Making addition and initiation of services more flexible and without installing new hardware.
- Enabling scaling up/down services in real time.
- Providing the ability to allocate resources optimally; for example using less number of servers during off-peak hours and more servers during peak hours reduces energy consumption
- Rising adoption of echo-systems by boosting virtual appliances market.
- Encouraging participation of small investors and academia to increase innovation rate.
- Supporting multi-tenancy by allowing different network operators to share the same infrastructure with the separation of administrative domains.

Furthermore, applying NFV to mobile core networks targets a more cost efficient deployment, flexible coping with the increasing traffic demands, optimized resource consumption, hardware abstraction (no need to hardware upgrades),

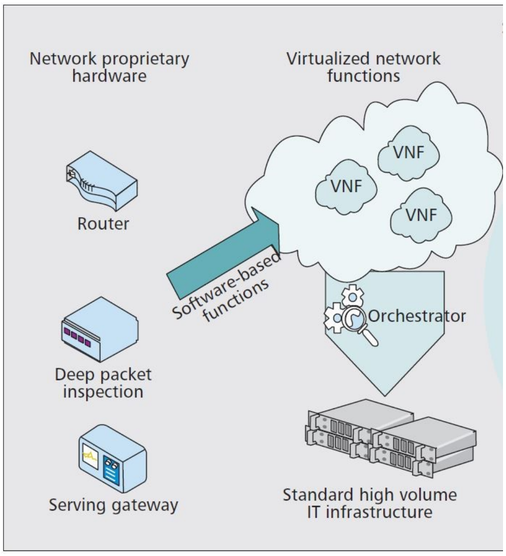

Figure 2.14: Network function virtualization concept; NFV differs from SDN.

easy multi-tenancy support and flexible incremental functional additions without the need to install new hardware [10].

2.3.4 Changing network landscape

NFV introduces major changes to telecom industry landscape. Network operators need to manage moving to the NFV market. This positioning is facilitated due to vendors that are recently implementing their proprietary software on standardized hardware in a proprietary scheme. Standardizing this kind of implementation is the key that will allow network operators to migrate their hardware-based functions to software-based while maximizing existing systems and processes reuse [10].

2.3.5 Enablers for NFV

NFV can be achievable by leveraging recent technologies such as those virtualization mechanisms-based used in cloud computing; the usage of hypervisors for hardware virtualization, linking traffic between virtual machines using virtual Ethernet switches.

Regarding communication functions, packet processing with high-performance are adopted and running on high-speed CPUs, smart Ethernet NICs are used to achieve load balancing and offloading, usage of poll-mode Ethernet drivers, routing packets directly to virtual machine memory.

Cloud computing servers run orchestrators and management systems that automatically instantiate, re-instantiate or migrate virtual network elements and virtual machines, also they manage resource allocation in terms of assigning instances to adequate server physical components. Open APIs such as OpenFlow and OpenNaas [11] may be used to provide control over the data plane.

From an economic perspective, NFV can be implemented of industry standard high volume servers built according to standard IT components. Adopting such servers will result, in the near future, in decreasing demands on Application Specific Integrated Circuits (ASICs), which come at a high cost, in favor of increasing demands on low cost, relatively, general purpose machines [10].

2.3.6 Challenges

Applying the concept of NFV is faced by several challenges that need to be addressed and studied in order to accelerate the development of this concept [10]. The main challenges are:

- Portability/Interoperability: The importance of portability is that it allows freely to set the location and the required resources by virtual instances in an optimal way. Unified interfaces must be defined between different data-centers from different vendors, though standardized, that decouple software from hardware.

- Performance trade-off: Appliances that used to be implemented on specialized hardware, now run on general purpose standard hardware which will result in performance degradation. It is a necessity to use the appropriate software technologies to prevent this degradation from affecting the whole operation output and limit the effect of processing delay, latency and throughput.

- Migration, co-existence and compatibility between legacy and NFV: NFV must support interacting with network operator's legacy equipment and be compatible with current existing management systems. NFV must be able to operate in a hybrid network, running both classical physical appliances and virtual appliances.

- Management and orchestration: Such systems are needed in order to flexibly add new virtual appliances and manage physical resources, this could be done through standardizing and abstracting North Bound Interfaces, which are the interfaces that connect lower level components with higher level components, to be rapidly aligned with management. SDN for example can be used to integrate switches in the network that may be controlled by NFV and its virtual appliances.

- Security and resiliency: Verifying that when introducing NFV, networks are not impaired. Securing virtual appliances as physical ones is a paramount to success. NFV shows great potential to improve resiliency and availability since failing functions can be re-instantiated on demand.

- Simplicity: making sure that introducing NFV will turn current networks architectures to be simpler. Avoiding to replace current network's problems with new NFV-related problems.

- Integration: Seamlessly integrating, using servers and virtual appliances from different vendors.

2.4 Virtualization in LTE

Current market needs of higher data rates, adding more services, guaranteed bit rate, better user experience, and exponential growth of data volume on the transport plane and the demand of faster deployment in addition to the required flexibility in changing services criteria to cope with network dynamics, resulted in decreasing operators' revenues. This decrease motivates researchers for finding new concepts to reduce total cost ownership (TCO). Networks functions Virtualization (NFV) and Software Defined Networking (SDN) are two main concepts that help operators in achieving cost reduction for the favor of enlarging revenues margin.

The LTE EPC has a static fit-to-purpose architecture with dozens of standardized interfaces. Each hardware component in the EPC is dedicated to perform one functionality, which means adding more functionalities will require integrating more hardware entities to the core network. Also accommodating a larger number of users during peek times is done by duplicating entities that will not be used during idle times. Evolving EPC along these lines will result in a more complex architecture and protocols leading to an increase in the capital and operational expenditures (CAPEX/OPEX) while the average revenue per user (ARPU) from a subscriber will be decreasing [12].

In past decades, the traditional architecture has been successful and reliable. But currently, with the introduction of new technologies, such as cloud computing and mobile applications on smart-phones, to the mobile operators, cellular networks technologies are becoming congested and unable to sustain market needs and growth [13].

Diving deeper into the EPC's mechanisms, we notice that tunnel management is centralized in the EPC, where tunnel routing depends on the control of IP routing which is distributed. This results in routing instabilities since not all elements in the network will be able to complete reconfiguration at the same time.

Centralizing all data-plane functionalities in the PGW has many drawbacks. All data traffic must be forwarded to PGWs which will increase congestion and delay. Since PGWs are not modular, adding a functionality to the core network will require the operator to buy whole new PGWs and pay for other functionalities that are not of its interest. Also, an operator cannot integrate capabilities and functionalities from different vendors and manufacturers due to compatibility issues [14, 15].

2.5 Challenges

Applying virtualization on LTE's EPC is not a trivial task. It is faced by many challenges that need a thorough study to overcome. From these challenges, we list the following:

- From architectural perspective, applying the SDN concept on LTE requires designing a robust architecture for carrier-grade flow-based forwarding that provides a wide room for innovation and enhancement at the LTE EPC. Such architecture must preserve all functionalities of the traditional core network after moving to a software based functionalities in order to make the forwarding layer fully software driven. For example, the software emulators of the MME, SGW and PGW must be maintained and connected efficiently to prove the flexibility, programmability and openness of the proposed architecture without introducing changes to the UE. A key success for this architecture is the ability to easily interact with legacy EPC network elements.

- Tunnel management in current EPC deployment is centralized in the MME which is responsible for activating/deactivating bearers. Tunnel routing strongly depends on IP routing which is a distributed task. Hence, after a change in routing, because of a failing element for example, IP controlled routing will take some time to converge. This delay will cause routing instabilities and misrouted packets need to be re-transmitted weakening the QoS of the service being delivered and increasing network congestion.

- Fine-grained service policies are needed to direct traffic through the right middleboxes. In large networks, the number of paths resulting from policies grows exponentially, and can cause the data-plane state to explode. With small switch tables, supporting fine-grained policies becomes a burden. In order to determine which policy clause to apply, packets should be classified at the network edge which is challenging since few gateways must direct traffic to thousands of base stations at line rate. Operators have little control on UE mobility which may cause packet loss. When it comes to service policy, all packets must go through the same sequence of middlebox instances in order to preserve consistency. Therefore, the ability of handling network dynamics may not be efficient.

- LTE/EPC requires guaranteeing always on connectivity service in terms of resiliency and load balancing. Always on connectivity service is about transparently moving active sessions from one network entity to another without interruption. Resiliency is restoring active session after failure of network entity, while load balancing preserves equal amount of load for each network entity during peak time. However, LTE EPC does not provide enough visibility and control elasticity to enable this service.

A lot of work has been done to overcome the aforementioned challenges. Researchers proposed many variations to apply the concept of NFV and SDN to LTE EPC as described in next chapter. Though these new technologies are considered to be the foundation of designing scalable high performance cost efficient mobile cellular networks, a little attention has been given to the impact of introducing NFV and SDN on the the network load and data-plane. Whereas integrating NFV to mobile core networks brings important advantages, it requires steering all data traffic to a datacenter, where the functions are virtualized, which forces additional load on the transport network and imposes longer delay on the data-plane depending on datacenters locations. Since the introduction of SDN comes along with a supplementary control-plane, the load overhead on the network will increase proportionally with the amount of control that SDN is granted.

In this thesis, we survey several related works that addressed the aforementioned challenges. Then we point out the main limitations of the surveyed work. We observed that most of the related work lack analyzing the impact of virtualization on EPC's performance. Our contribution will shed the light on the impact of virtualization in terms of added network load and increased packet delay.

Chapter 3

Related Work

Introducing virtualization to LTE EPC was addressed in many related work by integrating the Software Defined Networking (SDN) and the Network Functions Virtualization (NFV). In this chapter we survey these related work pointing out to their limitations. Each section in this chapter is entitled by the name of the paper that the section describes.

3.1 Moving the Mobile Evolved Packet Core to the Cloud

SDN based EPC architecture was proposed in [16] along with the changes that should be applied on OpenFlow switches and protocol in order to support GPRS Tunneling Protocol Tunnel Endpoint Identifier (GTP TEID) routing. The paper aims to simplify and centralize IP routing by integrating SDN in EPC. Running IP routing protocols in the same controller or datacenter will guarantee converging at the same time and reduces the number of misrouted packets. As a result, several applications that were not possible to implement in traditional EPC with distributed IP routing, now can be enabled when introducing SDN to mobile communication systems such as:

- **Selective Flow Routing for In-Line Services:** Suppose that an operator provides a set of services that run on some application flows. The number and the order of applying these services depend on the application flow. In the standard case of distributed routing, after applying each service, the flow has to go back to the router, which forwards it to the server running the next service to be applied. OpenFlow has the ability to encapsulate and decapsulate GTP packets, present them as simple IP packets, and steer flows between different services without trombone routes back to a router.

- **Multihomed Terminals:** OpenFlow discards the network topology and treats IP addresses as identifiers. Routing happens according to the flow rules instead of the longest prefix match.

- **Security Isolation of Mobile Networks:** After scanning infected terminals by a malware, they are isolated on a different virtual network where they can update or download a self-protection anti-virus system system that removes the malware.

To support GTP TEID routing, the current OpenFlow forwarding table was extended with two additional fields: the GTP TEID field and the GTP header flag that specifies whether the packet must be processed on OpenFlow's fast path GTP TEID routing or on slow path. When a packet is fragmented, the OpenFlow Switch must reassemble the fragments before passing the packet to the flow table because only the first fragment will contain the GTP header. Thus, in order to support GTP routing and hide the complexity of tunneling, a virtual port is added for each physical port that performs encapsulation and decapsulation. These ports are needed in the SGW, PGW and the wired network interface of the eNB. The OpenFlow controller programs these ports via a configuration protocol that must support messages allowing the controller to perform the following functions:

- Verifying if the switch supports GTP fast path virtual ports. If it does, then query which port numbers are for fast path and which are for slow path.

- Instantiating a GTP-U fast path virtual port to be used in the OpenFlow table as an output port, in addition to binding a GTP-U virtual port to a physical port.

The controller handles tunnel management. Indeed, when it receives a GTP-C packet from the gateway control plane, it programs a gateway switch with the correspondent rules and actions. OpenFlow protocol was modified in terms of structures and messages in order to allow matching on TEID and GTP header in addition to adding and deleting TEID parameter table entries.

From architectural perspective, the paper proposed a prototype consisting of a controller composed of computer blades, implementing on top of it the gateways control part and MME are implemented. The data forwarding plane is a collection of enhanced OpenFlow switches. The communication between the two planes is done via the modified OpenFlow protocol as mentioned earlier.

3.2 MobileFlow: Toward Software-Defined Mobile Networks

Software defined mobile networks (SDMN) [17] is a paradigm based on SDN that aims to provide more flexibility in future carriers through programmability to obtain a software-driven forwarding layer. Like in SDN, in SDMN the user plane is decoupled from the control. The main key enablers are MFFE (MobileFlow forwarding engine) and MFC (MobileFlow controller). The OpenFlow paradigm must be also applied to the IP/Ethernet transport network. MFFEs can be used to enable multi-tenancy and virtualization, and support security functionalities and GTP-U encapsulation and decapsulation. MFFEs communicate with the MFC via a lightweight protocol in order to handle control messages and rules installation. MFFEs are more complex than an OpenFlow switch but less complex than legacy EPC entities, they can be deployed by enhancing an OpenFlow switch.

An MFC has 3 interfaces, the southbound interface communicates with MFFEs, the horizontal interface communicates with other MFCs, and the northbound interface is used to apply network services and applications development (MME, GW-C). MFCs include a functional block for mobile network abstraction to perform topology auto-discovery and viewing and monitoring network resources. They also include a network functions block to deal with tunnel processing, charging routing and many other functions. To facilitate 3GPP-IP convergence, MFFEs can be combined with OpenFlow transport switches in order to form one combined forwarding element, the MFC can also be combined with the OpenFlow controller.

Mobile network applications running on top of the MFC can be developed adopting NFV and implementing gateways that communicate with MFFE via MFC (1 to 1 mapping between the virtual gateway and the MFFE); i.e., LTE entities and added value functionalities are implemented as applications. Regarding mobility management, since all 3GPP mobile network control functionality can be implemented as applications on top of the controller, then GTP signaling and data is transformed into flow rules sent to the MFFE via MFC.

3.3 SoftCell: Scalable and Flexible Cellular Core Network Architecture

SoftCell [18] is a network architecture that is able to support fine-grained mobile devices policies in cellular networks. SoftCell is based on two techniques: Multi-dimensional aggregation and smart access edge / dumb gateway edge. Multi-dimensional aggregation is a technique that takes advantages of the traditional location based routing and the tag based routing (e.g. MPLS), which leads to decrease the size of switch flow tables. Aggregation is performed on three dimensions: Policy UE identifier, policy identifier, and local UE identifier. In SoftCell, access edges are responsible of performing packet classification in order to reduce the load on the internet gateway edges. These access edges embed policy identifiers in packets to avoid reclassification of the returned traffic at the gateway edges that perform only simple forwarding.

SoftCell obviates the need of specialized hardware entities (e.g. SGW, PGW), instead it is mainly composed of a controller, access switches, core switches and middleboxes. The controller has access to UE attributes (billing plan, OS version) and uses them in order to install rules in switches in order to direct traffic through middleboxes. The access switches can be software switches implemented in servers. They perform classification. Classifiers for UEs are cashed in local agents in order to minimize the overhead of interacting with the central controller. Core switches are hardware switches that perform matching rules and forwarding to middleboxes while gateway switches are connected to the internet and only perform basic forwarding (cheaper than PGWs). Middleboxes are like virtual machines and dedicated appliances that perform some processing with functionalities (e.g. video transcoder, firewalls), some middleboxes require packets to traverse the same instance in both directions. Service policies are represented as predicates that depends on user attributes to determine how the traffic is steered through middleboxes.

To overcome the challenge of supporting fine-grained service policies, while having small switch tables, the technique of multi-dimensional aggregation is used. Multi-dimensional aggregation combines traditional destination IP address aggregation with tag-based routing in order to scale to a large number of service policies in large networks. In order to determine which policy clause to apply, packets should be classified at the network edge which is challenging since few gateways must direct traffic to thousands of base stations. Since traffic is initiated by UEs, SoftCell performs classification only at the access edges. Classification information are piggybacked so the edge switches only perform basic forwarding.

