Optimizing Mobile Backhaul Using Machine Learning

Abdulkadir Mohammedadem

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 30.05.2019

Thesis supervisor:

Prof. Raimo Kantola

Thesis advisor:

Dr. Jose Costa-Requena

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Abdulkadir Mohammedadem
Title of the Thesis: Optimizing Mobile Backhaul Using Machine Learning
Date: Language: English Number of pages:8+56

Department of Communications and Networking

Professorship: ELEC3029 - Communications Engineering

Supervisor: Prof. Raimo Kantola
Instructor: Dr. Jose Costa-Requena

The thesis focuses on the analysis of current limitations of the mobile backhaul solutions technology when applied to 5G technology. The fast growth in connected devices along with the introduction of 5G technology is expected to cause a challenge for efficient and reliable network resource allocation. Moreover, massive deployment of Internet of Things and connected devices to the Internet may cause a serious risk to the network security if they are not handled properly. To solve those challenges, the Mobile Back haul (MB) infrastructure must increase capacity, improve reliability, availability and security.

Software Defined Networks (SDN) and Machine Learning (ML) techniques were used on top of the basic IP routing to measure and estimate the available resources in the network and apply Traffic Engineering (TE) logic to reallocate available resources to newly added slices. The experiment was performed in a virtual environment using Mininet simulator tool and other opensource software and ML algorithms.

In this thesis, a system was developed to measure the existing resources in the mobile backhaul and redistribute dynamically to different network slices either existing or new slices to make sure that each slice requirements are met.

The thesis includes an early prototype of the Mobile Backhaul Orchestrator (MBO) that will be simulated to confirm it can effectively allocate resources to new slices while maintaining existing slices, and that it can contain the traffic within a slice during peaks without affecting traffic in other slices.

Keywords: SDN, TE, MB, ML, Mininet, slice

Preface

I would like to thank Allah, the almighty for finishing my thesis successfully. My first appreciation and thanks go to my professor Raimo Kantola for his continuous support and guidance throughout my thesis work. I would like to express my special thanks to my advisor Dr. Jose Costa-Requena, for your constructive comments and suggestions. I could not have imagined to finish my tesis without your support. You are simply a true mentor.

Last but not least, I would like to send my special thanks to my family. Your encouragement and support have been huge.

Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
BGP	Boarder Gateway Protocol
CS	Circuit Switching
CIoT	Cellular IoT
CP	Control Plane
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
ECMP	Equal Cost Multipath
EPS	Evolved Packet System
EPC	Evolved Packet Core
ETSI	European Telecommunication Standard Institute
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
GTP	GPRS Tunneling Protocol
HLR	Home Location Register
HSS	Home Subscriber Server
HTTP	Hypertext Transport Protocol
IEEE	Institute of Electrical and Electronics Engineers
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
ITU	International Telecommunication Union
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
LTE	Long-Term Evolution
LTE-M	LTE Cat-M1
MBO	Mobile Backhaul Orchestrator
MME	Mobility Management Entity
MPLS	Multiprotocol label Switching
MSC	Mobile Switching Center
NB-IoT	Narrow Band IoT
ODL controller	Open Daylight Controller
OPEX	Operational Expenses
OSPF	Open Shortest Path First
PCRF	Policy Charging Rules Functions
PDN	Packet Data Network
PDN-GW	Packet Data Network Gateway
PELR	Packet Error Loss Rate

PS	Packet Switching
QoS	Quality of Service
RAN	Radio Access Networks
RIP	Routing Information Protocol
SDN	Software Defined Networking
S-GW	Serving Gateway
SLA	Service Level Agreement
TE	Traffic Engineering
TEID	Tunnel End point ID
UDC	User Data Convergence
UE	User Equipment
URL	Unified Resource Locator
USIM	Universal Subscriber Identity Module
3GPP	3rd Generation Partnership Program

1 Introduction

This thesis studies the current limitations of networking technologies when applied to 5G mobile backhaul. 5G architecture needs to address new requirements to enable network slicing, Ultra Reliable Low Latency (URLLC) and Multi-access Edge Computing (MEC). The existing networking technologies used in mobile backhaul are suitable for fixed IP networks where fully distributed routing algorithms provide optimal paths based on link costs and react efficiently upon link breaks. IP networking delivers a best effort network in which each packet is treated in the same way. However, 5G mobile networks aim at new features such as network slicing where the same network provides multiple network overlays each with different traffic requirements.

A network slice in 5G is a set of packet transport links and nodes, set of computing elements and software for the network functions to run a network using the assigned resources. A slice is set up for a particular use case of the 5G network and they can be provisioned in advance to guarantee the QoS requirements for URLLC, Massive Internet of Things (MIoT) or enhanced Mobile Broadband (eMBB) communications.

The mobile backhaul networks have fulfilled the traffic requirements based on over-dimensioning and pre-provisioning that ensure enough capacity for best-effort IP based networks. However, pre-provisioning will be inefficient in 5G networks since the set of assigned resources can be increased and decreased in size based on user needs and policies that change over time. The slices might be created, updated and terminated dynamically based on end-user requirements.

An example of one slice is the one that only carries the human consumer traffic between the device and public Internet. This slice will route the traffic from the device to the point of attachment to the Internet and back to the user. Another example of a slice would carry only machine to machine URLLC traffic that needs low delay, high reliability and very high security.

The mobile operator can over-dimension and pre-provision the network with the required slices for URLLC and the required network resources can be reserved to isolate the URLLC traffic to its own slice. Thus, pre-provisioning allows to tailor the network resources to meet the traffic requirements in the best possible way knowing user needs in advance. However, the network needs to dynamically re-allocate the available resources to include new slices without disrupting existing ones. Thus, in case of un-predictable emergency situations, the network should dynamically re-allocate available resources to support new ad-hoc URLLC or other types of slices. Moreover, in case the machine to machine (M2M) traffic increases drastically due to

their schedule based transmission patterns, the M2M slice has to be isolated from others.

Therefore, the network must re-allocate available resources dynamically to guarantee URLLC requirements for the new slice without disrupting existing slices. Also, network slices would be used to isolate unpredictable peaks of traffic e.g. M2M from URLLC or best effort traffic. Moreover, the slice management has to be done seamlessly to ensure high reliability of existing network slices as they might be in use for smart grids protection or other mission critical industrial applications, which cannot afford any disruption due to network updates.

1.1 Background and Motivation

The exponential growth in connected devices along with the introduction of 5G technology is expected to cause a challenge for efficient and reliable network resource allocation. Moreover, massive deployment of Internet of Things and connected devices to the Internet may cause a serious risk to the network security if they are not handled properly. During the 5G era, network operators will have a chance to dynamically create and deploy different use cases or services such as massive IoT, URLLC, Mobile broadband etc., in the existing network infrastructure. Therefore, service providers should come up with a solution to ensure the security, reliability and allocation of the necessary resources to customers that require URLLC services while optimizing the usage of remaining available resources for new customers and satisfy their demand.

5G brings network slicing which allows operators to share a single mobile network between several use cases with different requirements. The operator can reserve some of the available resources for pre-configured network slices in order to guarantee URLLC type of communications to selected customers. They can be over-dimensioned to provide sufficient resources for URLLC communications and the remaining resources can be allocated to additional network slices without strict reliability requirements.

However, additional URLLC slices with short lifespan which have not been pre-provisioned in the network can be requested at any time e.g. for emergency situations. This means the operator must reallocate residual network resources that have to be optimized for short lived network slices. Such changes when handled by the same routing algorithm that is used for all the network slices in the network infrastructure might lead to instability or misuse of available resources. The operators are claiming that their network will support billions of connected machines. However, the M2M type of communications follow a pattern based on scheduled transmissions (e.g. NB-IOT devices attach periodically for sending data and detach immediately to save

battery). This traffic might create peaks of traffic that will disturb other URLLC or best effort traffic and in some cases, it can bring the network down if not contained.

In order to fulfil specific traffic requirements for Industrial Internet and low latency requirements 3GPP has designed the architecture for 5G networks based on slices to address these requirements.

The current mobile networks have not encountered similar situations previously, where low latency highly reliable communications would be sharing the network infrastructure with massive M2M, emergency communications and best effort end user traffic.