In order to deal with network dynamics and to make sure that the packets of a flow pass through the same sequence of middle boxes, in SofteCell, after UE handoff, the old flows continue traversing the same path to the old base station while new flows are steered through nearer middleboxes. To reroute old flows, SoftCell establishes long lived connections between nearby base stations. To handle controller failure, SoftCell keeps a copy of the controller state. Upon failure, the controller is replicated and rebuilds correct UE locations by querying the local agents.

3.4 New control plane in 3GPP LTE/EPC architecture for on-demand connectivity service

[19] suggests to apply SDN paradigm in LTE/EPC to overcome the weakness of LTE EPC in guaranteeing always on connectivity service.

It proposed an architecture in which the OpenFlow protocols replace the control protocols running between the eNB and the MME from one side and between the MME and the SGW from the other. Since the MME is dedicated to control functions, it is centralized, along with the intelligence of the SGW (SGW-C), as an application on top of the OpenFlow controller. The SGW data plane (SGW-D) is actually an enhanced OpenFlow switched with GTP Encapsulation\Decapsulation enabled. SGW-C allocate unique TEID values per session and these values remain the same when migrating a session from one SGW-D to another.

Resiliency is crucial in LTE networks in order to ensure reliable connectivity. In the traditional architecture, resiliency is not transparent for the user and requires lot of signaling since active sessions are cut off and not restored until the user sends a service request. In OpenFlow based LTE/EPC architecture, SGW-D failure is detected when exchanging echo request/reply messages. The SGW-C selects a new SGW-D and updates the SGW-D IP address in both PGW and the eNB while TEID's remain the same. In legacy LTE architecture, the MME assigns load to a PGW according to its capacity relatively to other SGWs serving the same area. The drawback of this technique is that it does not take into account the load on SGW in real-time. While in OpenFlow based EPC, due to the ability of getting real-time statistics about the SGW-D load and the session type from packet headers, the controller can perform more efficient load-balancing. The main challenge is to extend the OpenFlow protocol to transparently support the exchange between the MME and the UE. Also the switch must be enhanced to support GTP encapsulation and decapsulation.

3.5 A Virtual SDN-Enabled LTE EPC Architecture: A Case Study for S-/P-Gateways Functions

In LTE's EPC each core network node is deployed in a separate hardware. The SGW and PGW that are responsible for handling of control and data plane were analyzed in [20] to derive their main functions considering the main scenarios in which they are involved such as UE attach/detach, S1-U bearer release, service request (default bearer), service request (dedicated bearer), update to the tracking area and handovers. Based on these scenarios, with respect to NFV, many function categories where derived: control signaling, resource management logic, data-plane forwarding rules, data-plane forwarding, GTP Matching, data-plane filtering and classification in addition to charging Control.

The deployment of these functions based on SDN technology is possible, however to provide each function category, new SDN component framework is needed for additional modules to be integrated to the basic OpenFlow switch. Based on these frameworks, the authors developed four different architectures.

Regarding control-plane related functions, for the control signaling function, a module is needed to be added to the OpenFlow controller to perform signaling management. The centralized switching module which is already a function in the core of every OpenFlow controller can perform the resource management logic, but it needs to be slightly modified to support user profile and policies that are used in resource management decisions.

Regarding data-plane related functions, both data-plane forwarding rules and data-plane forwarding are main operations of a basic OpenFlow controller, but SGW and PGW use GTP to forward data and signaling which is not the case of the basic OpenFlow controller. To support such functions via SDN, four different frameworks are proposed.

1. The first framework implements the GTP function as a controller module, this approach conforms to the OpenFlow operation but it requires every flow packet to be processed by this controller which causes overhead of data.

2. The second framework uses middleboxes that host GTP function so the data go through for additional processing. The difference between the two frameworks is that the middleboxes are deployed next to OpenFlow switches.

3. The third framework enhances the OpenFlow switch in hardware (OF Network Element plus NE+) to support this function . The implementation at the switch can be sometimes beneficial but makes the hardware less flexible.

4. The last framework provides switches with a programmable platform to implement additional functions. This imposes enhancing OpenFlow capabilities to interact with deployed functions (OF Network Element plus with Programmable Software Platform NE+ with SW-platform). The advantage of this approach is increased flexibility.

The packet filtering and classification function can be achieved through matching rules. The filters are matched via the IP five tuple (source IP address, destination IP address, source port, destination port, protocol ID).
Lastly, the Charging function can be deployed using OpenFlow counters and statistics exchanged between the controller and the OpenFlow switch. A module

that collects CDRs (Charging Data Records), based on OpenFlow counters and stats, is needed in case of offline charging. In case of online charging, NE+ or NE+ with software development platform maybe needed because the OpenFlow switch cannot allocate charging events.

These frameworks are related to the OpenFlow aspect, but other aspects such as performance depend on where to place EPC functions and need to be discussed. Four different generic architectures to where the EPC functions are placed either in data centers (operator cloud) or at transport network elements (OF switches), were considered:

1. Full cloud migration architecture where SGW and PGW are virtualized in an operator cloud. Since MME only performs control-plane functions, it is also moved to the cloud. SDN is used to manage signaling and traffic ingoing or outgoing from the cloud in addition to intra-cloud forwarding. This architecture brings advantages such as cost saving in comparison with proprietary hardware, flexible resizing of EPC components, and flexible upgrades. But these advantages are limited by larger scale flexibility that covers the whole network which is not yet exploited for this architecture. Also cloud infrastructure performance is critical since the operations at SGW and PGW are happening at a high frequency with large amount of data.

2. The second architecture separates the S-/PGWs functions between control-plane and data-plane related. It kept the control plane functions in the cloud while it spanned the data-plane on a distributed hardware covering the entire transport core network. Keeping performance in mind, forwarding rules function and data processing functions have to be together on the data-plane elements to avoid passing packets to the cloud. Complying with the SDN platform, this architecture still has the control plane centralized with an API between the resources management logic function and the forwarding rule function. This architecture allows instant migration of the virtualized data-plane depending on traffic volume or services requirements. It also allows offloading in addition to the fact that centralized control plane allow a global view of the network.

3. In a third architecture, only the signaling control function is migrated to the cloud while all other functions reside on data-plane elements. This architecture makes the data plane less dependent on the cloud and more resilient in case of connection failure to the cloud, but it does not full profit from the powerful computing and storage capacities of the cloud. Also the global view and management of the resources is lost.

4. In the fourth and last architecture the authors propose to deploy functions in both cloud and data-plane nodes. The scenarios introduced above are

executed in either deployment scheme according to its requirements in the mean of latency and arrival frequency so the scenarios requiring low latency are directed to the data-plane infrastructure while the scenarios requiring high data processing are directed to the cloud. Synchronization between the two deployment schemes is necessary in order to avoid duplicate assignments which may affect overall performance.

3.6 Applying NFV and SDN to LTE mobile core gateways, the functions placement problem

The effect of virtualization (via NFV) and decomposition between control and data planes (via SDN) on the transport network load overhead and on the data-plane delay was studied in [21]. The main focus was on the functions of the SGW and PGW since they involve the most important data/control plane functions in LTE networks. So the objective is to solve the functions placement problem such that for each gateway, decide whether to virtualize all its function in a datacenter or decompose it with compliance to SDN (i.e. decompose it as a controller operating in a datacenter in addition to a network element at the transport network). Since data-plane delay strongly depends on the path length on which packets travel, the optimal datacenters placement was investigated. Optimally solving this problem will result in minimizing network load subject to predefined data-plane delay, number of potential datacenters and SDN control volume.

Figure 3.1 (a) illustrates the current gateway architecture which is a dedicated hardware entity. Figure 3.1 (b) represents a virtualized gateway according to the NFV concept; all gateway functions are migrated to a datacenter and the gateway is replaced by a standard networking element that directs traffic to the correspondent datacenter. Figure 3.1 (c) represents an SDN decomposed gateway in which control plane is separated from data plane is done by implementing gateway's control functions as a module on top of the controller and replacing the gateway by an enhanced NE that supports data plane functions such as GTP tunneling. The paper assumes that datacenters are placed where an operator already has implemented infrastructure in order to reduce floor space cost.

A demand is a data flow between an SGW and its PGW. Data-plane traffic delay is defined as the sum of propagation delay, T_{prop}, on each link of the packet path between the access network and the Public Data Network (PDN) adding to it the processing delay, T_{proc}, performed by each node on this path. Obviously, T_{prop} depends on the link length, hence the importance of finding datacenters optimal location. Regarding packet processing delay, GTP packet processing is the dominant. For the virtualized gateway, a java GTP packet processor was developed. While for the SDN case, since GTP headers are not in the OpenFlow

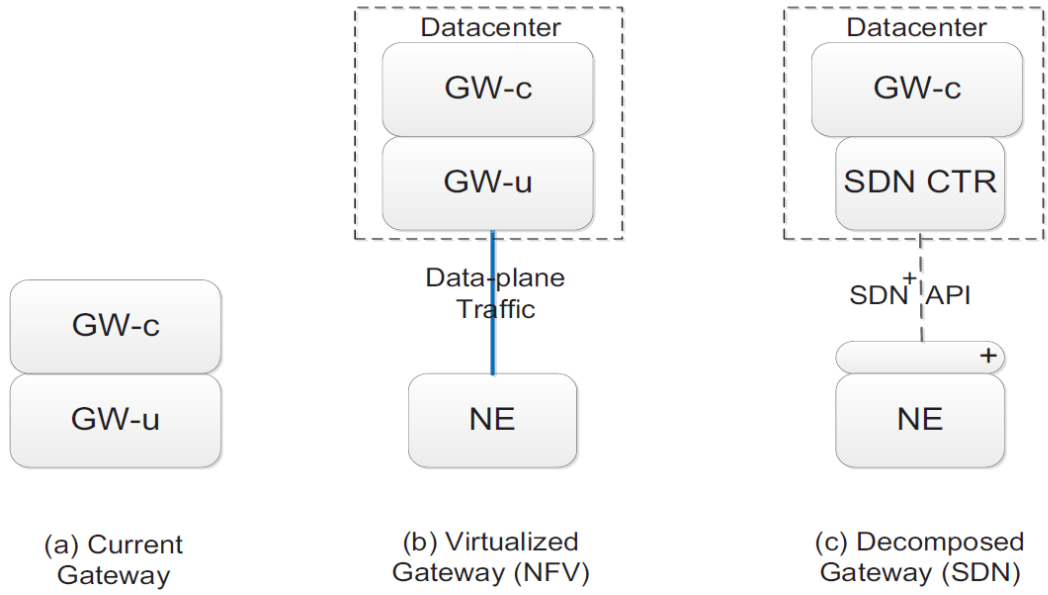

Figure 3.1: Mobile core gateways re-design

matching tuple, GTP processing was emulated using OpenFlow Modify action that modifies header bits. Measurements of processing delay for both cases were performed on different data rates, number of established tunnels and number of packets per second. In the SDN case, packet processing was faster than the virtualized case which is expected since packet processing in hardware is faster than processing in software.

According to this architecture, a demand has four possible paths: between a virtual SGW and a virtual PGW, between a virtual SGW and a decomposed PGW, between a decomposed SGW and a virtual PGW, or between a decomposed SGW and decomposed PGW. Adding to these paths choosing a possible datacenter location. So the selection of a path for a given demand result in locating the datacenter and will decide on the function placement for the involved SGW and PGW. The total network load is the sum of the data-plane traffic and SDN control traffic multiplied by path length.

The problem can be formulated as an optimization problem where the objective is to find each gateway function placement and select datacenter locations in a way that minimizes network load:

$$\text{minimize} \quad \sum_{c \in C} \sum_{d \in D} \sum_{p \in P} \delta_{c,p,d} N_{c,p,d} \tag{3.1}$$

Where C denotes the set of all possible K datacenters locations, D the set of demands, and P the set of four paths. $\delta_{c,p,d}$ is a binary variable that is set to one if the path p is chosen for the demand d with the datacenter in location c. $N_{c,p,d}$ is a pre-calculated load for the combination $c \in C$,$p \in P$ and $d \in D$. This objective is subject to the following constraints:

$$\begin{array}{rcll} \sum_{c \in C} \delta_c & = & K & \\ \sum_{p \in P} \delta_{c,p,d} & \leq & \delta_c & \forall d \in D, c \in C \\ \sum_{c \in C} \sum_{p \in P} \delta_{c,p,d} & = & 1 & \forall d \in D \\ \sum_{p \in P} \delta_{c,p,d} L_{c,p,d} & \leq & L_{budget} & \forall d \in D, c \in C \end{array} \tag{3.2}$$

The first constraint assures that K datacenters are under operation where δ_c is a binary variable that determines whether a datacenter c is selected. The second constraint ensures the possibility of selecting a path $p \in P$ if a datacenter c is chosen for a demand d. The third constraint forces the selection of a single path p and a single datacenter c for each demand d. Traffic delay budget satisfaction is guaranteed by the last constraint, the chosen function placement, i.e. a path p, of a demand d with datacenter c must remain under a certain pre-calculated threshold.

The traffic that affects network originates from data traffic between SGW and PGW, and the control packets sent by SDN controllers. The Volume of SDN control is dependent on the protocols used by the operator and the customization added in order to implement some functions.

3.7 SDN and NFV Dynamic Operation of LTE EPC Gateways for Time-varying Traffic Patterns

NFV and SDN were applied in [22] on the main LTE core components handling control and data planes, SGW and PGW. The aim of the paper is to find the optimal data centers places, running virtualized gateways, to minimize the total network load taking into consideration the variation of traffic patterns with respect to time, subject to data-plane affordable delay constraints. Also power saving models according to datacenters available resources and traffic-patterns fluctuations were provided.

From an architectural perspective, in this approach there is only one path type (unlike the previous approach where there are four types) because there is only one gateway architecture that is a virtualized gateway hosted by a data center and replaced by an SDN NE as represented in figure 3.2 [22]. NFV gives more flexibility in resource allocation by transforming gateways to software instances running on standardized hardware, while SDN gives more flexibility in control since it allows steering traffic coming from access networks to different datacenters and changing traffic routs on demand.

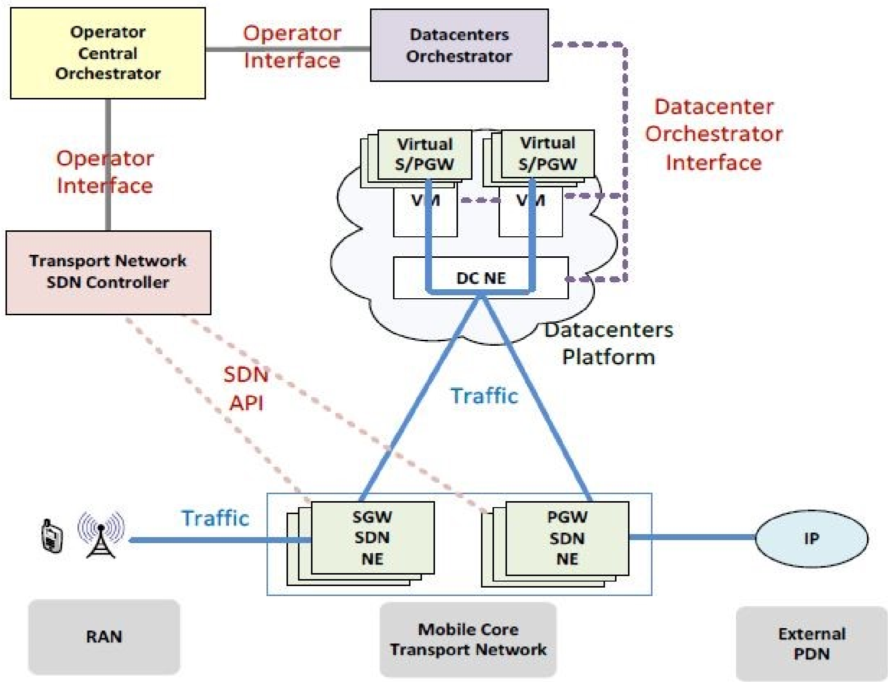

Figure 3.2: Architecture of virtual mobile core gateways and SDN transport NEs

With the integration of these technology to LTE's core, additional controllers and orchestrators are needed. In figure 3.2 that represents the architecture of virtual mobile core gateways and SDN transport network elements; datacenters Orchestrator (DC-O) manages the resources allocated for each gateway instance in order to achieve a performance as high as a hardware gateway. Also it handles synchronization and migration of virtual instances in addition to connections between physical hosts within the datacenter. Transport SDN controller (SDN-C) controls network elements to route traffic dynamically on runtime to the appropriate datacenters according to the installed rules. The Operator Central Controller (OCC) interfaces with the DC-O and SDN-C in order to enforce operator's requirements. It is responsible of network changes as dimensioning the network, adding new instances, load balancing, shutting down parts of the network for the sake of saving energy.