Dynamic routing in the Internet has been a huge success – it allows to run very large networks with minimum management effort. The challenge it is facing is that, dynamic routing as we know it today treats all traffic either with best effort or voice traffic based on Differentiated Services (DiffServ). It cannot support the concept of slicing where some type of traffic needs to be isolated from other traffic.

Some of the changes must take place dynamically as they might be needed for network slices with short lifespan assigned to ad hoc services. Thus, the operator might pre-provision the network with few slices to deliver URLLC services to selected customers. However, adding ad-hoc short-lived slices for URLLC, MIoT or eMBB using existing routing protocol will disrupt the existing slices and operator might not be able to maintain the required resources for pre-provisioned URLLC slices.

Moreover, the network slices are required to constrain unexpected high peaks of traffic e.g. M2M under pre-defined set of resources such that other traffic is not affected and can keep its allocated resources.

1.2 Objective and Scope

5G networks are setting strict requirements in terms of reliability, bandwidth and delay that must be provided for different network slices. This is the most challenging issue network operators are facing right now. It is challenging to meet those different requirements for different traffic allocated to separate slices but using single network infrastructure. Operators need to develop a system which helps monitor and manage the resources allocated for each slice to meet their requirements.

The research in this thesis is mainly focusing on the following objectives:

1. Provide resources for different types of slices and ensure that each slice quality requirements are met.
2. Measure and evaluate the existing network resources in each slice.

3. How to provide resources dynamically to new slices with relatively short lifespan with phases of setup, use and decommissioning.
4. Resource reallocation from an existing slice to another existing slice or newly created slice without disturbing or with minimum disturbance to the operation of other slices.

The thesis will also analyse whether technologies such as SDN and Machine Learning can be used on top of basic IP routing to efficiently manage network slices.
The system evaluates the existing resources and delivers new routing rules or TE logic that takes resources from existing network slices with best effort (BE) traffic and reallocates them to accommodate the new URLLC slice or ensure slices utilize only the given resources.

In this research we will consider Machine Learning (ML) techniques to estimate the available resources in each link based on different network features and calculations made in the network so that it would be used as an input for the routing algorithms to decide the best route. Those features include link bandwidth usage, end to end latency, hop count, packet loss etc. The thesis looks at centralized SDN based management of resources combined with ML to effectively allocate network resources based on new requirements for existing or new network slices.

The thesis includes an early prototype of the Mobile Backhaul Orchestrator (MBO) that will be simulated to confirm it can effectively allocate new URLLC slices while maintaining existing slices, or that can contain the traffic within slice during peaks without affecting traffic in other slices.

1.3 Structure

The thesis is structured in 6 chapters. This Chapter 1 includes the Introduction with the motivation and objectives of the thesis.

Chapter 2 deals with the theoretical background of the research and provides an overview of current transport technologies in mobile networks. This chapter includes the literature review of Software Defined Networking (SDN) in the mobile backhaul network. In this chapter we also discuss different types of machine learning techniques and their integration with SDN.

Chapter 3 explains the network slicing and next generation 5G mobile network features. This chapter presents different types of network services in 5G such as Ultra Reliable Low Latency Communication (URLLC), Enhanced Mobile Broadband(eMBB), massive Internet of Things (MIoT), etc. It also presents how TE works in current mobile backhaul based on DSCP and Quality of Service parameters

assigned to radio link based on QoS Class Identifier which could be used to deploy end to end network slicing.

Chapter 4 presents the design and implementation of the Mobile Backhaul Orchestrator (MBO) that integrates network monitoring system with SDN functionality and ML to deliver optimal network resource reallocation. This chapter includes the validation results of the system when adding new URLLC slices using the proposed MBO. This chapter includes the results showing whether the proposed solution can support the management of network slices and whether allocated resources are maintained to the slices without the slices being affected by external factors such as congestion in the network.

Chapter 5 provides the discussions and limitation of the research. Finally, Chapter 6 provides the conclusion and pros and cons of the research and suggests how the research can be extended for feature works.

2 Background

This chapter provides an overview of the mobile networking technologies. The literature review of machine learning technologies and the usage of Software defined networking (SDN) in the mobile backhaul network are also introduced in this chapter.

Tremendous increase in connected devices is the reason for the growth in the data traffic in recent years and this increase does not seem to slow down any time soon [4], [5]. The introduction of 5G technology along with the growth of the Internet of Things (IoT) is another reason for the increase in data traffic. The connected devices are expected to grow to 50 billion by the end of this decade [6]. The legacy mobile networks, 2G, 2.5G, 3G which are still in use, are not designed to support such huge data traffic. 2G and 2.5G networks use time division multiple access (TDMA) and different protocol standards unlike the new generation networks 4G and 5G which use all IP packet-based technology [6] [7]. Therefore, those networks cannot provide the necessary features which satisfy the user experience and quality of service in an efficient and cost effective-way [4].

Communication and accessing remote data through mobile networks are becoming mandatory features. However, transferring data through an air interface is only half of the story. The mobile backhaul system is the other half which moves the data from the cell site to the external packet data networks (PDN) such as public Internet and other points along the way [9].

2.1 Mobile Backhaul

Mobile network operators are having a big challenge to balance the fast-growing trends of smart phone users which require high bandwidth and provide the necessary resource to meet those demands. Therefore, to catch up with the user demand the Mobile Backhaul (MB) infrastructure must increase capacity, improve reliability, availability and security and at the same time the operators need to keep the operational (OPEX) costs as low as possible to stay in the market and profitable.
The mobile networks are structured to scale and support different types of deployments based on the restrictions in different areas.

Thus, mobile networks are separated into Radio Access Network (RAN), Mobile Backhaul (MB) and Core network (CN) each of them with different transport technologies.

2.1.1 Mobile access network

The mobile Radio Access Network (RAN) which is connected to the core network via the back-haul network, contains the radio technology that ensures the user traffic modulation. It provides radio access and coordinates management of radio resources across different cell sites. The access network of different generations of mobile networks contain different cells. Such cells contain the necessary hardware and software technologies which are used for communication with user equipment. The technologies are different for different mobile generations as shown in Figure 1. For example, the 2G access network, which is called BSS (Base Station Sub System) is different than the 3G access network which is the UTRAN (UMTS Terrestrial Radio Access Network). [10] The LTE (Long Term Evolution of 3G) or eUTRAN (evolved UTRAN), the access network for 4G, contains eNodeBs and it has flat architecture as it does not have centralized controller unlike its predecessors.

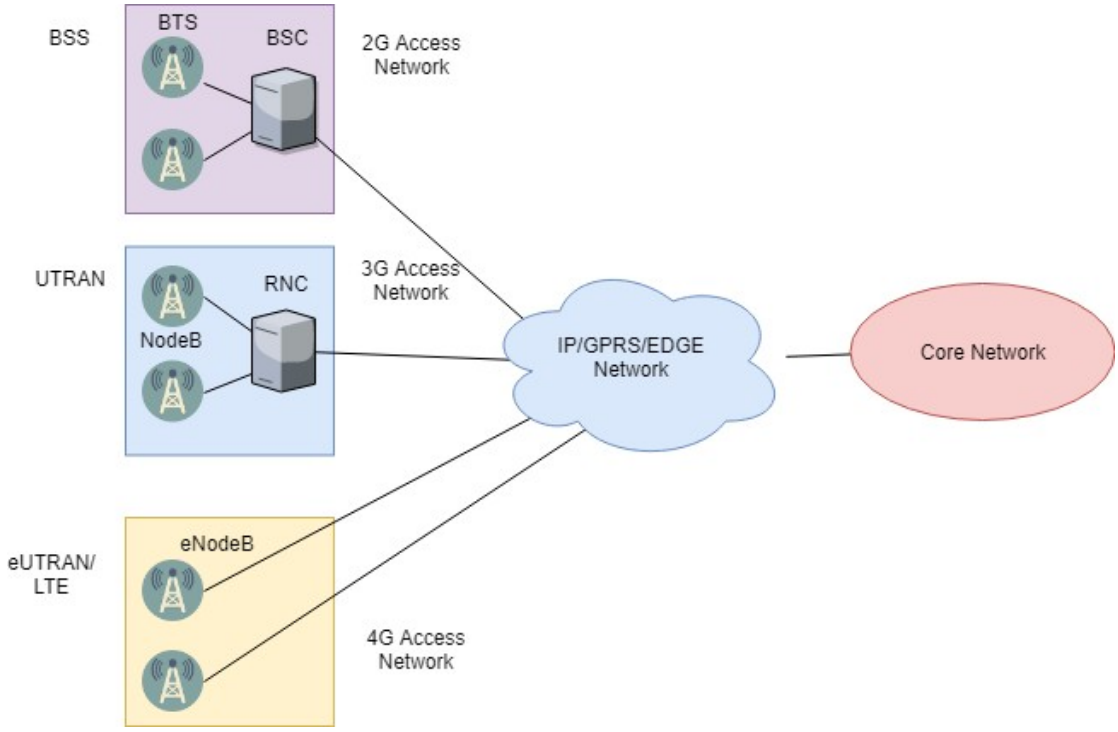


Figure 1 Radio Access Network (RAN) architecture of different generations

The 4G architecture significantly differs from its predecessors. Some of the changes are removal of circuit switching capabilities and carrying all traffic in packets including voice traffic. Therefore, backward compatibility is maintained by segmenting voice data into packets and routing them using VOIP (Voice Over IP) technology. The other difference is the integration of wireless LAN technology in to 4G. [11]

2.1.2 Mobile backhaul networks

The mobile backhaul is the network infrastructure that provides the connectivity between the cell site air interface (RAN) to the wireline core network which subsequently is connected to the data centre or the Internet [10]. The MB is part of the end to end mobile network system which serves as a link between the Radio Access Network (RAN) and the Internet. It transfers the mobile data from the radio base station to the PDN and to other traditional mobile networks [44]. This connectivity can be of different types such as optical fibre, microwave radio, copper DSL, satellite etc.

The mobile communication technology has been evolving through time starting from the first generation (analogue systems) to the currently expected fifth generation (5G) and the features of these different technologies are listed in Table 1. Geographical location of the cells, bandwidth (BW) requirements, policy rules and regulations are some of the factors which affect the connectivity of the Mobile Backhaul system [9] [11].

Table 1 shows different technologies used for various mobile generations. It also shows, the capacity and the support for different backhaul technologies, which is changing every time and increasing visibly.

Table 1. Mobile network technologies

Generation	Technologies	Device data rates	BTS support	Backhaul support
2/2.5G	GSM/GPRS/ EDGE	64kbps/64-144kbps	Channelized TDM	PDH/SDH
W-CDMA(3G UMTS)	UMTS	384Kbps/384Kbps	ATM	ATM
HSPA(3.5G)	HSPA	14.4Mbps/384Kbps 14.4Mbps/5.72Mbps		
HSPA+	HSPA+	28Mbps/11Mbps 42Mbps/11Mbps	Ethernet/IP	Ethernet/IP
LTE(4G)	LTE	138Mbps/37Mbps	Ethernet/IP	Ethernet/IP

After the evolution of the second generation of mobile networks the MB has undergone a lot of changes both in terms of technological advancement and capacity.

The main reasons for those changes are, mobile devices require different quality of service and demand higher bitrates. The issue is that, traditional networks cannot afford to meet those demands [13]. Based on the current traffic growth, it is expected that the bandwidth usage will exceed 1Gbps from each macro base station towards the MB. The traffic will grow based on current trends of video streaming, longer content of viewing times, increasing interest in high resolution video etc. [3] [13] [14] Moreover, virtual reality and other bandwidth intensive and jitter sensitive services are posing a new challenge to MB. On the other hand, these services are also creating new business opportunities and encourage innovations.

The driving force for the introduction of all IP based architecture in mobile networks was increasing BW demand and user's requirement for quality of service. This architecture reduces the operational cost of mobile operators [4]. The new generation of the mobile networks such as Long-Term Evolution (LTE) and High-Speed Packet Access (HSPA) are examples of all IP based mobile networks unlike its predecessor-ATM based 3G and PDH and SDH based 2G, which are still existing but cannot fit the demand of the increasing network traffic [16].

The Long-Term Evolution (LTE) network consists of two parts: the backhaul network which connects the access network to the core network containing different logical components or so-called network nodes. Those network nodes are:
Mobility Management Entity (MME) -is the main component of the mobile core network which provides session management and control plane functions. It is responsible for handovers between eNodeBs as well as selecting the serving gateway, roaming, paging and user status.
Home Subscribers Server (HSS) -is in charge of storing and updating user or subscriber information in the database. The database contains IP address of the user, the subscriber's profile, user status (active/idle, attach/detach) and other important QoS information of the subscriber.
Serving Gateway (SGW) -is an interface between the radio part of the network and core network. It is used to forward user data between the PDN gateway and the access network.
Packet Data Network Gateway (PGW) – is a gateway between the EPC and the external IP networks. It provides IP address, routing and other functionalities and enforces data flow policies.
Policy Charging Rules Function Server -is another component of the Core network which ensures the Quality of Service (QoS) charging bills and manages data flow policies.

Figure 2 shows the different components and architecture of the Evolved packet core which is connected to the access network and the internet.

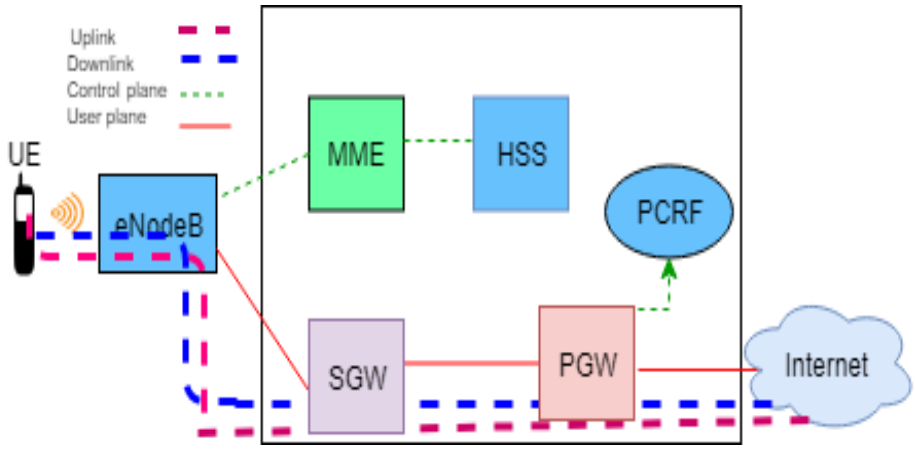


Figure 2 EPC network architecture

If the LTE is to meet user demands, it has to provide high quality of real time and non-real-time mobile broadband access in a reliable and efficient way. To achieve this, the backhaul network is very important as the whole traffic load that comes from the access network to the Internet and from the Internet to the devices passes through this network [7].

Today users are accessing rich multi-media contents and they want to access the content at anytime, anywhere on any device without any delay. The current 4G network is not enough to support the huge data traffic that results. Therefore, to meet the challenge, network operators have two options:
The first one is increasing the capacity and coverage of the existing LTE networks by deploying a number of 4G radio resources and scaling up the backhaul accordingly. This approach is both costly and not feasible in the long run as customers do not show any interest to increase their payment for the service [14].

The second option is to add some intelligence and make the mobile backhaul agile and self-adapted when scaling the network to support future services.
Software defined networks (SDN) is the proposed technology to dynamically react to the changing data traffic demand to provide enough capacity when needed. This technique seems the better solution in terms of cost and resource optimization [14].

2.1 Packet routing in mobile backhaul

Legacy mobile backhaul networks before the introduction of 3G were initially designed to carry only voice traffic using TDM. After the introduction of 3G technology,

which initially used ATM and turned to Ethernet later, data traffic starts to grow and the shortcomings of TDM in the backhaul start to appear due to cost and bandwidth limitations. Ethernet starts to emerge to replace TDM as a backhaul technology to avoid the drawbacks followed by the All IP which is used in most recent wireless technologies such as WiMAX, HSPA + and LTE. However, current MB network architecture cannot meet the demands of customers given the continuous growth of the IoT devices coupling with the introduction of 5G technology.

Therefore, research is going on to develop a system which is flexible enough to accommodate the exponential growth of devices and consequently data traffic can be allocated to different slices of the network as shown in Figure 3.