Current cellular networks lack flexibility unlike the dynamic nature of users demands. Since such behavior is not being considered, network entities alternate between periods of overuse and periods of idleness. The inability to scale up or down the network infrastructure to fit the traffic volume decreases operator revenues. Hence the importance of studying traffic patterns and its correlation with time and location. In order to determine the traffic pattern of a city whose demands are sent to the same sgw, we need to quantify its intensity which is a function of population and time slots. The traffic of a city c with population $p(c)$ at a time t and intensity $i(t)$ can be calculated as:

$$f(c,t) = i(t) \times p(c) \tag{3.3}$$

The traffic at each sgw is the sum of all cities traffic connected to it:

$$TR_{sgw}(t) = \sum_{c \in C} f(c, time_{c,sgw}(t)) \times b_{c,sgw} \tag{3.4}$$

where $time_{c,sgw}(t)$ is a function that calculates the local time of a city c depending on the local time of the sgw. $b_{c,sgw}$ is a binary function that determines whether a city c is connected to the sgw or not.

The total network load is an important metric to decide on network dimensioning since it is directly related to the traffic delay and the cost imposed on the operator. The total network load is defined as:

$$\sum_{t \in T} \sum_{d \in D} Tr_{d,t} \times lengthPathe_{d,t} \tag{3.5}$$

where t is a time slot, and again d is a demand between each SGW and PGW. $Tr_{d,t}$ is the demand d traffic volume at time t, while $lengthPathe_{d,t}$ is the length of the path taken by the demand d at the time t which spans the distance between an SGW NE, a datacenter and a PGW NE. Thus, the chosen path results in locating the correspondent datacenter and assigning it a demand moreover calculating the data plane delay on the path. This approach also considers that a datacenter may be placed in a location where the operator already has a deployment.

As mentioned before, three models were proposed; the first model finds the optimal datacenters placement and assign them demands at each time slot, the remaining two models deal with power saving.

The first model can be formulated as a minimization problem:

$$\text{minimize} \quad \sum_{c \in C} \sum_{d \in D} \sum_{t \in T} \delta_{c,d,t} N_{c,d,t} \tag{3.6}$$

where C is the set of all possible K datacenters locations, D the set of demands and T the set of time slots. $\delta_{c,d,t}$ is a binary variable that is set to one if at time t, the demand d is assigned for the datacenter in location c. $N_{c,d,t}$ is a pre-calculated load for the combination c, d and t. This objective is subject to the following constraints:

$$\begin{array}{rcll} \sum_{c \in C} \delta_c & = & K & \\ \delta_{c,d,t} & \leq & \delta_c & \forall d \in D, c \in C, t \in T \\ \sum_{c \in C} \delta_{c,d,t} & = & 1 & \forall d \in D, t \in T \\ \delta_{c,d,t} L_{c,d,t} & \leq & L_{budget} & \forall d \in D, c \in C, t \in T \end{array} \tag{3.7}$$

The first constraint assures that K datacenters are under operation where δ_c is a binary variable that determines whether a datacenter c is selected. The second constraint ensures the possibility of assigning a demand d at a time t for a chosen datacenter c. The third constraint forces the selection of a single datacenter c at a time t for each demand d. Traffic delay budget satisfaction is guaranteed by the last constraint, the resulted latency of assigning a demand d to a datacenter c at time t must remain under a certain pre-calculated threshold.

The Second model, which aims to minimize the power consumption, is also formulated as a minimization problem of the total network load at each time slot. It takes the set of chosen datacenters from the first model and allows operating less than k datacenters which will result in decreasing power consumption and operating less hardware boxes. So the problem is formulated as:

$$\text{minimize} \quad \sum_{c \in C_s} \sum_{d \in D} \delta_{c,d,t} N_{c,d,t} \quad \forall t \in T \tag{3.8}$$

where C_s denotes the set of chosen datacenters from the first model. The first constraint of the first model is changed to allow operating less than k datacenters as follows:

$$\sum_{c \in C} \delta_c \quad \leq \quad K \tag{3.9}$$

To ensure operating at least one data center and that the resources required to handle the assigned demands for each datacenter at each time slot, $R_{d,t}$, do not exceed its available resources, R_c, the following constraints are added:

$$\begin{aligned} \sum_{c \in C_s} \delta_c \quad & \geq \quad 1 \\ \sum_{d \in D} \sum_{t \in T} \delta_{c,d,t} R_{d,t} \quad & \leq \quad R_c \quad \forall c \in C_s \end{aligned} \tag{3.10}$$

The difference between the second and third models, is that the latter allows for a room to exceed the available resources by some factor P, so the last constraint of the second model is changed to:

$$\sum_{d \in D} \sum_{t \in T} \delta_{c,d,t} R_{d,t} \quad \leq \quad R_c \times P \quad \forall c \in C_s \tag{3.11}$$

3.8 Comparison and observations

Table 3.1: Rubrique table of the work related to applying virtualization on EPC.

Feature	[20]	[16]	[17]	[18]	[19]
NFV on S/PGW	✓				
GTP	✓	✓	✓		✓
Routing details		✓		✓	
OpenFlow 1.0	✓				✓
OpenFlow 1.2	✓	✓			
Custom OpenFlow		✓	✓	✓	
Original switch	✓				
Switch with software development platform	✓				
Enhanced switch	✓	✓	✓	✓	✓
Virtual ports (encapsulation/decapsulation)		✓			
Full cloud migration architecture	✓				
Control-plane cloud migration architecture	✓	✓	✓	✓	✓
Signaling control cloud migration architecture	✓				
Scenario based cloud migration architecture	✓				
Policy tagging				✓	
Legacy infrastructure			✓		
Multi-dimensional aggregation				✓	
Smart access edge, dumb access edge				✓	
Always On connectivity					✓

Table 3.1 compares the key features of the surveyed work. We can observe that all of them applied a control-plane cloud migration architecture. Therefore all of them have used enhanced switches in order to support data-plane functions, mainly the GTP protocol. However many of them did not specify the changes that must be made to the OpenFlow protocol to support these additional functions.

Only [16] and [18] delved into the routing details when introducing virtualization. As for the remaining features, it is clear that each work focused on a specific topic. Hence, all of them failed to provide a complete framework for applying virtualization on LTE's EPC.

The impact of virtualization on LTE EPC is an important issue since it affects the network load and the packet delay. Although this issue is influential and decisive on whether it is convenient to apply these technologies, none of the work listed in table 3.1 pointed to it.

Table 3.2: Rubrique table of the work related to studying the impact of virtualization on LTE's EPC

Feature	**[21]**	**[22]**
NFV on S/PGW	✓	✓
SDN on S/PGW	✓	
Uniform demands	✓	✓
Time-dependent demands		✓
Population-dependent demands		✓
Bearers characteristics dependent demands		
Propagation delay	✓	
Processing delay	✓	
Network load quantification	✓	✓
Datacenter available resources		✓
Power saving		✓

Table 3.2 presents the main features of two related work [21, 22] that studied the impact of virtualization on LTE's EPC. Both [21] and [22] proposed applying NFV on SGW and PGW, however only [21] suggested decomposing the gateways between control plane and data plane using SDN. Both papers quantified and minimized the network load, however [21] assumed that the demands are uniform while [22] was more precise and it considered time-dependent and population dependent demands. [21] provided details about propagation delay and processing delay. [22] proposed models that take into consideration datacenter available resources and save power. None of the two related work considered the characteristics of the bearers being established despite the fact that the bearer packet delay budget and resource requirements may differ depending on the QCI of the bearer and its other QoS parameters.

In [21], demands are considered to be uniform and should not vary with respect to time and intensity of the area of deployment. This assumption is a major limitation in the approach. In fact, demands are strongly correlated with time variations and intensity of the area being served. In other words, gateways are more loaded during peak times than off-peak times, moreover demands are more frequent in cities than in suburbs, i.e, the traffic load in cities during peak times is much higher than in other time slots.

Another limitation is not taking into account datacenter available resources and locations. Some datacenters may be overloaded in comparison with others, therefore assigning functions to these overloaded datacenters will certainly affect the total load in the network and will increase data plane delay.

The paper did not mention how to choose the number of solved data centers, nor the strategy to set the SDN control volume nor how to calculate the load for a certain combination.

The approach in [22] only considers full virtualization of a gateway; i.e., all gateway's functions are implemented in a datacenter and the gateway is replaced by a basic SDN networking element that steers traffic to different datacenters. The drawback of exclusively considering such architecture is the impact on delay-critical functions. With the increasing propagation delay to transfer large volume of traffic in and out datacenters, some function wont be able to meet the required QoS.

Both approaches only incorporate S/P-GW functions, other important mobile network functions such as the MME must also be studied.

Chapter 4

Proposed Solution

Current core network infrastructure lacks flexibility. Network operators are unable to cope optimally with traffic fluctuation so their networks are over-utilized during peak hours and under- utilized during off-hours. Network function virtualized is a technology that is able to provide flexibility in resource allocation while Software defined networking allows more flexibility in control. These technologies give the operator more granularity when it comes to dimensioning and managing the network to adapt to traffic patterns changes with respect to time and achieve better load balancing and power saving. In this chapter, we propose a hybrid architecture where on each gateway both SDN decomposition and NFV virtualization are applied. We also use the LTE QCI in order to determine the delay budget of each set of bearers. Then we formulate the problem as an optimization problem and we finish by explaining how to calculate the problem parameters.

4.1 Basic Concepts

The application of NFV allows network operators to take advantage of the cloud technologies in term of cost saving in comparison with proprietary hardware components. In addition it provides high computational power, flexible dimensioning, optimal resource utilization, rapid upgrade and new functionality integration. Yet it may increase packet delay since each packet must traverse the link from the forwarding element to the data center where the functions are virtualized. Also packet processing in hardware is usually faster than packet processing in software. In this chapter, virtualization or virtualized network gateway means applying the concept of NFV.

Applying SDN decomposition on the gateway by implementing its control part in the data center while spanning the data plane part on advanced networking elements with heavy packet processing abilities, avoids forwarding all packets to the datacenter which decreases the end to end delay. Instead, it offloads traffic on distributed components while keeping a global network view by the centralized gateway control plane. Nonetheless, the application of SDN decomposition adds a control layer which might increase the total network load causing more congestion, packet loss and energy consumption.

A demand between an SGW and a PGW comprises bearers of different quality of service requirements. Some bearers belong to delay critical applications while some others might belong to high bit rate and heavy packet processing applications. On the other hand, SDN deployment increases the network load but it decreases the network delay, while NFV deployment decreases the network load. Load is decreased with NFV since it does not contribute to additional network layers but it increases the delay. Therefore depending on the nature of data flows the decision of which technology to use must be made carefully.

In previous work, [21] and [22], the whole gateway (all its control and data plane functions) is either virtualized or decomposed. To make the most of the two technologies, we will adopt a hybrid approach in which each gateway is virtualized and decomposed at the same time, as described in the next section. This architecture gives a more granular control over the network. In our proposed work, for each demand, depending on the nature, type and requirements of its data flows or bearers and the network state, we choose whether to activate the virtualized deployment or the SDN decomposed deployment. For delay critical data flows it might be better to use SDN decomposed deployment since the latter decreases the delay,while for the other data flows NFV deployment might be a better choice since it decreases the network load. Thus there is a trade off between the advantages of each architecture and loss of optimality. We think that adopting this hybrid architecture will result in a more optimal resource allocation and maximizes the benefits of using SDN and NFV. In addition, the approach provides flexibility in changing the assignment of data flows to either deployment. For example, if at time $t1$ the flows of the same requirements belonging to a certain demand are assigned to the virtualized deployment, they may be assigned to the decomposed deployment at time $t2$. This makes the proposed approach adapt quickly and at any time to the network conditions.

4.2 Proposed Architecture

Figure 4.1 represents the architecture of proposed hybrid SGW and PGW. In this figure, each gateway is replaced by a networking element (NE) connected to a datacenter and interconnected between them. The left NE replaces an SGW, while the right NE replaces a PGW. $L1$ is length of the path between the NE replacing the SGW and the datacenter, $L2$ is the length of the path between the two NEs and $L3$ is the length of the path between the NE replacing the PGW and the datacenter.

The application of SDN decomposition on a gateway necessitates enhancing the NE from an ordinary data forwarding element to support the gateway data

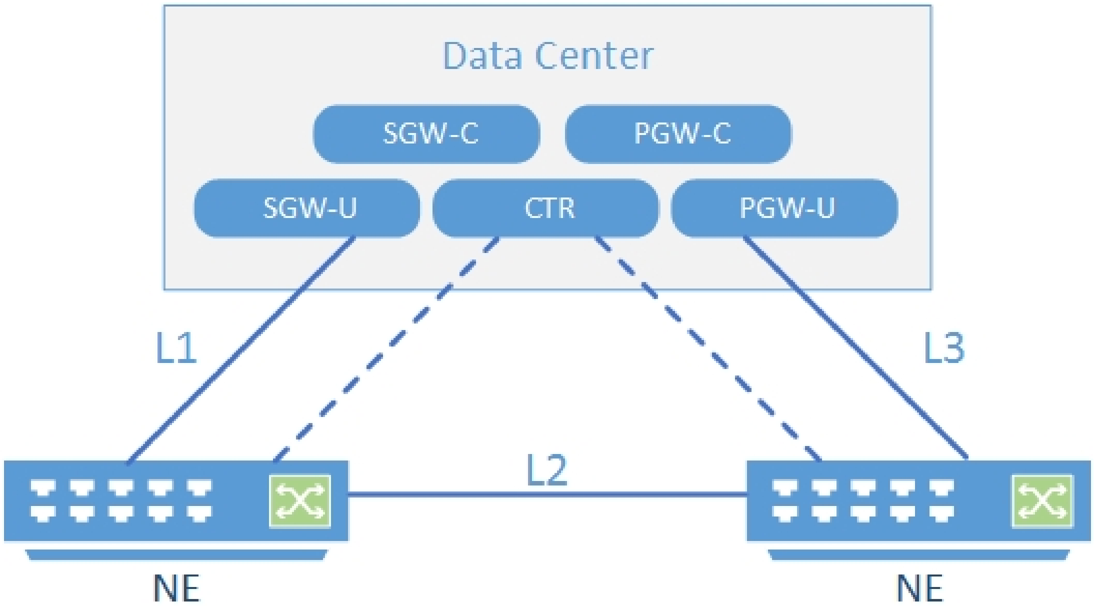

Figure 4.1: Hybrid architecture

plane functions, such as GTP. Regarding the control plane functions of each gateway, they are implemented as software instances in the datacenter, namely SGW-C and PGW-C. The NEs are connected to the control plane instances via the SDN controller (CTR), (which also resides in the datacenter) via the dashed links which are dedicated to transport control messages and exchange flow tables rules (no user data on this link).

As for the NFV case, both data plane (SGW-U and PGW-U) and control plane (SGW-C and PGW-C) functions of each gateway are implemented as software instances in the datacenter while the NEs are used only to forward packets from/to the datacenter and between them. The NEs are connected to the user plane instances in the datacenter via links depicted as solid lines, that transport user data plane packets.

A hybrid architecture can be achieved by implementing a single control plane instance for each gateway since both of SDN decomposition and NFV require running the control part as software application in the datacenter. While there are two implementations for the data plane functions, one in the datacenter that will be activated in case of NFV deployment, while the other is implemented in the NEs and will be activated in the control plane element. Such an architecture with combined technologies provides flexibility in shifting between one and another depending on the network state. Considering an SGW deployed in a crowded city, on Monday afternoon, most likely the NFV deployment is used in order to minimize the network load, however this does not imply that the virtualized deployment will be used at all times. The hybrid architecture will allow the activation of the SDN decomposed deployment during the night to shorten the average packet delay.