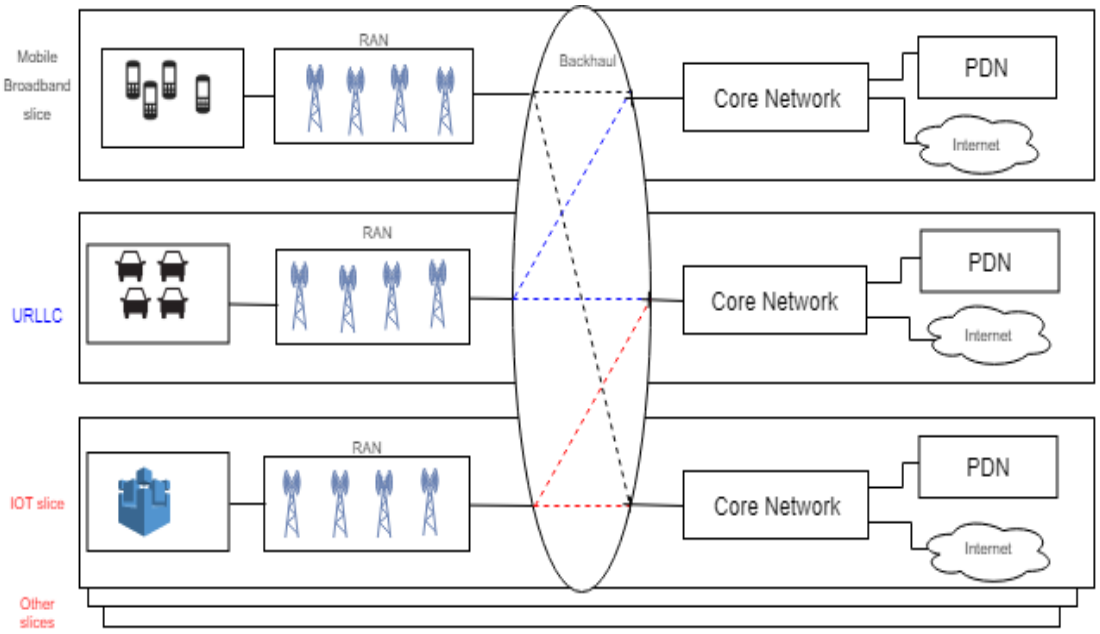


Figure 3 Mobile Network slice architecture

2.2.1 IP/MPLS routing

Routing is a process of finding a feasible path to send a packet from source to its destination using different rules or protocols, steps(algorithms) and different tables or maps.

Basically, there are many types of routing systems depending on different factors such as: 1) Source routing where the source (router) has full information about the route on how to send the packet to its destination and 2) hop-to-hop routing, where the node only cares about the next node to send the packet, i.e. it sends the packet to its neighbour(hop) and the next hop forwards the packet to its next hop and network keeps on doing this until the packet reaches its destination.[17]

As stated above, there are different rules (protocols), routing tables(maps) and steps (algorithms) which govern the routing process so that each network device will follow those rules to forward packets to their destination. Routing tables are information stored in the router- an intermediate device responsible for routing packets. An entry in the routing table is identified by a destination address prefix of variable length. Routers have the Forwarding Engine (FE) and the Routing Engine (RE). FE has the forwarding table that is used to forward packets. RE calculates and stores the routing table, that is used to create the forwarding table. RE runs routing protocols (OSPF, BGP etc) while the FE forwards the packet after longest match prefix search (LMPS) in the FT to find the entry that will be used. LMPS input is the destination address extracted from the packet at hand. Therefore, when the router gets a new packet, it looks at its routing table and forwards it to its destination address based on the routing information stored on the table. Routing tables could be static, where they are updated manually or dynamic tables which are updated automatically. [18]

Routing protocols: are configured on routers for the purpose of sharing routing information. It is used to update the routing tables and generate them. A protocol is a kind of standard or an agreement format which network devices agree to send packets among each other based on this standard. Protocols differ from one another depending on the functionality, security, reliability. Routing protocols are divided in to two: Intradomain and interdomain routing protocols. Intradomain routing protocols are protocols which work in an autonomous system i.e. within a domain network. Examples of the intradomain routing protocols are Distance vector protocols such as Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP) and Link state protocols such as Intermediate System to Intermediate System and Open Shortest Path First (OSPF). Interdomain protocols are protocols which connect devices on different autonomous systems or different networks. Path vector routing protocols are examples of inter domain routing protocols. [18] [19]

In a distance vector routing protocol, the router announces its topology changes every certain interval of time or when there is a change in the topology in some cases as in the case of EIGRP. A route is advertised as a vector of distance and direction to their immediate neighbours. Distance means hop count or metric distance and direction referred to as next hop address and exit interface of the next router. [19]

Routing tables of routers are updated and created using some algorithm. One of the known algorithms is Bellman-Ford algorithm. Distance vector routing protocol uses Bellman-Ford routing algorithm to calculate routing paths. In this algorithm routers can advertise their information at the same time and could create routing loops especially in RIPv1. However, this can be avoided using different techniques such as split horizon with Poisonous Reverse technique, addition of hold time which prevents routing loops, etc. The algorithm is also iterative, that means routers advertise until all have the same routing information. Once the router updates its routing table, the routers find the shortest path between two nodes using Bellman-ford equation. [19]

$$Dx(y)=minv\{c(x,v)+dv(y)\} \quad \text{Eq(1)}$$

Where Dx is the least cost path from node x to node y. c (x, v) is the cost and dv(y) is the distance.

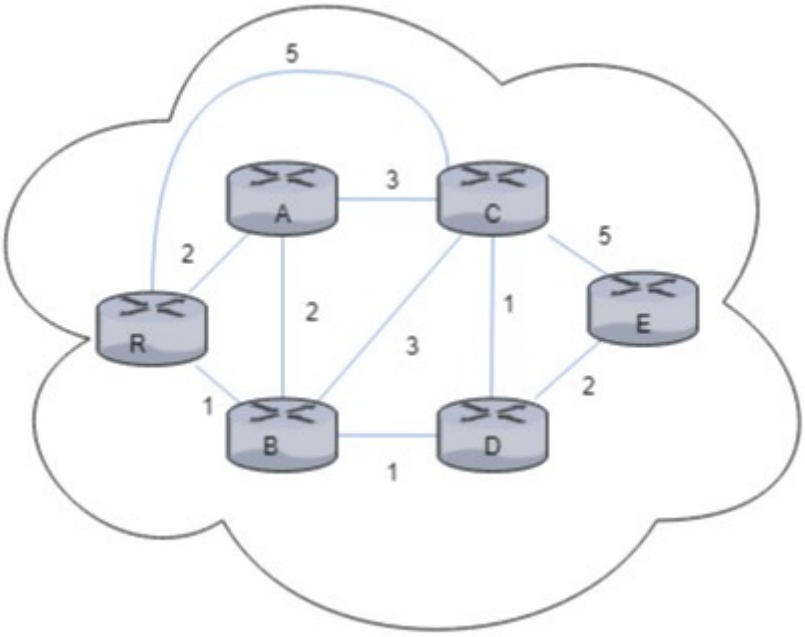


Figure 4 Dijkstra algorithm

Link state routing protocol: an intradomain routing protocol used in a single autonomous system. Unlike distance vector routing protocol where routers advertise every certain interval of time, link state protocols like OSPF etc have both triggered updates and periodic updates. Periodic updates are every 30 minutes by default. Due to Hello protocol, periodic updates can be much less frequent than distance vector (RIP) that does not have Hellos. Link state database contain node id, list of links, sequence number and age.

Figure 4 shows R is for the root node and all other nodes can be reached using Dijkstra algorithm. The numbers in the graph show the cost of the edges. Initially or in case of routing information changes, each node generates a link state advertisement and sends all its information to its closest neighbours and nodes which receive link-state information store a copy on their link-state database and propagate updates to other nodes. Once every node has updated its link-state database, routers start to calculate the shortest path to the destinations using Dijkstra algorithm.

The MPLS label switching has been in use for quite some time. Unlike IP routing technology which uses destination IP address to forward packets to its destination, MPLS uses tags to forward IP packets. Label switching paths are built using Label

Distribution Protocol (LDP) based on IP routing table or using RSVP (Resource Reservation Protocol) protocol. Tagged packets are forwarded once label switching paths are established. The first (ingress) router pushes the label on packets and forwards them. Subsequent label switching routers look at the attached tags, not the IP header to forward packets towards the egress router. The egress router then removes the tags and forwards the packets to their destination. [20] In IP routing packets are forwarded by looking at the destination IP address and matching the best path in the routing table. However, IP packets lookup table can be complex or not complex but the LMPS has been optimized to find the right entry with one or two memory references. MPLS label tabels can be read in one memory reference (max million rows, so the label can directly reference the entry or be an index to the entry. Since 1 memory read takes something like 10ns (or in static memory even less?), the difference between the two methods is not significant in terms of performance

2.2.2 MPLS Traffic Engineering

Traffic engineering is the ability to monitor the traffic through the network and apply not only shortest path but instead use the resources that happen to be available. The need for IP routing is to get the traffic across the network as quickly as possible. Every IP routing protocol has a cost associated with the links in the network. The accumulation of different metrics such as hop count, delay, the cost of every link of a path is used to calculate the best path to route the traffic through. [20]

The forwarding paradigm of IP is based on the shortest path forwarding. IP packets are forwarded to their destination based on destination IP addresses. However, available instantaneous bandwidth capacity of a link is not taken in to consideration on the IP forwarding paradigm.

Therefore, the router keeps forwarding packets even though the link starts dropping packets due to congestion or low available bandwidth on the link, as a result some links are over utilized where as others are underutilized. It is possible to monitor the links and add bandwidth in case of heavy traffic load. However, adding bandwidth to the links cannot be done instantly, it needs planning to upgrade the link capacity. Moreover, the traffic patterns are changing over time from site to site and they are not permanent, TE can bring a solution to avoid congestion on the links which are loaded. [19]

Traffic Engineering is important for analysing measuring and predicting network traffic and suggests optimized traffic paths to improve resource utilization and QoS. In traditional network architecture TE was based on two techniques, IP-based TE and MPLS-based TE. [21] [22]

IP based-TE uses shortest path first for load balancing the traffic to prevent congestion. However, it uses link weight to control the routing paths of the network and traffic cannot split arbitrarily. Therefore, this approach limits full utilization of resources. Another drawback of this approach is when for some reason a link fails, the algorithm needs time to distribute new weights and recalculate optimal paths and distribute it in full convergence. This convergence delay leads to congestion and packet loss. Moreover, the dynamic creation of network slices with own requirements will lead to continual delays in reaching optimal path. To avoid those two challenges, MPLS-based TE includes labelling which allows packets to be forwarded using tags or labels instead of IP headers. There are two variants of MPLS: (1) IP/MPLS where setting up the label switched paths depends on the IP routing protocols and (2) MPLS-TP (for "transport profile) where label paths are set up with network management. MPLS-TP is "carrier grade" while IP/MPLS is not. MPLS-TP does not define how the management is implemented, it just defines what are the resources available in the network nodes and on the "wire".
However, this approach is very complex and creates network overheads and cannot meet the above demands [20]. It is difficult to control and manage traffic which needs flexibility and with efficient utilization of resources using traditional TE-approach. Therefore, there is a need to introduce a new approach to solve those challenges, thus SDN is proposed as the technical solution [23].

2.3 SDN usage in mobile backhaul

SDN is a new technology introduced to solve the challenges faced by traditional networks by separating the control and data plane of the network device to ensure QoS of the network traffic in an efficient way. SDN provides network programmability, flexibility and agility. [10] SDN has the following three features:
Control: the SDN controller stores the whole networking information such as network topology and network status.
Programmability: the main advantage of SDN is to give a chance to network administrators to program the network in such a way that it utilizes the available resources efficiently.
Openness: forwarding devices have interfaces which can communicate with the controller which is not vendor specific and through those interfaces the controller gets status updates and makes routing decisions.

Although SDN is suitable for TE, there are still issues of compatibility with the existing technology and its ability to co-exist for longer time with the existing traditional networks [24]. SDN network architecture is divided into three parts as shown in Figure 5 below.

The **Controller** is considered as the brain of the network. It manages the network flow and resource allocation. It is the core of the network and runs based on Open Flow protocol [25] to communicate with the hardware devices and uses an interface called north bound interface to communicate with the application layer of the network.
The **application layer** contains different software programmes designed and executed to perform certain tasks in the network environment using the controller to interact with the hardware devices.
Infrastructure layer is also called the physical layer which consists of the physical devices in the network. These devices receive instructions from the controller via the Open Flow protocol. The instructions are triggered by the programmes in the application layer.

The OpenFlow controller has the global view of the network topology and monitors the network flow using open flow messages. Technology companies such as Google use OpenFlow to interconnect their data centres and balance network capacity utilization among them based on application demands.
OpenFlow can be used for TE to improve utilization of resources and reduce packet loss and delay. An example of SDN based approach for TE is Adaptive Routing Video Streaming (ARVS) with QoS. [24] [25]

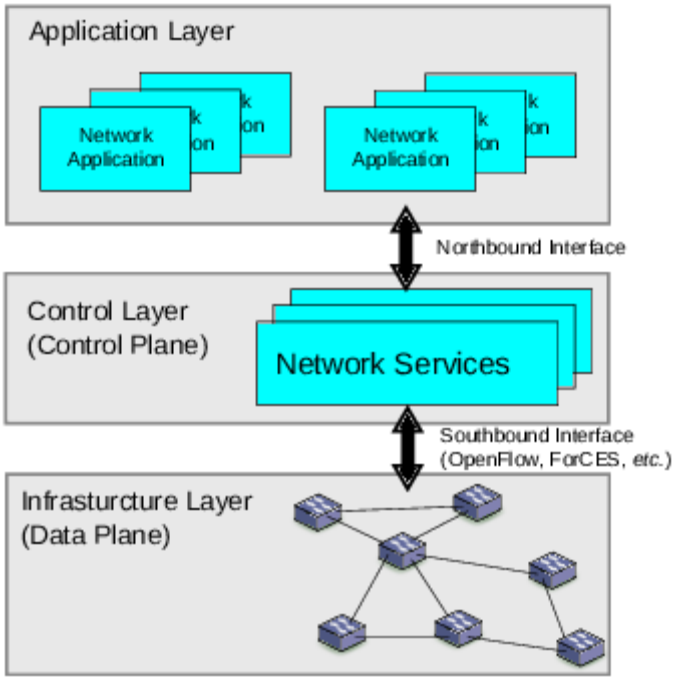


Figure 5 SDN Network architecture [33].

In the existing routing technologies, if congestion happens on the links, it delays the delivery of the packets until the link costs are changed while they are unchanged rout-

ing stays the same and the problem persists. In SDN, the controller can dynamically change the state of the network to adapt to the changes in the topology. Hence, it tries to reduce congestion and increase QoS [27]. However, every time the path of an ongoing flow changes, the flow experiences disturbance in terms of delay and packet loss. Therefore, disturbances caused by such changes need to be minimized as much as possible.