4.3 QoS in LTE

The data flows should be classified since each has different requirements in terms of delay and bit rate, thus each has its impact on the network load. Therefore, the different flows cannot receive the same treatment when choosing between virtual and data plane deployments.

Section 2.1.5, described how data is transferred in the EPC between the UE and the PDNs using bearers. It was also mentioned that each bearer is associated with a certain QoS depending on the service data flows that constitute the bearer. This section dives more into the details of the QoS assigned to bearers and its relation with the network load.

Each data flow carries the packet flow of a certain application between the UE and the PDN. The application requirements in terms of delay, error rate and data rate are specified in the QoS associated to the bearer. Thus the network load can be inferred from studying the QoS parameters [23] that need to be guaranteed for each established bearer in the network. These parameters can be described as following:

- *The QoS Class Identifier (QCI)* is the most important QoS parameter. It is an 8-bit number that maps the bearer to four values: the resource type, packet error/loss rate, packet delay budget, and QCI priority. *The resource type* indicates whether the bearer must be a GBR or a non-GBR bearer to have this QoS class. *The packet error/loss rate* determines the upper bound of the proportion of lost packets to the total number of packets due to transmission and reception errors. *The packet delay budget* is the upper bound for packet transfer delay between the UE and the PDN. *The QCI priority* is used for scheduling purposes. Indeed, in congested networks, the delay budget for bearers with QCI priority p is met before bearers of QCI priority $p+1$. Some QCI classes are standardized [24] to receive consistent QoS when roaming. A network operator can also configure customized classes for non-roaming mobiles. The mapping between Standardized QCI classes and correspondent quantities are listed in Table 4.1 [23].

- Each bearer is associated with another QoS parameter, the *allocation and retention priority* (ARP), which has three values: The *ARP priority level* that defines the priority to establish or modify a bearer in congested networks, the *pre-emption capability* that specifies whether the bearer can take resources from another bearer belonging to a lower priority. The *pre-emption vulnerability* that specifies whether the bearer can give resources to other bearers with higher priority.

QCI	Resource type	Priority	Packet delay budget	Packet error loss rate	Example services
1	GBR	2	100 ms	10^{-2}	Conversational voice
2		4	150 ms	10^{-3}	Conversational video (live streaming)
3		3	50 ms	10^{-3}	Real time gaming
4		5	300 ms	10^{-6}	Non-conversational video (buffered streaming)
5	Non-GBR	1	100 ms	10^{-6}	IMS signaling
6		6	300 ms	10^{-6}	Video (buffered streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7		7	100 ms	10^{-3}	Voice Video (live streaming) Interactive gaming
8		8	300 ms	10^{-6}	Video (buffered streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
9		9			

Table 4.1: LTE standardized QCI characteristics

- The *guaranteed bit rate* (GBR) and *maximum bit rate* (MBR) parameters determine respectively the lower and upper bounds of bit rate that a GBR bearer can have.
- For non-GBR bearers, the *UE aggregate maximum bit rate* (UE-AMBR) and the *APN aggregate maximum bit rate* (APN-AMBR) parameters determine the maximum total bit rate from all non-GBR bearer per UE and per APN respectively [25].

4.4 Problem Formulation

In this section, we formulate the problem as an optimization problem where the objective function is to minimize the total network load by finding the optimal datacenters placement at each time slot for each QCI of the bearers established for each demand under the constraint of a certain packet delay budget. Optimal datacenter placement means that at each time slot, for each set of bearers having the same QCI belonging to a demand, we must find the optimal location of the datacenter and choose which deployment to activate. Complying with a certain delay budget. The possible datacenter, locations are where the operator already has a deployment.

We adopt the QCI standard classes in order to classify the data flows of each demand since it helps to identify the threshold of packet delay and to estimate the traffic volume when combining it with other QoS parameters, namely the MBR, GBR, UE-AMBR and APN-AMBR.

Before we proceed in the formulation, we define the following notations:

- Q: the set of standardized QCI values ranging from 1 to 9.
- C: the set of of datacenter locations.
- D: the set of demands.
- T: the set of time slots.
- P: the set of paths.
- K: the number of datacenters.

In our model, Q is the set of QoS standardized classes that can be assigned to bearers in LTE's core network, where each class defines its own values as seen in the previous section. To establish and maintain bearers, we execute a set of functions that will be eventually needed to decide on selecting the right deployment will process them.

P is the set of all possible paths that a set of bearers might take between the SGW and the correspondent PGW. In total there are four possible paths:

1. Between a virtualized SGW and a virtualized PGW.
2. Between a virtualized SGW and a decomposed PGW.
3. Between a decomposed SGW and a virtualized PGW.
4. Between a decomposed SGW and a decomposed PGW.

Our goal is to minimize the total network load by choosing for each QCI q of bearers established on demand d, a datacenter c with a path p at each time slot t. The problem can be formulated as:

$$\text{minimize} \quad \sum_{q\in Q}\sum_{c\in C}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P}\delta_{q,c,d,t,p}N_{q,c,d,t,p} \tag{4.1}$$

where $\delta_{q,c,d,t,p}$ is a binary variable that is set to one if at time t, the bearers of QCI q of the demand d are assigned for the datacenter in location c on the path p. $N_{q,c,d,t,p}$ is a pre-calculated load for the combination q, c, d, t and p. The constraints of this minimization problem are:

$$\begin{aligned}
\sum_{c\in C}\delta_c &= K \\
\sum_{p\in P}\delta_{q,c,d,t,p} &\leq \delta_c \quad \forall q\in Q, d\in D, c\in C, t\in T \\
\sum_{c\in C}\sum_{p\in P}\delta_{q,c,d,t,p} &= 1 \quad \forall q\in Q, d\in D, t\in T \\
\sum_{c\in C}\sum_{p\in P}\delta_{q,c,d,t,p}L_{q,c,d,t,p} &\leq L_{budget} \quad \forall q\in Q, d\in D, t\in T
\end{aligned} \tag{4.2}$$

The first constraint assures that K datacenters are under operation where δ_c is a binary variable that determines whether a datacenter c is selected to be under operation. The second constraint ensures that in case a datacenter c is chosen, then a path $p \in P$ can be selected for bearers of QCI $q \in Q$ that are established for demand d at time slot t. The third constraint forces the selection of a single path p and a single datacenter c for each QCI q on demand d at each time slot t. Traffic delay budget satisfaction is guaranteed by the last constraint; for QCI q's bearers of demand d at time slot t, the delay produced by choosing a datacenter c and path p, namely $L_{q,c,d,t,p}$, must remain under q's delay budget.

4.5 Calculating problem parameters

The next step is to quantify the pre-calculated problem parameters, namely the network load N and the latency, for each combination of QCI q, datacenter location c, demand d, time slot t, and path p.

4.5.1 Calculating network load

Similar to [22], the traffic of a city $ct \in CT$, where CT is the set of considered cities, at a time, t, can be represented as the product of the intensity at time, t, $i(t)$, and the population of the city, $p(ct)$: $f(ct, t) = i(t) \times p(ct)$ (equation 3.3). The traffic, TR, at an SGW, which is equivalent to the traffic caused by a demand since a demand is defined between each SGW and its PGW, is expressed as: $TR_{d,t} = TR_{SGW}(t) = \sum_{ct \in CT} f(ct, time_{ct,SGW}(t)) \times b_{ct,SGW}$ (equation 3.4), where $f(ct, time_{ct,SGW}(t))$ is the traffic of city ct at time t, and $b_{ct,SGW}$ is a boolean value that is set to one if and only if city ct is covered the considered SGW. However this formula ignores the fact that the demand is composed of bearers belonging to different QCIs having different characteristics; hence not all of them have the same impact on the total network load. Also the formula does not take into account the extra load added by the control plane when adopting a path p where one or both gateways are decomposed.

In our formulation, we modify equation (3.4) to reflect the impact of each QCI, q, on the network load by integrating the average bit rate of bearers belonging to q, denoted as BR_{avg_q}. Not mentioning the load added by SDN control, so far the traffic generated by an SGW (i.e. by a demand) for a given QCI q at a time t, can be expressed as:

$$TR_{q,d,t} = (\sum_{ct \in CT} f(ct, time_{ct,sgw}(t)).b_{ct,sgw}) + BR_{avg_q} \tag{4.3}$$

where $TR_{q,d,t}$ is the traffic of QCI q generated by the SGW of demand d at time t. $TR_{q,d,t}$ depends on d and t because the traffic of an SGW is the traffic between the SGW and the correspondent PGW at a time t. It also depends on q because we have added the bit rate of the QCI q. We add the bit rate to equation 3.4 in order to account for the considered QCI. Therefore the traffic of two different QCIs will have two different values. The QCI with greater bit rate will have a greater traffic value. The multiplication between the QCI's bit rate and the city traffic may also be used but we prefer the addition to avoid large numbers in the equation. But in both cases, the traffic will increase with the bit rate.

To account for the load added by the control plane of the SDN decomposition, we need a coefficient α denoting the SDN control volume percentage of the traffic generated by an SGW. This percentage depends on the protocol adopted by the operator. It can be calibrated empirically by the operator to determine the best value. The load added by SDN control plane depends also on the chosen path because when choosing a path where both gateways (the SGW and the PGW) are decomposed, the amount of control messages is roughly double the amount when a single gateway is decomposed. When none of the gateways is decomposed, the SDN control messages are absent. Considering and SDN control volume percentage $\alpha = 0.1$ and a path p where only one gateway is decomposed. To account for the load added by SDN control plane, the traffic generated by an SGW must be multiplied by $1 + 1 \times 0.1$, however if two gateways are multiplied, it must be multiplied by $1 + 2 \times 0.1$. If no decomposition, it must be multiplied by $1 + 0 \times 0.1$, i.e., the traffic wont be affected. Therefore depending on the chosen path p and the SDN control volume α, the traffic must be multiplied by a coefficient $\beta_{p,\alpha}$, where β can be expressed as:

$$\beta_{p,\alpha} = 1 + \gamma \times \alpha \tag{4.4}$$

where $\gamma = 0$ in case the path p has no decomposed gateway, $\gamma = 1$ when only one gateway in the path p is decomposed, $\gamma = 2$ if two gateways are virtualized in the path p.

Now the traffic volume generated by the bearers of QCI q, taking the path p, on the demand d, at a time t, can be expressed as:

$$TR_{q,d,t,p} = (\sum_{ct \in CT} f(ct, time_{ct,sgw}(t)).b_{ct,sgw} + BR_{avg_q}) \times \beta_{p,\alpha} \tag{4.5}$$

$TR_{q,d,t,p}$ is the traffic volume of all bearers of demand d having QCI q. BR_{avg_q} is the average bit rate of of all bearers constituting demand d and having a QCI q, this value can be computed based on MBR and GBR value. $\beta_{p,\alpha}$ represents the added traffic of the SDN control plane depending on the control percentage α and the chosen path p. Paths that involve SDN decomposition have higher $\beta_{p,\alpha}$.

Now the network load for the combination q, c, d, t, p (recall that $c \in C$ is the set of datacenters locations) can be expressed as:

$$N_{q,c,d,t,p} = TR_{q,d,t,p} \times length_{c,d,p} \tag{4.6}$$

where $length_{c,d,p}$ is the length of the path p for demand d passing through the datacenter at location c (i.e. the length of the path the packets cross). Calculating the length of the path of a demand d with a datacenter at location c strongly depends on the chosen path p. Recalling from figure 4.1, $L1$ is the distance between the NE replacing the SGW of demand d and the datacenter at location c, $L3$ is the distance between the NE replacing the PGW of the same demand d and the datacenter at the same location c, while $L2$ is the distance separating the two NEs. Depending on the path p, $length_{c,d,p}$ can be calculated as:

- If both SGW and PGW are virtualized, then the communication between the NEs will happen through the datacenter the length of the path is: $length_{c,d,p} = L1 + L3$.
- If the SGW is virtualized and the PGW is decomposed then the NE replacing the SGW needs to communicate with the data center when receiving packets while the communication between the NEs happens through the direct link since PGW is decomposed and therefore the data plane de[ployed in its NE is activated. Therefore the length of the path is: $length_{c,d,p} = 2L1 + L2$.
- If the SGW is decomposed and the PGW is virtualized then the length of the path is the same as in the previous case, however $L3$ will be multiplied by two instead of $L1$: $length_{c,d,p} = 2L3 + L2$.
- If both SGW and PGW are decomposed, then the communication between the NEs will happen directly between them, then the length of the path is: $length_{c,d,p} = L2$.

We abstracted the path lengths between the gateways and used euclidean distances because, after extensive search, we could not put hands on a real deployment depicting the routing process between gateways. However the routing must have the same pattern between all gateways, and we used euclidean distance abstraction on each path which wont affect the selection of a path over another.

4.5.2 Calculating network delay

T_{proc} is the time needed to process the packet at each network node, i.e., at the SGW and the PGW. This value varies depending on whether the gateway is virtualized or decomposed. Typically, processing in hardware is faster than processing in software, so when the gateway is virtualized, its processing delay is greater than when it is decomposed. The number of bearers established at the gateway at each time slot affects the processing delay such that the more you have packets in the queue, the more time a packet must wait before being served. Therefore T_{proc} is subject to: 1) the demand since it specifies the involved gateways and the cities connected to them which affects the number of established bearers, 2) the chosen path since it determines whether each gateway is virtualized or decomposed, and 3) the time slot since the number of active bearers varies according to time . In [21], values of T_{proc} for virtualized gateways and decomposed gateways were estimated. Decomposed gateways have a constant processing delay regardless of the number of established bearers while the processing delay for virtualized gateways increases with increasing number of bearers. These values will be adopted in simulation in next chapter.

T_{prop} is the propagation delay time on each link between the SGW and the PGW. T_{prop} is subject to: 1) the demand since it determines the involved gateways and hence their location, 2) the datacenter location, and 3) the chosen path because it determines how to calculate the length as seen in the previous section. The propagation delay can be expressed as:

$$T_{prop} = \frac{pathlength}{propagationspeed} \tag{4.7}$$

The total network delay between the access edge, i.e. SGW, and the IP domains gateway, i.e PGW, can be expressed as the sum of the packet processing delay T_{proc} and the packet propagation delay T_{prop}:

$$L_{c,d,t,p} = T_{proc_{d,t,p}} + T_{prop_{c,d,p}} \tag{4.8}$$

Chapter 5

Simulation and Results

Simulation and results are presented in this chapter. We first describe the simulation environment and the topology of the core network that we tested our model on, then we detail the measurement of the problem parameters and inputs, specifically the network load and the traffic delay. Finally we analyze the output and runtime of different test cases.

5.1 Simulation environment

The optimization problem was implemented using the java interface of the Gurobi optimization problem solver [26]. For simulation, we developed a java framework to depict the US mobile core gateways based on the US LTE coverage map [27] and the US population [28] since it is hard to access a real deployment topology of a mobile core operator. The mobile core topology shown in figure 5.1 is composed of 4 PGWs, represented as red rectangles, and 18 SGWs, represented as green rectangles, clustered as shown in the figure. The connection between an SGW and a PGW is represented as a blue line. There exists a demand between each SGW and the PGW to which it is connected, therefore in total we have 18 demands. The numbers in the figure are the ids of the gateways. This topology is similar to the one presumed in [21] and [22]. We used their map in order to know where to locate the network gateways. Then we measured the distances between the gateways using the Measure Distance Map on FreeMapTools [29]. Also we assume that each two gateways can be connected together (i.e. the network is meshed). This assumption is important so when a PGW is selected as a datacenter, the other gateways can still connect to it.

The optimization problem parameters, mainly $N_{q,c,d,t,p}$ and $L_{q,c,d,t,p}$, are calculated for all combinations of QCIs, datacenters locations, demands, time slots and paths. They are computed before running the optimization problem solver in order to decrease its running time and gain the ability to run it and retrieve its results instantly.