TE techniques introduced in SDN:
Google's B4 is a technique designed to solve the reliability, performance and failure problems in wide area networks. It assigns resources dynamically to the competing applications and services. In this case TE is deployed on top of the routing protocols. This means that if TE logic faces problems when calculating the optimal routes, the normal routing algorithms like Shortest Path First (SPF) will take over and forward packets to their destination.
Deep packet inspection technique (DPI), inspects payload packets and searches for known patterns, keywords or regular expressions. The aim of this technique is to monitor, control and secure the network infrastructure and manage network resources. SDN can play an important role as it provides an overall view of the network topology and can enforce better traffic policy over the network. DPI technique is used to provide more comprehensive characteristics of the packet flow and share information with decision makers. [27] However, DPI technique has some considerations to be taken into account to implement using SDN. First, SDN uses Open Flow which is a stateless protocol whereas DPI is a state full analysis tool and the other reason is while Open Flow offers a chance to reconfigure network elements dynamically which helps to monitor and control network traffic flows in a real time whereas DPI provides solid traffic classification and control. [10][27]

Machine learning (ML): can be used to develop a control logic on the network behavior It uses several flow level features such as end-to-end latency, packet count, link capacity, network delay, hop count, flow count etc., to classify the traffic. [25]

ATLAS framework classifies traffic based on applications. Users need to have software agents installed on their devices. The controller gets statistical information such as active network sockets and net stat logs, are collected using software agents. The controller runs machine learning trainer tool called C5.0 based on application types. The controller collects flow features such as packet size of the flow and these are used to train the ML tool. [27]
MSDN-TE (Multipath SDN), is a multi-path traffic forwarding engineering module that is used to forward traffic in such a way that it avoids congestion on any link from where it collects network information. This TE mechanism gathers information on the state of the network and considers the actual path's load to forward the flows on multiple paths. The MSDN-TE is a module which extends Open Daylight (ODL) controller. ODL is an open source platform used to implement SDN and Network function virtualization. It consists of three components:

A monitoring function- used for gathering information about network states and flows in the network; for example, status of flows, link utilization, network topology, usage of resources such as bandwidth, end-to-end latency etc. The path matrices are refreshed every 10–15 s;
TE algorithm, which calculates the number of paths, which have the lowest traffic load, between the source and destination node.
Actuating function, which supports TE algorithm module. It takes certain actions and dynamically allots flows to the selected paths.[27]

2.4 SDN based network measurement

Network measurement parameters are a set of values which represent the current network status. The SDN network measurement includes the following three parameters:
Network topology parameter- includes number of network nodes, link bandwidth, port status (i.e. up or down) etc. SDN discovers the network topology using Link Layer Discovery Protocol (LLDP). SDN controller sends an LLDP packet to the switch as packet-out message. A switch which receives the LLDP packet from the controller sends it to all neighbouring switches. The switches that receive the LLDP packet from other switches send packet-in message to the controller to handle the packet since the switch does not know where to route the packet. When the controller gets packet-in-messages from the switch it analyses and identifies which switch this switch is connected to and construct the global view of the network topology. The controller can also be configured to be a silent OSPF or IS-IS listener [25] to discover the network topology.

Network traffic parameter- refers to the traffic volume that pass-through network equipment or a network port. This parameter collects the total number of packets and the speed (bytes per second) in each port. Network traffic parameters are considered as the basis for detection of the status of the current network and for predicting user behavior in the network. [25]. Basically, there are two types of network traffic in SDN network, the control and data traffic. The control traffic contains data flows that are transmitted between the SDN controller and the network equipment. Data traffic consists of data flows that originate and terminate at hosts and are transmitted between network nodes
. To determine flow characteristics, Statistical information must be collected from each port of the switch. The information collected includes end-to-end traffic matrix of the entire network, the number of packets, size of packets etc. This information in the traffic matrix represents the volume of network flows between any two network nodes.

Network performance parameter- is used to evaluate the state of the network, to check whether the network flow is in a healthy state or not. Performance parameters such as network throughput, bandwidth utilization, latency, packet loss, and jitter are used to measure and monitor the network status.

2.5 Traffic engineering using machine learning

The main advantage of SDN is the programmability and flexibility that allows the network administrators to program and manage their network resources efficiently. SDN plays an important role in traffic engineering and it can be used for dynamic load balancing. Dynamic load balancing (DLB) reroutes traffic using statistics collected from each device in the network. DLB is a mechanism that takes statistical parameters from each network device and evaluates the network traffic to modify the flow accordingly. [6] [29] [30].

Optimizing the usage of network resources using DLB is far better than static round robin routing algorithm. Round Robin Algorithm (RRA) is the default load balancing strategy which treats all available paths equally. The drawback in large networks is the performance cost and overhead for collecting statistics from each device in the network and computing the best route. Moreover, DLB needs high computing power to calculate the best path [31]. Therefore, there is a need to find some other mechanism to measure and optimize the usage of resources given the complexity of the network and traffic growth. Thus, studying the trends of traffic to identify which time of the day traffic load is quite high in some areas and low in other areas is important to design optimal TE logic. For example, in areas where there is a high number of office buildings, the traffic load is high during day time and week days, but it is quite low on weekends. This kind of trends are similar every working day around those buildings or shopping malls.
In this case, machine learning (ML) can be used to develop a control logic on the network behavior and train the system so that traffic can be rerouted with no need to calculate and compute new optimal paths [32]. Machine learning is basically divided in three parts: Supervised learning, unsupervised learning and reinforced learning each of them described in following sections.

2.5.1 Supervised learning

Supervised learning consists of machine learning where there is training data that can predict the typical outcome in a similar situation. Rather than writing algorithms to do the load balancing of the network, the supervised learning takes several samples of labeled data and trains the devices. Therefore, if similar data is encountered the device can recognize it and take actions based on the training [33]. In supervised learning the

device is trained for some known output. The idea behind supervised learning is that for some inputs we want to have certain value as an output. The machine learning algorithms run based on the inputs until getting output values close enough to the target value we already set. Supervised learning can be further divided in to two categories: classification and regression problems.

Classification problems- is the category that classifies the output as yes or no or positive or negative. For example, classifying flows as elephant flow or non-elephant flow, congested or non-congested etc.

Regression problems- is the category where the trend of a certain variable/output is continuous.

$$y=f(x) + \varepsilon \qquad \text{Eq (2)}$$

In equation 2:

x is input, y is the output and ε stands for the error. Using supervised learning the system tries to learn 'f' through training by example. Someone observes both the input and output of the system and collects a set of observations (xi, yi), i= 1……N where N is a positive integer number. The input values of xi are put into the learning algorithm to produce f(x). The artificial system tries to modify the input (xi) and output f(x) relationship based on the difference between the original (yi) and generated output f(x) i.e. evaluate yi-f(x) until the result is close enough to the targeted output. So that the system can fit for real input values [33].

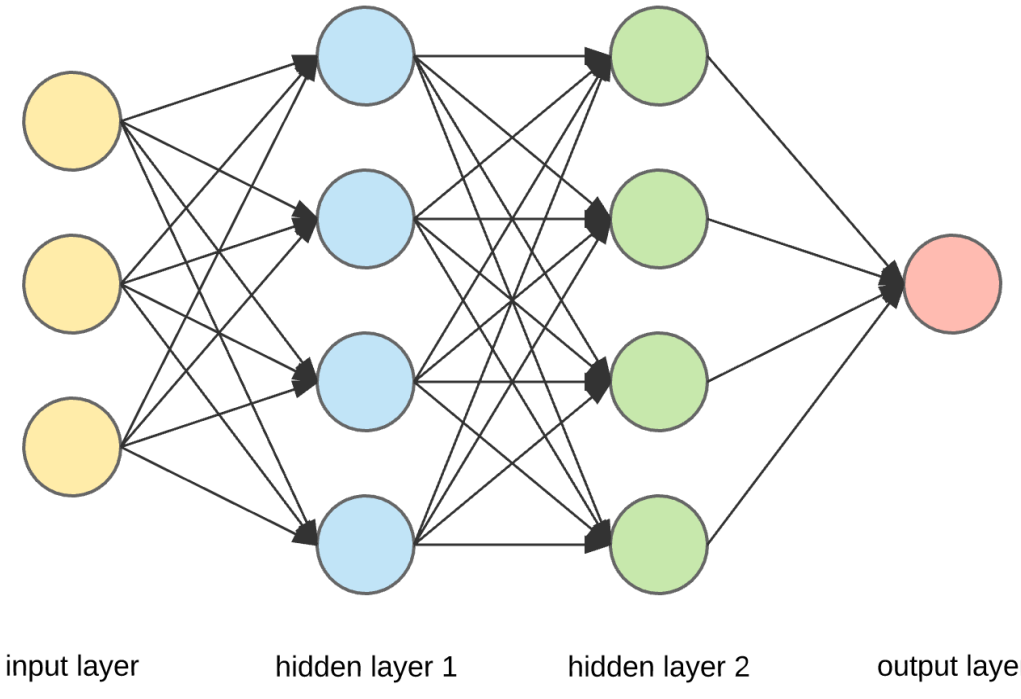


Figure 6 Structure of artificial neural network. [34]

An artificial neural network is one example of supervised learning algorithm which resembles more like the human brain setup. It has non-linear and self-learning behaviors. Unlike other machine learning algorithms such as logistic regression and proba-

bility statistic, artificial neural network does not have limit on input vector [6]. Basically, Supervised learning has three sections: the input layer, the hidden layer and the output layer as shown in Figure 6 below. The hidden layer can be one or more layers depending on the suitability of the selected model. Every node in one layer is connected to every node on the other layer. [34]

2.5.2 Unsupervised learning

Unlike supervised learning, in unsupervised learning the outcome of the input values is not known. The main goal of the unsupervised learning is to observe the relationship and pattern among the input variables. That means it tries to visualize if similar items are placed in one position and dissimilar items are placed in another position. One good example of unsupervised learning is the email spam detector or network intrusion detector. In the email spam detector emails are classified whether they are spam or normal email using relative frequencies of keywords, punctuation marks and other similar techniques. However, this type of machine learning is not studied very well and sometimes is confused with supervised learning. [33].