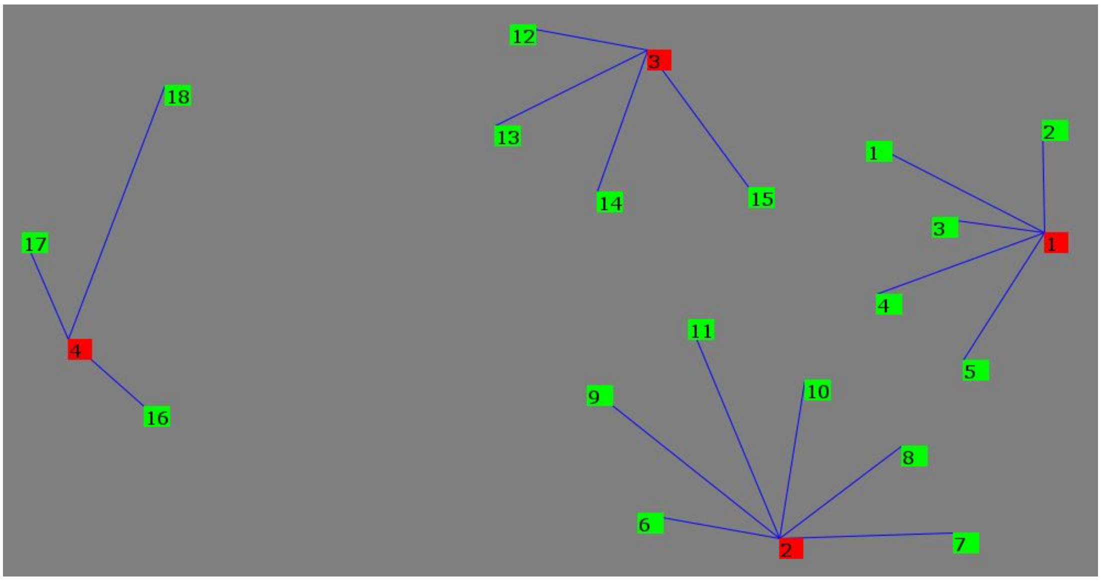

Figure 5.1: The US mobile core Network

Regarding the size of the optimization variabl δ, it is a five dimensional matrix because we have five parameters, namely q, c, d, t and p. However, for better performance this matrix is a actually a vector where its size is the multiplication of the size of each parameter as shown later. In LTE, there are 9 QCIs so the size of Q is 9. There are 22 possible datacenter locations (18 SGWs and 4 PGWs) so the size of C is 22. There are 18 demands (18 SGWs) so the size of D is 18. We cover the 24 hours of the day hence the size of T is 24. Finally there are 4 possible paths for each demand as described in the previous chapter therefore the size of P is 4. Multiplying the size of each parameter will result in the δ's vector length: $9 \times 22 \times 18 \times 24 \times 4 = 342\,144$. For example, When running the optimization problem with this vector of variables at 12:00 am, we will obtain, for the next 24 hours, the optimal combinations of q, c, d, t and p. However, in order to minimize the runtime, we can minimize the number of variables by running the optimization problem at each time slot since the length of the variables vector will be multiplied by 1 instead of 24, and thus δ's vector length becomes: $9 \times 22 \times 18 \times 1 \times 4 = 14\,256$.

5.2 Traffic and Network Load Measurement

To calculate the total traffic of each SGW, we need the population size of all cities that are in the vicinity of the considered SGW. The values we used are presented in table 5.1. Using the US coverage map, we summed the population sizes of all cities situated in the vicinity of each SGW to obtain the SGW's population. To obtain the network usage intensity over a day, we extracted from graph in figure 5.2 [30] the intensity values for each one hour time slot. Having values for population and time intensity, we can now compute the first part of 4.5 which is $TR_{q,d,t,p} = (\sum_{ct \in CT} f(ct, time_{ct,sgw}(t)).b_{ct,sgw} + BR_{avg_q}) \times \beta_{p,\alpha}$ (i.e. the summation part which is equivalent to function 3.4 which is $TR_{sgw}(t) = \sum_{c \in C} f(c, time_{c,sgw}(t)) \times b_{c,sgw}$).

SGWs	sgw1	sgw2	sgw3	sgw4	sgw5	sgw6
Population	680,250	11,440,424	1,976,619	1,759,733	1,599,774	5,833,211
SGWs	**sgw7**	**sgw8**	**sgw9**	**sgw10**	**sgw11**	**sgw12**
Population	2,025,195	668,249	818,308	1,300,875	467,055	468,479
SGWs	**sgw13**	**sgw14**	**sgw15**	**sgw16**	**sgw17**	**sgw18**
Population	4,997,604	655,819	3,295,240	7,326,590	1,089,464	1,772,282

Table 5.1: Population of each SGW

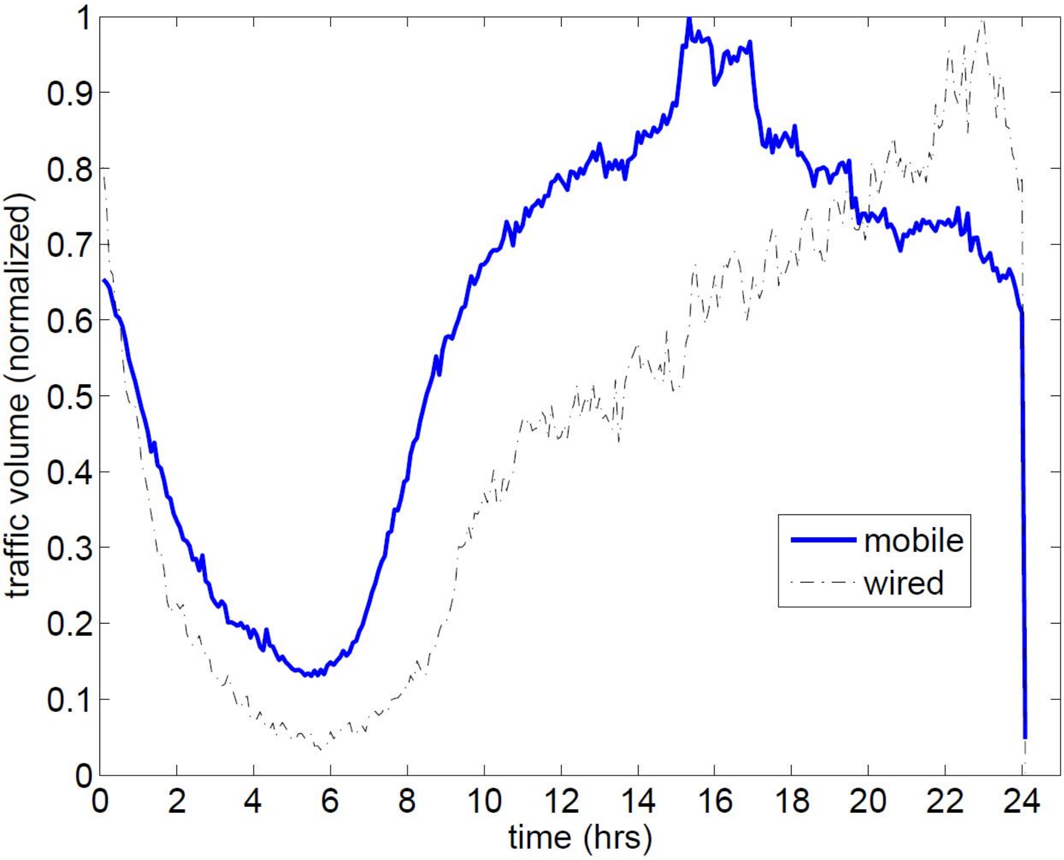

Figure 5.2: Traffic intensity at each hour over one day

Applications	Services		Traffic attributes	
			Average session duration	Mean service bit rate
Existing applications	Voice (multimedia and low rate data/ conversational)			64 kbit/s
	Video phone (medium multimedia/ conversational)			384 kbit/s
	Packet	E-mail (very low rate data/background)		1 kbit/s
		Video mail (medium multimedia/background)		512 kbit/s
		Mobile broadcasting (high multimedia/streaming)		5 Mbit/s
		Internet access (high multimedia/background)		10 Mbit/s
Town monitoring systems	Voice (multimedia and low rate data/ conversational)			64 kbit/s
	Video communication (medium multimedia/conversational)			384 kbit/s
	Medium rate data transmission for town information monitoring (super-high multimedia/interactive)			384 kbit/s
	Low rate data transmission for reservation of restaurants, etc. (very low rate data/interactive)			1 kbit/s
	File transfer (super-high multimedia/ background)			50 Mbit/s

Table 5.2: Traffic attributes of typical applications and services

To compute equation (4.5), we need to estimate the average bit rate of each QCI q. Table 4.1 shows typical example services for each QCI value, while Table 5.2 shows traffic attributes, namely the mean service bit rate, of such applications. Based on these values, we estimated the proper bit rate of each QCI value as shown in Table 5.3. For example, a typical application of QCI 4 is buffered video straming (Table 4.1) which is similar to Internet access services and file transfer services which have a mean service bit rate of 10Mbps and 50Mbps respectively (Table 5.2), so we approximated QCI 4 average to 20Mbps.

QCI	**1**	**2**	**3**
BR_{avg}	64kbps	384kbps	16kbps
QCI	**4**	**5**	**6**
BR_{avg}	20Mbps	1kbps	19Mbps
QCI	**7**	**8**	**9**
BR_{avg}	384kbps	20Mbps	20Mbps

Table 5.3: Estimated bit rate of each QCI value

To finish with equation 4.5, we need to calculate $\beta_{p,\alpha}$, which is the coefficient that accounts for the population p and the SDN control volume α, as specified in equation (4.4). We already specified how to calculate γ when we explained equation (4.4), however we will set the value of α to 10% which is the SDN control volume. This value is determined by the operator, in [21] they also used 10%.

To be able to compute the network load, $N_{q,c,d,t,p}$, for each combination, we computed the path length, $length_{c,d,p}$, for each combination and based on the US core network that we presumed.

5.3 Delay Measurement

5.3.1 Actual Delay Measurement

As mentioned in the previous chapter, the traffic delay between the mobile core gateways is the sum of packet propagation delay, T_{prop}, and packet processing delay T_{proc}. Computing T_{prop} is straight forward, a direct application of the equation (4.7). Regarding T_{proc}, we use the same values used as in [22] and presented in Table (5.4). The values in the table show that there is a correlation between the number of established bearers and the processing delay for a virtualized gateway. Indeed, higher number of bearers results in a longer processing delay. However,the processing delay for decomposed gateways remains constant.

A question may be raised about how to specify the number of bearers for each QCI value of each demand at each time slot in order to use the correspondent T_{proc} value. First of all, the total number of bearers must be generated, to generate a number of bearers of each QCI accordingly. Well, the total number of established

no. of Tunnels	10	100	1 K	10 K
bits/sec	1 M	10 M	100 M	1 G
packets/sec	83	830	8.3 K	83 K
Virtualized GW T_{proc}	62 μs	83 μs	109 μs	132 μs
Decomposed GW T_{proc}	15 μs	15 μs	15 μs	15 μs

Table 5.4: Processing delay average

bearers depends on the population of the cities within the demand's gateway coverage range and the intensity of the considered time slot. For example, in crowded cities, at peak times, the number of established bearer is higher than at off peak times. Also the number of bearers established in cities, is higher than its value in suburbs. Therefore, we use equation (5.1) to compute the total number of bearers at time t given a population pop:

$$nbOfBearers_{t,pop} = \frac{intensity(t) \times pop}{\sigma} \tag{5.1}$$

where σ is a parameter used to normalize the output of the equation, it can be determined empirically. In our test case we used the value $\sigma = 500$. We chose this value because it gave numbers of bearers of different orders (i.e 10, 100, 1 K or 10 K) for SGWs of different order of populations. We determined it empirically by trying different values of σ, and we found that this value gives number of bearers proportional to the number of population.

To generate a number of bearers for each QCI of each each demand we divide the total number on the nine QCIs. To do so, we split the total number, obtained in equation (5.1), of bearers into nine random numbers that sum up to the total number, and assigned each QCI value only one of these random numbers.

5.3.2 Delay Budget

The delay budget is the tolerance of the total delay (processing delay and propagation delay) between the gateways. LTE specifies delay budgets of each QCI as shown in table 4.1, however the considered values are the delay budgets between he user's UE and the server running the service, this is why the values are relatively high in comparison to the fixed delay budget value used in [21] which is $5.3ms$. In our simulations we used a fixed value of $4.94856ms$, this value was determined empirically on the presumed core network topology, in a way that any smaller value will cause the fourth constraint of the model (4.2) to be met, resulting in infeasible model. Then we did a relaxation on that value, so we used delay budget values for each QCI proportional to its correspondent in table 4.1 but normalized to the order of the fixed value. In the table we have the values $50ms$, $100ms$, $150ms$ and $300ms$, we map them respectively to the values $4.94856ms$, $5.0ms$, $5.1ms$ and $5.3ms$.

5.4 Analysis

5.4.1 One datacenter

In this section we will study the results considering one single data center. Figure 5.3 shows the topology of the US core network after running the optimization problem at time slot 15 (between 3:00 pm and 4:00 pm) where the intensity is at its highest value (0.86). Table 5.5 is a numerical representation of figure 5.3 to help understanding and analyzing it. The chosen datacenter is PGW with id 2 circled in orange. The figure also shows the path taken by each demand for QCI 3. We observe that all the QCIs of all the demands took the same path, so for simplicity we only depicted the paths of QCI 3. The path of each demand is represented with a different color. For the SGWs that were not originally connected to the PGW selected as datacenter, the path goes from the SGW to the datacenter then back to the PGW that it is originally connected to it. For example, in figure 5.3 the path of the demand generated by SGW 18 goes to PGW 2, which is selected as a datacenter, then goes back to PGW 4 which SGW 18 is originally connected to. For the SGWs that were originally connected to the PGW that is chosen as the datacenter there is a single line going from the SGW to the PGW. This is, for example, the case of SGW 11, in figure 5.3, where the traffic goes directly to the PGW that is connected to, PGW 2, which is select as a datacenter. Regarding the types of paths, each demand was either virtualized (both gateways are virtualized), represented by solid line, or decomposed (both gateways are decomposed), represented by dashed lines. We concluded that the farthest SGWs from the datacenter (the SGWs that belong to other PGWs than the one chosen as the datacenter) have taken the full decomposed path, while the nearest SGWs have taken virtualized path. This result is expected and can be justified as following. The demands from the farthest SGWs will face a higher propagation delay, therefore to remain within the delay budget, the compensation happens by choosing the decomposed path since it requires lower processing delay than the processing delay required by a virtualized path. Two questions may be asked here. The first is why the other demands did not take a decomposed path since the decomposed gateways path saves delay. Well the answer is that the objective function aims to minimize the total network load and the decomposed paths put more load on the network than the virtualized paths. The distance of the nearest SGWs can handle the extra delay caused by virtualized paths in the favor of minimizing the total load by avoiding decomposed paths. The second question is why the single chosen datacenter happens to be PGW 2 instead of an SGW that falls in the center of the network, SGW 11 for instance. Well, again the goal of the objective function is to minimize the total network load, and a path from the SGW to the PGW goes through the datacenter. In the case of SGW 6, 7, 8, 9 and 10, the path from the datacenter to the PGW is eliminated because the datacenter is the PGW itself which minimizes the total network load without violating the constraints. Choosing an SGW as a datacenter, will double the path length of the aforementioned SGWs which will increase the total network load.