2.5.3 Reinforcement learning

Reinforcement learning is also called semi-supervised learning. It allows a machine to automatically learn from the feedback that it receives from its environment. Once it starts learning, it adapts as time goes by. It resembles human learning style and learns from past-experience. If it encounters a similar problem, it acts based on what it has learned. One example of reinforced learning is the large collection of pictures, some are labeled as cat, cow or dog, and the majority are not labeled. In contrast to supervised learning where a teacher is needed to supervise the outcome, reinforcement learning does not need a teacher. Rather the student tries to learn by himself and gets a reward for what he did good and penalized for the error. The main task of this learning is that the agent must read the environment and act accordingly. [14]

3 Network slicing and 5G features

This chapter discusses various features of next generation 5G mobile networks. It also presents different 5G use cases such as Ultra Reliable Low Latency Communication (URLLC), Enhanced Mobile Broadband (eMBB), Massive Internet of Things (MIoT), etc.

The ever-increasing demand of network capacity along with the introduction of Internet of Things (IoT) is becoming a challenging task for network operators. Network slicing is proposed as a solution to meet the requirements of the growing demand while meeting bandwidth and delay requirements to selected traffic. The term slicing has been in use for quite some time in the networking world, however this term is now associated with network virtualization: different logical networks are functioning under a single network infrastructure. Traditional network architecture follows the one size fits-all style, that means there is no separation in infrastructure. In the traditional mobile network both network and service provider are functioning as one. It limits the flexibility and network management, which does not encourage new business innovation and customization of network infrastructure. The terms network slicing and virtualization allow having multiple network service providers also known as tenants operating under a single physical network infrastructure using Software Defined Networking (SDN) as shown in Figure 5. [35].

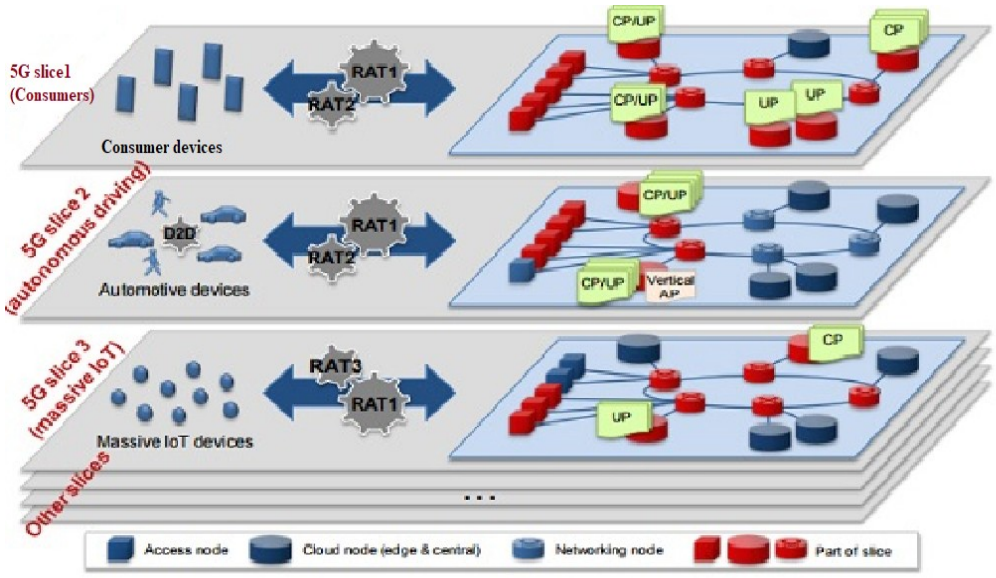


Figure 7 Network slicing architecture (adapted from [35])

Figure 7 demonstrates how multiple network services are running on a single network infrastructure in the 5G network.

The intention is to take a single network infrastructure, spectrum resources, network equipment, devices, etc. and create multiple subnetworks or slices with different requirements. Each slice shares the same physical network with other slices but with its own specific requirements such as bandwidth, security, reliability, latency, charging etc. [36]

Next generation mobile network, 5G, will provide diversified services. Those services include enhanced Mobile Broad Band (eMBB), ultra-Reliable Low Latency Communications (uRLLC), massive IoT (mIoT), etc. Different services require different control functions which the existing mobile networks cannot handle. The existing mobile networks instead provides a single logical control function for diversified services. [37] Next generation mobile networks, 5G, follow service-oriented network architecture where logical control functions can be abstracted as an independent function components. This form of architecture helps provide diversified services in mobile networks.

Next generation mobile networks are expected to be more agile where network slicing services are dynamically generated, maintained or terminated according to their requirements which greatly benefit service providers to increase their revenue.

3.1 Network slicing

It is difficult to achieve the required quality of service and performance for different use cases associated with the introduction of 5G technology. Network slicing in mobile networks is referred as dividing the mobile broadband networks into multiple end to end virtual networks. This allows that each service or use case has its own separate network architecture with different specific requirements and network functions to meet those requirements. Each slice or network tenant shares the same physical network infrastructure with other tenants and different slices. Network slicing technology encourages innovation of new business application, services etc. It provides customized network operations for third parties who lack physical network infrastructure to run their services and applications.

Resource allocation for a specific slice depends on the type of service and application associated with that slice. For example, some applications/services require low bandwidth and tolerate latency while others require high bandwidth, are very sensitive to jitter and latency such as autonomous driving vehicles which require low latency communications.
The flexibility of Software Defined Networking (SDN) to dynamically allocate network resources plays an important role in network slicing and virtualization of network functions. It helps creating and managing end-to-end slices, isolation of services, dynamic and flexible allocation of resources. [38]. The dynamic allocation of

resources helps managing network resources efficiently and equitably for the network slices.

3.2 Slicing in network transport

Software Defined Networks (SDN) and Network Function Virtualization (NFV) are supposed to be potential enablers to create and manage end to end slices or use cases which have their own specific requirements. There are three important players which need to be clear while discussing about end-to-end network slicing. These are the infrastructure provider, tenant and the end user.
Network infrastructure provider refers to the owner of the network physical infrastructure and its resources such as data centers, physical switches and packet transport links. Such resources are virtualized and provided to tenants through Application Programming Interface (API).
Tenants- are service providers who lease virtual resources from a single or multiple infrastructure provider and provide their service to their end users.
End users-are individuals or groups of people who are using the services provided by tenants. [39]

Isolation of end-to-end network slices is one of the requirements that need to be met while running slices on top of a common infrastructure. Some of the advantages of isolation of network slices are:
Performance- as discussed earlier, each slice is designed to meet specific requirements. Hence regardless network congestion, performance levels of other slices, it has to meet those requirements.
Security and Privacy- with massive deployment of Internet of Things (IoT) slices, security and privacy could be the biggest challenge. Attacks and faults should be properly managed and those attacks and faults happening on other slices should not affect its operation. If access to slices is controlled by subscription, devices that have same subscription are able to attack anyone on this slice. Regular internet hosts do not see those devices as they may not have suitable subscription. Each slice should have separate security functions to deny read and write access on the slice configurations for nonauthorized forces. Therefore, in transport network resource allocation and security isolation is very important in end-to-end network slicing. [39]
Management- Each network slice should be treated as a separate and independent network. A Network slice need to have appropriate rules and policies on how to manage and maintain the resources and mechanisms to operate. [40]
Customizable and elastic- network slices could be added or removed dynamically based on the available resources. In case of un-predictable emergency situations, the network should dynamically re-allocate available resources to support new ad-hoc URLLC or other types of slices.