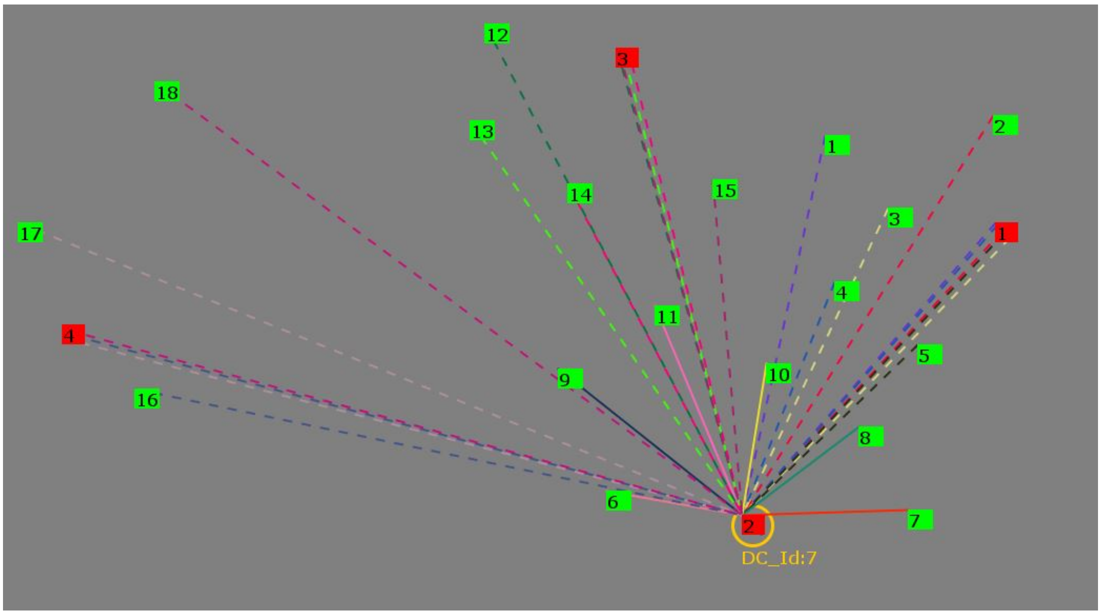

Figure 5.3: The new US core network topology at time slot 3:00-4:00 pm, for QCI 3 and with one datacenter

SGW	Distance to datacenter	Path type
1	1477.74	Decomposed
2	1791.5	Decomposed
3	1289.58	Decomposed
4	954.85	Decomposed
5	928.52	Decomposed
6	526.1	Virtualized
7	629.25	Virtualized
8	555.0	Virtualized
9	895.67	Virtualized
10	580.33	Virtualized
11	861.92	Virtualized
12	2109.92	Decomposed
13	1821.34	Decomposed
14	1422.53	Decomposed
15	1280.9	Decomposed
16	2362.28	Decomposed
17	2971.68	Decomposed
18	2778.92	Decomposed

Table 5.5: Summary of figure 5.3 as numerical values

5.4.2 Multiple datacenters

In this section we will study the effect of increasing the number of datacenters and the changes made on the core network topology. Beginning, with two datacenters, figure 5.4 shows that the new datacenter is PGW 1. Table 5.6 is a numerical representation of figure 5.4 to help understanding and analyzing it. The reason why this PGW is chosen instead of PGW 3 or 4 is related to the population of the SGWs in its vicinity. PGW 1's SGWs are much more populated than the others because they serve big cities like Boston and New York. Therefore this PGW is taken as a new datacenter allowing its SGWs to connect to it taking virtualized paths in order to minimize the total network load as much as possible.

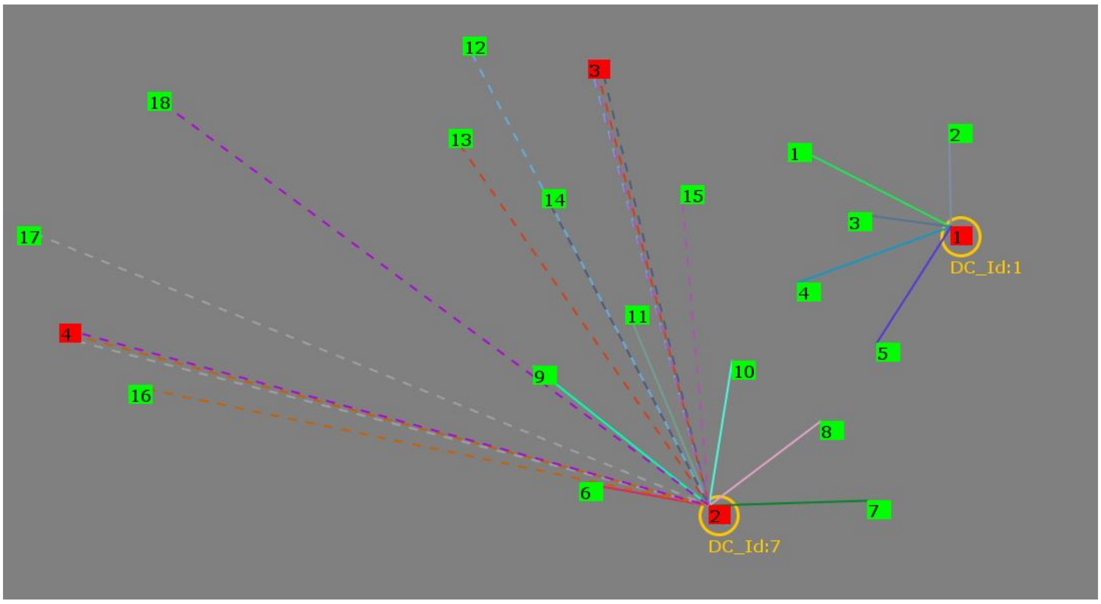

Figure 5.4: The new US core network topology at time slot 3:00-4:00 pm, for QCI 3 and with two datacenter

SGW	Distance to DC1	Distance to DC2	selected DC	Path type
1	729.01	1477.74	1	Virtualized
2	407.09	1791.5	1	Virtualized
3	412.68	1289.58	1	Virtualized
4	650.08	954.85	1	Virtualized
5	549.53	928.52	1	Virtualized
6	1796.3	526.1	2	Virtualized
7	1141.64	629.25	2	Virtualized
8	933.83	555.0	2	Virtualized
9	1755.06	895.67	2	Virtualized
10	1022.38	580.33	2	Virtualized
11	1332.76	861.92	2	Virtualized
12	2085.62	2109.92	2	Decomposed
13	2035.32	1821.34	2	Decomposed
14	1634.62	1422.53	2	Decomposed
15	1086.75	1280.9	2	Decomposed
16	3335.04	2362.28	2	Decomposed
17	3719.0	2971.68	2	Decomposed
18	3245.56	2778.92	2	Decomposed

Table 5.6: Summary of figure 5.4 as numerical values

In figure 5.5 we still have two datacenter, however it represent the topology state at 5:00-6:00 am for QCI 7 which has a higher bitrate and delay budget than QCI 3. We notice that some demands that were passing through datacenter 7, now are passing through datacenter 1. This is because during the time slot 5:00 6:00 am the load on datacenter 1 decreases, so demands generated from SGWs like 12 and 13 may use datacenter 1, shortening the length of their paths for the favor of minimizing total network load. Table 5.7 is a numerical representation of figure 5.5 to help understanding and analyzing it.

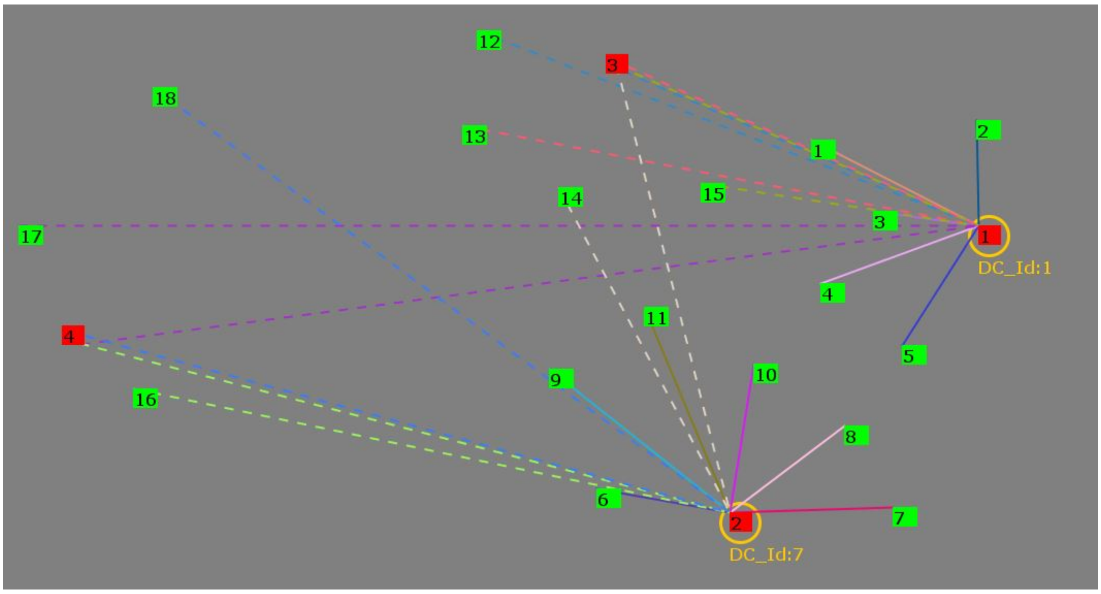

Figure 5.5: The new US core network topology at time slot 5:00-6:00 am, for QCI 7 and with two datacenter

SGW	Distance to DC1	Distance to DC2	selected DC	Path type
1	729.01	1477.74	1	Virtualized
2	407.09	1791.5	1	Virtualized
3	412.68	1289.58	1	Virtualized
4	650.08	954.85	1	Virtualized
5	549.53	928.52	1	Virtualized
6	1796.3	526.1	2	Virtualized
7	1141.64	629.25	2	Virtualized
8	933.83	555.0	2	Virtualized
9	1755.06	895.67	2	Virtualized
10	1022.38	580.33	2	Virtualized
11	1332.76	861.92	2	Virtualized
12	2085.62	2109.92	1	Decomposed
13	2035.32	1821.34	1	Decomposed
14	1634.62	1422.53	2	Decomposed
15	1086.75	1280.9	1	Decomposed
16	3335.04	2362.28	2	Decomposed
17	3719.0	2971.68	1	Decomposed
18	3245.56	2778.92	2	Decomposed

Table 5.7: Summary of figure 5.5 as numerical values

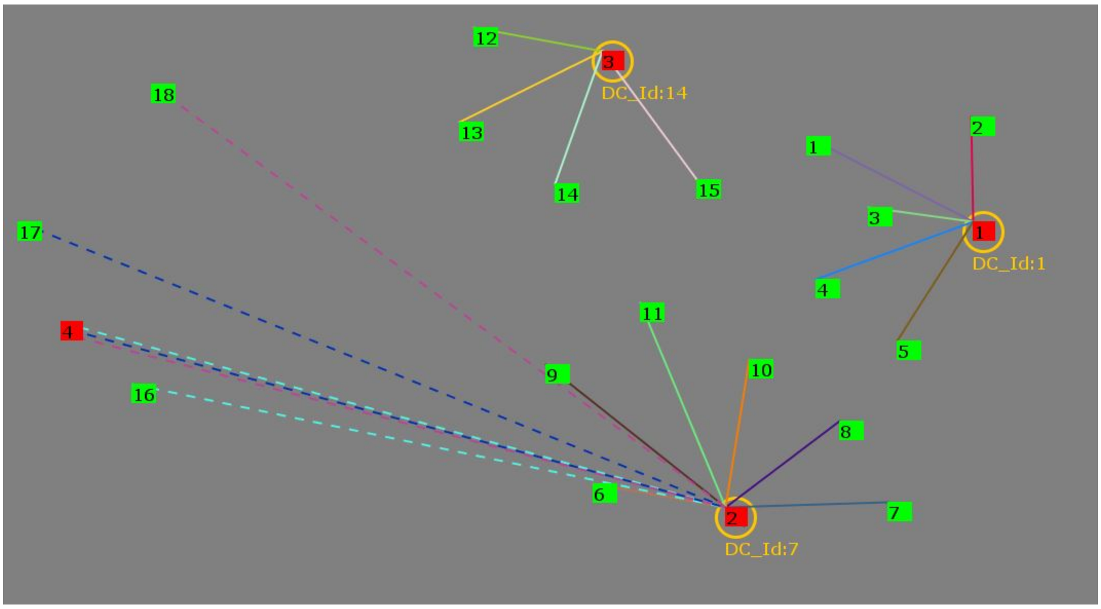

Figure 5.6: The new US core network topology at time slot 3:00-4:00 pm, for QCI 3 and with three datacenter

SGW	Distance to DC1	Distance to DC2	selected DC	Path type
1	729.01	1477.74	1	Virtualized
2	407.09	1791.5	1	Virtualized
3	412.68	1289.58	1	Virtualized
4	650.08	954.85	1	Virtualized
5	549.53	928.52	1	Virtualized
6	1796.3	526.1	2	Virtualized
7	1141.64	629.25	2	Virtualized
8	933.83	555.0	2	Virtualized
9	1755.06	895.67	2	Virtualized
10	1022.38	580.33	2	Virtualized
11	1332.76	861.92	2	Virtualized
12	2085.62	2109.92	3	Virtualized
13	2035.32	1821.34	3	Virtualized
14	1634.62	1422.53	3	Virtualized
15	1086.75	1280.9	3	Virtualized
16	3335.04	2362.28	2	Decomposed
17	3719.0	2971.68	2	Decomposed
18	3245.56	2778.92	2	Decomposed

Table 5.8: Summary of figure 5.6 as numerical values

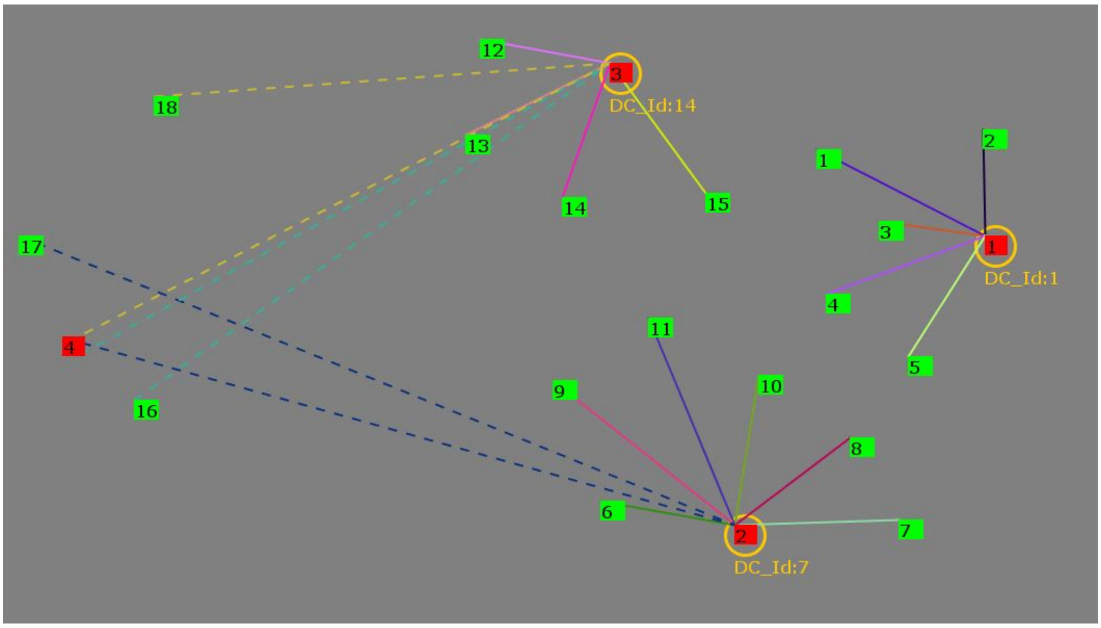

Figure 5.7: The new US core network topology at time slot 5:00-6:00 am, for QCI 7 and with three datacenter

SGW	Distance to DC1	Distance to DC2	selected DC	Path type
1	729.01	1477.74	1	Virtualized
2	407.09	1791.5	1	Virtualized
3	412.68	1289.58	1	Virtualized
4	650.08	954.85	1	Virtualized
5	549.53	928.52	1	Virtualized
6	1796.3	526.1	2	Virtualized
7	1141.64	629.25	2	Virtualized
8	933.83	555.0	2	Virtualized
9	1755.06	895.67	2	Virtualized
10	1022.38	580.33	2	Virtualized
11	1332.76	861.92	2	Virtualized
12	2085.62	2109.92	3	Virtualized
13	2035.32	1821.34	3	Virtualized
14	1634.62	1422.53	3	Virtualized
15	1086.75	1280.9	3	Virtualized
16	3335.04	2362.28	3	Decomposed
17	3719.0	2971.68	2	Decomposed
18	3245.56	2778.92	3	Decomposed

Table 5.9: Summary of figure 5.7 as numerical values

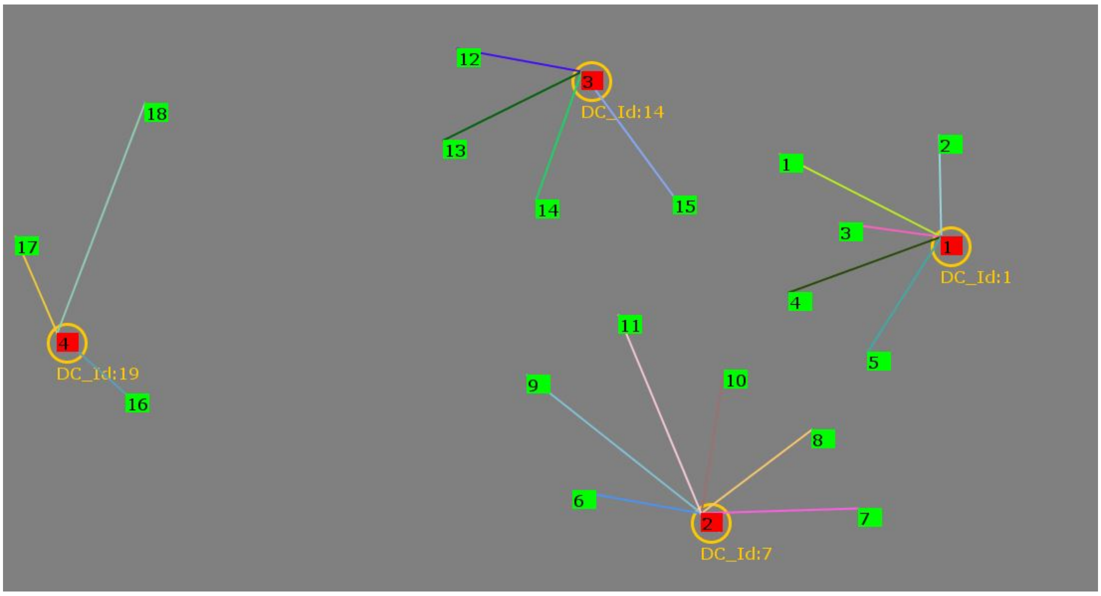

Figure 5.8: The new US core network topology at time slot 3:00-4:00, for QCI 3 and with four datacenter

We added a third datacenter as shown in figure 5.6. PGW 3 was chosen as the datacenter. Figure 5.7 is also with three datacenters but at a different time slot and different QCI. Tables 5.8 and 5.9 are a numerical representation of figures 5.6 and 5.7 respectively, to help understanding and analyzing them. The justification uses the same reasoning as for the case of two datacenters. As in the case of a single datacenter, we conclude that also for two and three datacenters, the nearest SGWs take a virtualized path while the farthest take a decomposed path.

For four datacenters, as seen in figure 5.8, we notice that each PGW is selected as a datacenter and all its SGWs are connected to it via virtualized path. Each SGW is near to its datacenter, so we have minimal propagation delay, therefore all the paths are virtualized in order to decrease total network load despite that the processing delay will increase but without exceeding the delay budget. We also notice that we only have virtualized paths, there are no decomposed ones, so we do not have any extra control layer. What is remarkable here is that the path of each demand is directly from the SGW to its PGW which is its datacenter as well so the path length is the same as in traditional core networks. Therefore, in this case, the total network load is equivalent to the total network load in traditional core network gateways.

5.4.3 Studying the effect of changing the number of used datacenters

Effect on total network load

At a specific time t, the value of the objective function is the minimal network load. Figure 5.9 shows how the value of the objective function varies with respect to time for one, two, three and four datacenters. First thing to notice is that the curves of different values of k have the same shape which is similar the curve's shape in figure 5.2. Between the two intervals time slot 0 - 6 and time slot 15 - 23 we notice that the load for the four cases decreases while it increases in the interval time slot 6 - 15. This behavior is similar to the behavior of the intensity value in figure 5.2, which means that the total network load and intensity are correlated, when the intensity increases, the load increases and when the intensity decreases, the load also decreases.

Another thing to notice is that, the four curves are stacked above each other without intersecting. Over all the time axes, the total network load is the highest when using one datacenter while it is the lowest when using four datacenters. The fact that the network load is the lowest with four datacenters could be predicted since we have seen in figure 5.8 that with four datacenters (each deployed where we had a PGW) there is not any decomposed path, so we do not have any extra control plane load to the original traffic load. This observation drives us to conclude that number of used datacenters and the total network load are inversly

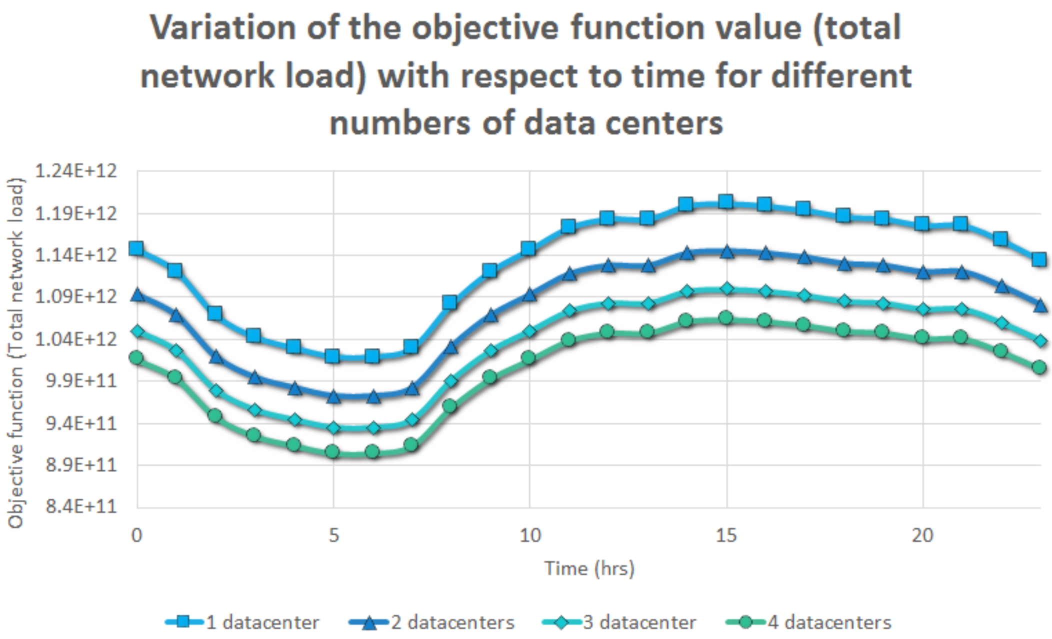

Figure 5.9: Variation of the objective function value (total network load) with respect to time for different numbers of data centers

proportional. When the number of datacenter increases the total network load decreases. We will explain the reason of this behavior after analyzing the graphs in the following section.

Effect on chosen paths

As we explained before, a demand between an SGW and its correspondent PGW may take one of four possible types of paths: 1) A virtualized path where both gateways are virtualized, 2) a decomposed path where both gateways are decomposed, 3) a path having a decomposed SGW and virtualized PGW, 4) a path having a decomposed PGW but a virtualized SGW. Figure 5.10 shows the count of virtualized and decomposed paths for different number of datacenters over 24 hours. What is obvious is that the count of paths, regardless the type, of each number of datacenter remains constant over the 24 hours. Therefore, to have a better understanding of the data, we analyze the relationship between the number of datacenters and the number of different types of paths in figure 5.11.

Figure 5.11 shows the count of of each type of path for different numbers of datacenters. With one datacenter there are 54 virtualized path. We could infer this number from figure 5.3 where there are 6 SGWs connecting to the datacenter via virtualized path, therefore 6 multiplied by 9 QCIs yields 54 virtualized paths. The same logic applies for the number of decomposed paths, which is 108, and

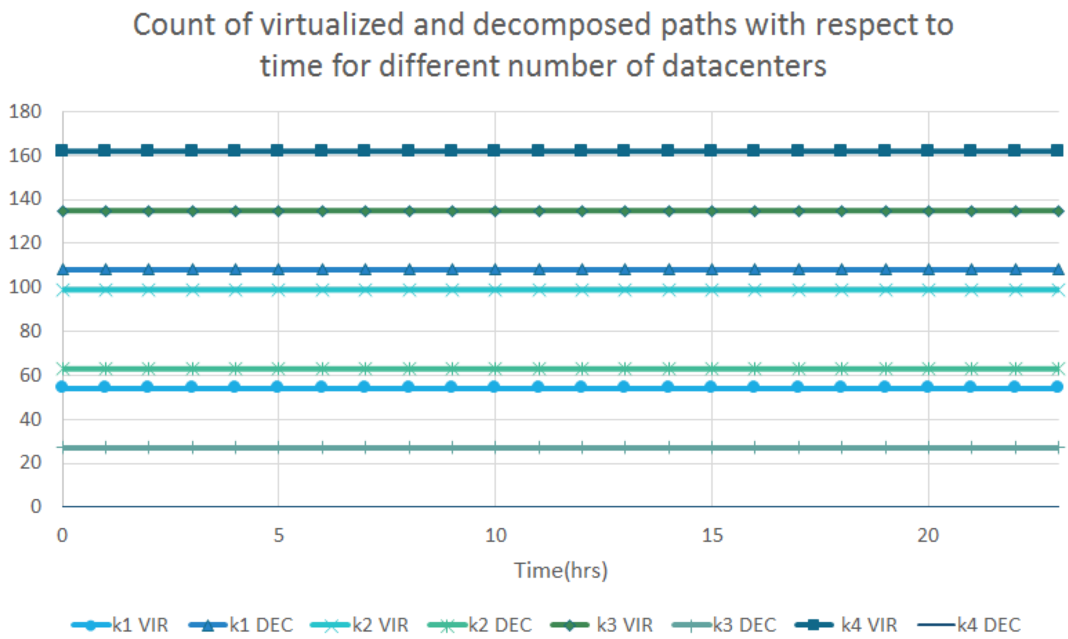

Figure 5.10: Number of virtualized and decomposed paths with respect to time for different number of datacenters

for the other cases with different number of datacenter. Always with the case of one datacenter, the number of virtualized paths is less than the number of decomposed paths this is because fewer gateways are closer to the datacenter so they can use virtualized paths an tolerate the processing delay. When k increases the number of SGWs that are in proximity of datacenters increases, so we will have more virtualized paths and lesser decomposed paths, keeping in mind that when the delay budget permits, the virtualized paths are more favorable because they do not increase the total load. Arriving to $k = 4$, all the PGWs are chosen as datacenters, so all SGWs can communicate with PGWs using virtualized paths, thus for $k = 4$ we have 162 virtualized path and zero decomposed path. So for $k = 4$ the load overhead caused by SDN control plane is null. This is why in figure 5.9 we have seen that with $k = 4$ the load is minimal in comparison with the load of less datacenters number. The conclusion that can be drawn here is that the number of datacenters and the number of virtualized paths are proportional.

Variations with respect to QCI and Time slots

We were expecting that the types of paths may change for the same demand between two time slots of extreme intensity e.g. a decomposed path at 3:00 pm may become virtualized at 5:00 am because the intensity changes and the network become less loaded and the delay decreases. However the results in figure 5.10

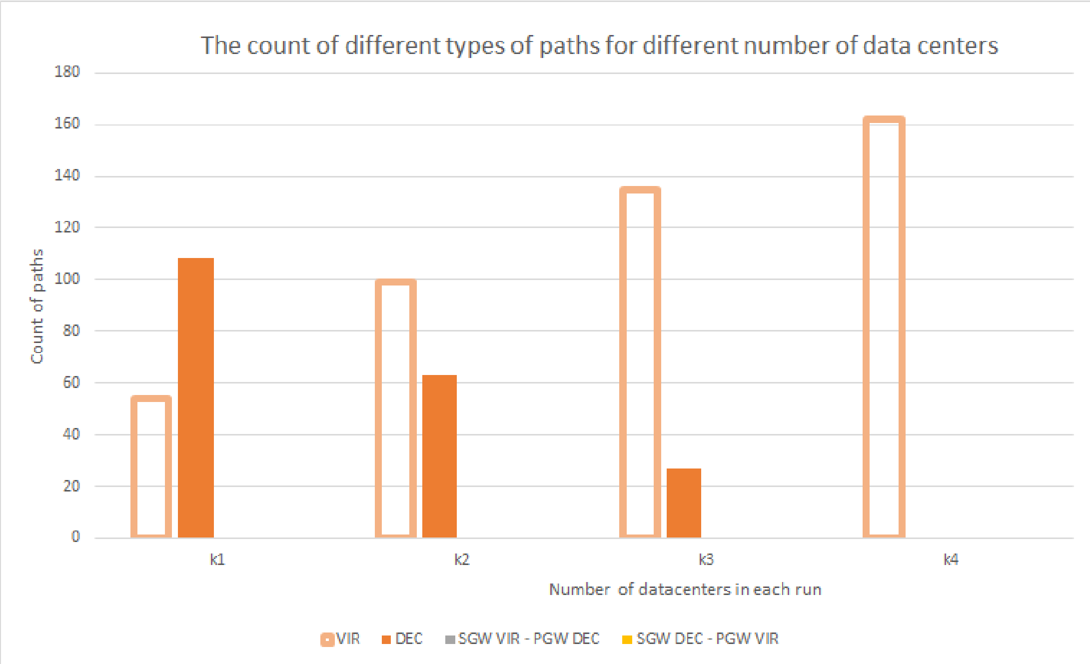

Figure 5.11: The count of different types of paths for different number of data centers

showed that the number of virtualized and decomposed paths is insensitive toward time change. When moving from one time slot to another, the changing parameter is the intensity. This result leads to conclude that the impact of changing the intensity on the objective function is masked by other factors contributing to the total load, primarily the population and the bit rate of the considered QCI.

At the beginning of the analysis, we used figures belonging to a random QCI because the paths did not change type with respect to QCIs, e.g. at a time t, the different QCIs of the same demand take the same type of path. The delay budget constraint is the main changing factor between different QCIs. We expected that for the same demand the QCIs of low delay budget may take a decomposed path while those of high delay budget may take a virtualized path, however both QCI classes took the same type of path whether it is virtualized or decomposed. We believe that the reason behind this behavior is the geographical nature of the considered core network, in other words, the distance between the SGWs and the datacenters is large enough to eliminates the difference between the QCIs regarding their delay tolerance. For example, in figure 5.4 all the QCIs of the demand between SGW 14 and DC of Id 7 take a decomposed path because the propagation delay is already high enough in a way that even if a QCI with a high delay budget, the processing delay of a virtualized path adding to it the propagation delay will surpass the delay budget.

5.5 Runtime

5.5.1 Runtime change with respect to the number of constraints

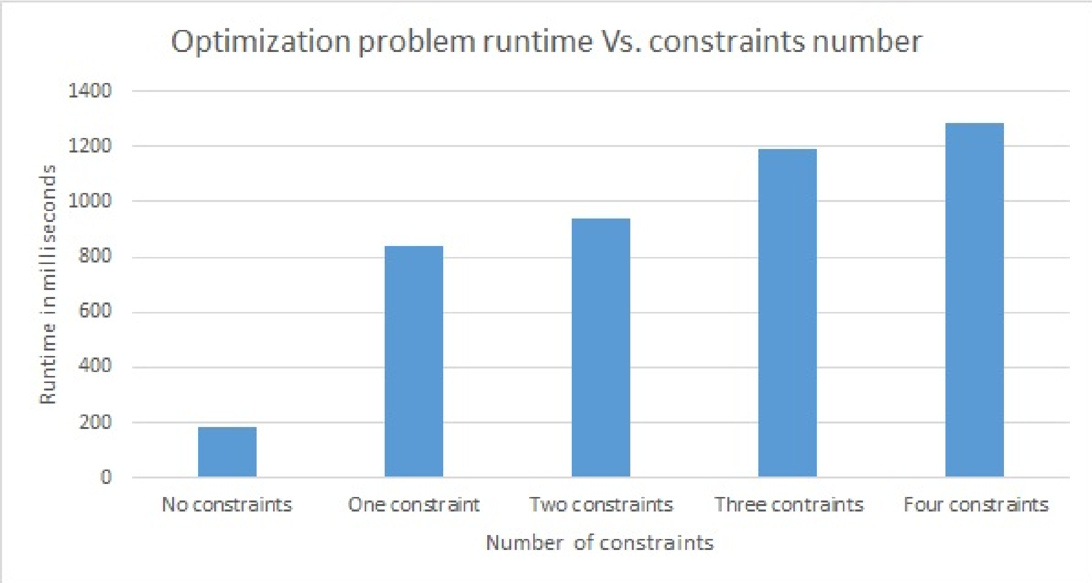

Figure 5.12: The variation of runtime with respect to the considered constraints

Considering, the depicted US core network topology, we ran the optimization problem for a random time slot however for different number of constraints. As shown in figure 5.12, the runtime is minimal (around 180ms) when no constraint is considered. The runtime increased significantly after adding the first constraint. The runtime kept on increasing slighter after adding the second and third constraint till it reached a value around 1280 ms after adding the last constraint.

5.5.2 Runtime change with respect to time slots

We collected the runtime when running the optimization problem on the depicted US core network over the 24 time slots and for different number of active servers. The results are summarized in figure 5.13. We observe that for a given number of active datacenters, there is no clear pattern of the runtime change over the timeslots. However, with one datacenter (the blue line with squares) the runtime has the highest values over all the time slots. While with four active datacenter (red line with circles), we can see that over most time slots, the runtime has the lowest values.

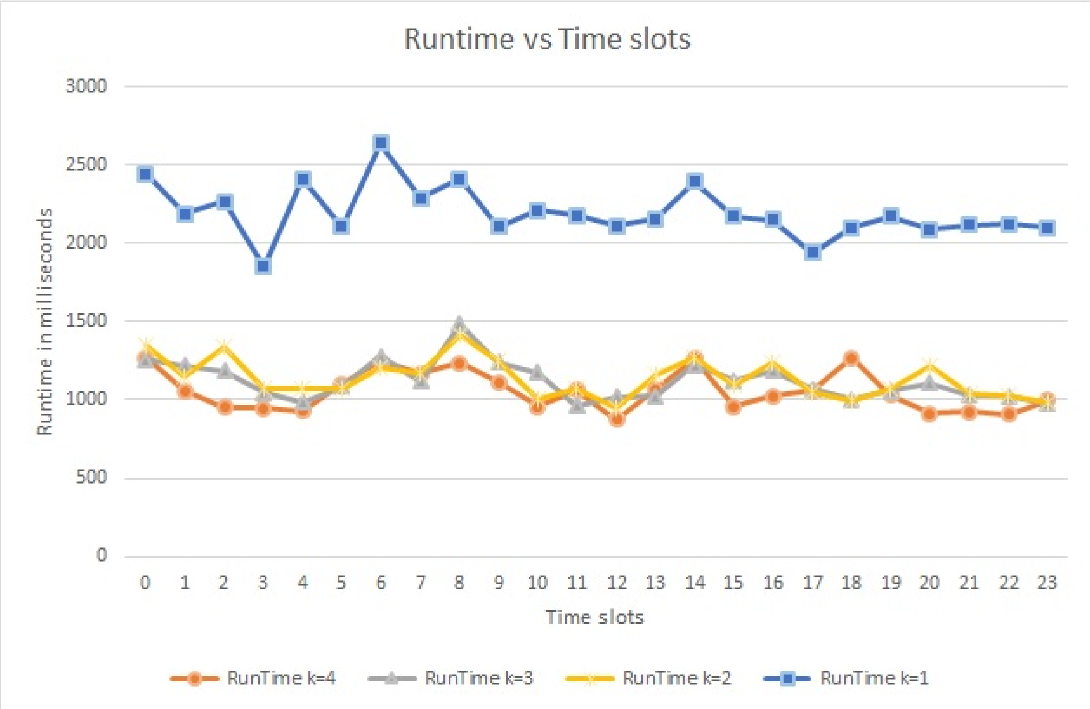

Figure 5.13: The variation of runtime with respect to time slots for different numbers of active datacenters

5.5.3 Runtime change with respect to the number of demands and number of active datacenters

Next, we ran the optimization problem for different number of demands (different network sizes) and for different number of active datacenters. We collected the runtime for each run and the results are presented in figure 5.14. First thing to notice is that for the case of 18 demands, the runtime is the highest for all the different numbers of active datacenters, then we have the case of 3, 2 and lastly on datacenter. This observation leads us to conclude that the larger the network, the higher is the runtime. What we can observe also is that whatever was the size of the network (5, 11, 15 or 18 demands), the runtime having one active datacenter is higher than the case when we have two three or four datacenters which aligns with the analysis done for figure 5.13.

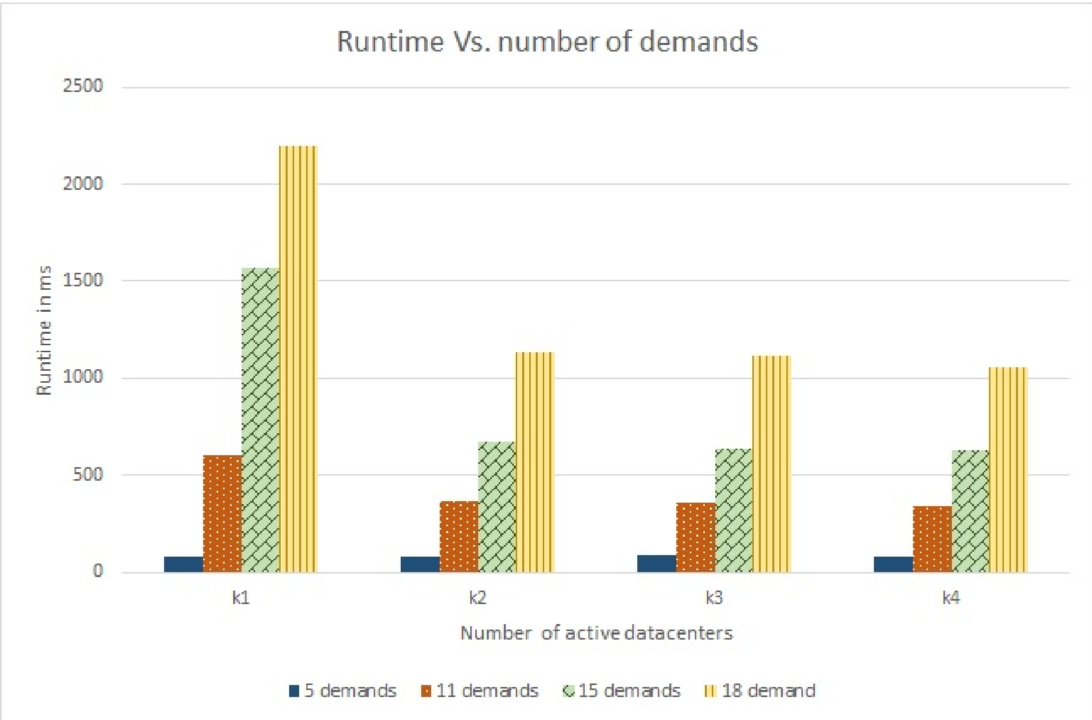

Figure 5.14: The variation of runtime with respect to the number of demands and number of active datacenters

Chapter 6

Conclusion and future work

In this thesis we studied the problem of virtualizing the LTE EPC using Software Defined Networks and Network Functions Virtualization. Most related work that we surveyed paid little attention to applying these technologies on the LTE EPC. Few researchers observed that applying SDN on EPC increases the total network load since SDN comes with an additional control plane, while applying NFV has drawbacks on the packet delay because it requires forwarding packets to a datacenter. In [21] and **??**, the problem was formulated as an optimization problem where the objective is to minimize the total network load subject to a set of constraints, mainly remaining under the packet delay budget. However, [21] considered uniform traffic demands while [22] considered time varying traffic demands but did not adopt SDN decomposition of the gateways and none considered the characteristics of the bearers being established though they differ in characteristics like delay budget and resource requirements.

In our work, we proposed to integrate non-uniform demands, SDN decomposed/NFV virtualized gateways in the same model. Instead of either decomposing or virtualizing a gateway, we proposed adopting a hybrid architecture by applying both technologies on each gateway aiming to find the optimal path of each set of bearers, having the same QCI, between each connected SGW and PGW (demand), while minimizing the total network load without exceeding the correspondent delay budget, at each time slot. In other words, for each QCI of each demand, at each time slot, we could find the optimal path, which determines whether to select the SDN deployment or the NFV deployment, and locate the optimal active datacenters.

For testing and simulation, it was hard to get a real operator core network data, so we depicted the US core network, as given in [21] and [22], using its LTE coverage map and the US population. We performed several runs with different parameters outputting the optimal topology after each run. We observed that the closer SGWs to the datacenter took virtualized paths while the farthest took SDN decomposed paths, which assert the fact that SDN decomposition decreases the network delay while it increases the total network load, and on the other hand, NFV virtualized gateway does not add any extra load to the network because of the absence of any additional control layer, however it penalizes on traffic delay. We also noticed that with more datacenters, the number of decomposed paths decreases while the number of virtualized paths increases. Regarding the runtime, it was in order of milliseconds, and it decreases when shrinking the size of the network.

For future work, further mobile core network components such as the MME could be added, with control-plane delay budget taken into consideration as well. Further constraints to the functions placement problem could be included, which consider the available resources at each datacenter or the transport network links available bandwidth. Also developing power saving models must be investigated. The main limitation we confronted was the lack of data. This is why we had to use what is available in research sites. In future we aim to test the model on smaller scale networks and other real data sets if possible.

Appendix A

Abbreviations

3GPP	Third Generation Partnership Project
1G	First Generation Networks
2G	Second Generation Networks
4G	Optimization Navigator
ACL	Access Control List
AMBR	Aggregate Maximum Bit Rate
API	Application Program Interface
ARPU	Average Revenue Per User
AS	Access Stratum
BSC	Base Station Controller
BTS	Base Transceiver Station
CAPEX	Capital Expenditure
CDMA	Code Division Multiple Access
CTR	Controler
DC-O	Datacenter Orchestrator
DPI	Deep Packet Inspection
eNB	Evolved Node B
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FDMA	Frequency Division Multiple Access
FTP	File Transfer Protocol
GBE	Guarenteed Bit Rate
GGSN	Gateway GPRS Support Node
GTP	GPRS Tunneling Protocol
GTP-C	GPRS Tunneling Protocol Control Part
GTP-U	GPRS Tunneling Protocol User Part
HLR	Home Location Register
HSDPA	High Speed Downlink Packet Access
HSS	Home Location Subscriber
IMS	IP Maltimedia Subsystem

LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
MBR	Maximum Bit Rate
MFC	MobileFlow Controller
MFFE	MobileFlow Forwarding Engine
M-GW	Media Gateways
MME	Mobility Management Entity
MSC	Mobile Switching Center
MTSO	Mobile Telecommunication Office
NAS	Non-Access Stratum
NE	Network Element
NFV	Network Functions Virtualization
NSS	Network Switching Subsystem
OCC	Open Central Controler
OF	OpenFlow
OPEX	Operational Expenditure
PDN	Packet Data Network
PGW	Packet Data Network Gateway
PSTN	Public Switching Telephony Network
QCI	QoS Class Identifier
QoS	Quality of Service
RRC	Radio Resource Control
RSS	Radio Sub System
SDMN	Software Defined Mobile Networking
SDN	Software Defined Networking
SGSN	Serving GPRS Support Node
SGW	Serving Gateway
TCAM	Ternary Content Addressable Memory
TCO	Total Cost Ownership
TDMA	Time Division Multiple Access
TEID	Tunnel Endpoint Identifier
TFT	Traffic Flow Template
UE	User Equipment
Virtual	Machine
VLR	Visitor Location Area
VoIP	Voice over IP

Bibliography

[1] F. Capozzi, G. Piro, L. Griecoc, G. Boggia, and P. Camarda, "Downlink packet scheduling in lte cellular networks: Key design issues and a survey," *IEEE Communication Surveys and Tutorials*, vol. 15, 2013.

[2] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Computer Networks*, vol. 72, pp. 74–98, july 2014.

[3] C. Cox, *An Introduction to LTE: LTE, LTE-advanced, SAE, and 4G Mobile Communications*. Hoboken, NJ: John Wiley & Sons, 2012.

[4] Ericsson, "Ericsson mobility report," June 2014.

[5] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future internet," *Computer Networks*, vol. 75, pp. 453–471, December 2014.

[6] *Software-Defined Networking: The New Norm for Networks, white paper*. ONF, April 2012.

[7] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Commputer Communication Review*, vol. 38(2), pp. 69–74, April 2008.

[8] *OpenFlow Switch Specification: Version 1.2 (Wire Protocol 0x03)*. ONF, December 2011.

[9] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "Nfv: State of the art, challenges, and implementation in next generation mobile networks (vepc)," *IEEE Network*, vol. 28, pp. 18–26, November 2014.

[10] M. Chiosi, D. Clarke, P. Willis, A. Reid, J. Feger, M. Bugenhagen, W. Khan, M. Fargano, C. Cui, H. Deng, J. Benitez, U. Michel, H. Damker, K. Ogaki, T. Matsuzaki, M. Fukui, K. Shimano, D. Delisle, Q. Loudier, C. Kolias, I. Guardini, E. Demaria, R. Minerva, A. Manzalini, D. Lpez, F. J. R.

Salguero, F. Ruhl, and P. Sen, "Network functions virtualisation - introductory white paper," *Proc. SDN OpenFlow World Congr.*, pp. 1–16, 2012.

[11] "OpenNaas." `http://opennaas.org/`.

[12] Tellabs, "Tellabs end of profit study executive summary, tech. rep.," January 2011.

[13] "Cisco visual networking index: Global mobile data traffic forecast update, 2012-17," February 2013.

[14] S. Elby, "Carrier vision of sdn and future applications to achieve a more agile mobile business," *Kynote at the SDN and OpenFLow world congress*, October 2012.

[15] B. Kim and P. Henry, "Directions for future cellular mobile network architecture," *First Monday*, vol. 17, December 2012.

[16] J. Kempf, B. Johansson, S. Pettersson, H. Lning, and T. Nilsson, "Moving the mobile evolved packet core to the cloud," *2012 IEEE* 8^{th} *International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 784–791, October 2012.

[17] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Communications Magazine*, vol. 51, pp. 44–53, July 2013.

[18] X. Jiny, L. E. Li, L. Vanbevery, and J. Rexford, "Softcell: Scalable and flexible cellular core network architecture," *ninth ACM conference on Emerging networking experiments and technologies*, pp. 163–174, December 2013.

[19] 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," *2013 IEEE* 2^{nd} *International Conference on Cloud Networking (CloudNet)*, pp. 205–209, November 2013.

[20] 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," *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, November 2013.

[21] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt, "Applying nfv and sdn to lte mobile core gateways, the functions placement problem," *the 4th workshop on All things cellular: operations, applications, and challenges*, pp. 33–38, August 2014.

[22] A. Basta, A. Blenk, M. Hoffmann, H. J. Morper, K. Hoffmann, and W. Kellerer, "Sdn and nfv dynamic operation of lte epc gateways for time-varying traffic patterns," 6^{th} *International Conference on Mobile Networks and Management*, September 2014.

[23] M. Alasti, B. Neekzad, Clearwire, J. Hui, and R. Vannithamby, "Quality of service in wimax and lte networks," *Communication Magazine*, vol. 48, pp. 104–111, May 2010.

[24] "3gpp ts 23.203 v. 8.8.0, policy and charging control architecture, dec.2009,"

[25] H. Ekstrom, "QoS control in the 3GPP evolved packet system," *Communication Magazine*, vol. 47, pp. 76–83, February 2009.

[26] "Gurobi Optimization - The Best Mathematical Programming Solver." `http://gurobi.com/`. Accessed: 2015-12-05.

[27] "LTE Coverage Map.." `http://platform.cellmaps.com/`. Accessed: 2015-12-05.

[28] "LTE Community Facts - Find popular facts (population, income, etc.) and frequently requested data about your community." `http://factfinder.census.gov/faces/nav/jsf/pages/community_facts.xhtml`. Accessed: 2015-12-05.

[29] "Free Map Tools - Measure distance map." `https://freemaptools.com/measure-distance.htm`. Accessed: 2015-12-05.

[30] C. Rossi, N. Vallina-Rodriguez, V. Erramilli, Y. Grunenberger, L. Gyarmati, N. Laoutaris, R. Stanojevic, K. Papagiannaki, and P. Rodriguez, "3GOL: Powerboosting ADSL using 3G OnLoading," *in Proceedings of the ninth ACM conference on Emerging networking experiments and technologies - CoNEXT '13, (New York, New York, USA)*, pp. 187–198, December 2013.