3.3 Network slicing architecture

The control plane of the SDN architecture dynamically configures and abstracts the underlying forwarding plane resources to provide tailored services to the clients on the application layer. [41] The open API in SDN allows dynamic control and automation of slice creation and operation. SDN has two major components- resource and a controller which controls the resource. A resource is something which can be utilized to provide services given to clients upon request. Resources include infrastructure, network devices etc. A controller is a logical entity instantiated in the control plane which manages resources at run time for the efficient use of resources. One of the key concepts of slicing is isolation of resources´.

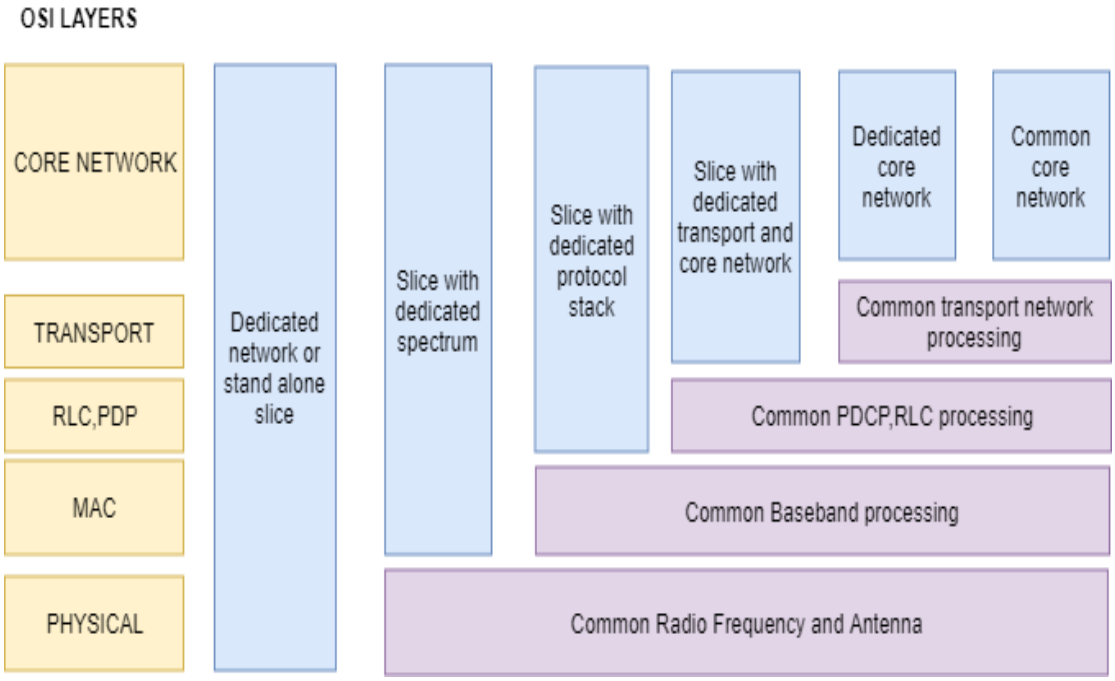


Figure 8 Network slicing architecture

Isolation of resources can be viewed from two perspective. The first one is operational level isolation where vertical customers have full control on the operation of a network slice. Customers can have separate control, monitor and configuration of a network slice. For example, network operators can provide operational isolation with or without network isolation by assigning different identification mechanisms to separate customers which belong to different slices who share the infrastructure. For instance, multiple IOT tenants can share a Narrow Band IoT network which can be assumed as preconfigured network slice.

The second one is network level isolation where vertical customers can have separate network resources which are not shared with other customers.
This level of isolation has different scenarios such as both the RAN network and the core network can be isolated or the RAN can be shared but the core network is isolated etc. [43]

Figure 8 shows designs of network slices in such a way that they can be standalone slices where they could have their own dedicated hardware and spectrum or they could have shared physical layer and spectrum but dedicated transport and core network.

3.4 Network slicing use cases

The idea behind network slicing is that, each slice in the network will have its own functionality and operation which satisfies the requirements of the business customer. Network operators will have a chance to customize their resources, i.e customization enables to create, provide and change network resources based on service demands. [43]

Here are some of the examples of network slicing use cases:
Ultra-Reliable Low Latency Communications (URLLC)- Customers may require operators to provide a URLLC slice for their industrial production to control and monitor their robots in the production line. Customers require low latency or near real time latency, stable and reliable network. Such scenarios require very good quality communications and could not tolerate data loss and it must be assured that a shared infrastructure should not cause any adverse effects on their operation. One good example is, in smart grid we have periodic measurements every few milliseconds sent from 5G device to another. If some of them are lost, there is a risk of a short circuit or shutting the grid down (possibly with half a city going dark). Such type of slice only accepts service specific traffic to avoid negative impact such as congestion on the operation. Service provider has to provide locations of access nodes for terminals to the transport network as well as to the location of the controller. The URLLC service may be limited to a specific area either indoor or out-door area. Therefore, the network operator has to make sure that resources are allocated across different access types and network domains. Monitoring of resources in this service is very important and is measured in the context of performance. Any decline in the performance would require resource adjustment so that the URLLC service will function properly.

Massive IoT- Sensor networks are nowadays very important in collecting data. Such networks are deployed in various areas such as agriculture, manufacturing, weather monitoring etc. Sensor embedded devices (IoT) can be more intelligent and make more frequent interactions with the network and this could be a challenge for the mo-

bile network. Deploying massive number of IoT terminals in the generic network can be complex to manage. One example of this type of network is machine to machine communication which is expected to dominate in industries, healthcare, transportation areas, smart cities. Hence, traditional mobile networks need to be upgraded to the level of creating connectivity fabric for such networks as the nature of different IoT slices may have different control and charging functions so that network operators can manage and deploy them easily and quickly in the next generation mobile networks. [44]

Some of the challenges raised for massive IoT networks are, scalability issues- it is expected that by the 2020, there will be billions of smart devices connected to the network which the 5G network has to deal with. [43] Security and reliability issues are also a big concern as those devices are connected to every household equipment, medical instruments, public facilities etc.
Enhanced Mobile Broadband (eMBB)- eMBB provides better data rates, capacity and coverage relative to Mobile Broadband (MBB). It requires high bandwidth up to 1Gbps and 10 to 40ms latency at the same time. One example of this type of slice is that Augmented Reality/ Virtual Reality (AR/VR) live broadcast. Live sports events, musical concerts, news etc could be broadcasted to users. [43]

3.5 Implementing QoS policy using DSCP

Network traffic is treated with relative priorities based on Type of Service (ToS) field. The network traffic classifier checks every incoming packet for different parameters in the IP header such as source IP address, destination IP address, type of traffic, etc. and assigns it to a specific queuing based of DSCP class value. Most commonly used DSCP values are:
Expedited Forwarding (EF): Packets marked with EF values are assumed as higher priority packets. Those packets require low latency, loss, jitter and guaranteed bandwidth traffic.
Assured Forwarding (AF): packets marked with AF values require reliable delivery for applications which require low packet drop. Those packets have higher priority than default packets.
Default forwarding (DF): packets which do not have the above markings are assigned as default forwarding packets.

3.6 Mobile backhaul Quality of Service

Generally, IP based networks operate on a best-effort bases, which means that all traffic has equal priority and probability of being delivered. Similarly, with best effort when a network becomes congested, all traffic has an equal probability of being

delayed or dropped in the worst case. Quality of Service (QoS) is one way of managing network resources such as bandwidth by selecting network traffic and giving priorities to a specific traffic according to its relative importance. It is possible to limit usage of resources such as bandwidth by a network and make network performance more predictable and resource utilization more effective.

The QoS can be enforced in the packets at link, layer 2 or transport layer 3. The port/ node level QoS classification at layer 2 is based on VLAN priority (PRI) field. This means that all tagged traffic is classified based on a VLAN PRI and untagged traffic classified as best effort (BE). The QoS classification at layer 3 can be done using Type of Service (ToS) if IP transport is used or packet tagging if MPLS is used.

According to Internet Engineering Task Force (IETF) there are two most commonly used quality of service architectures. In practice both architectures work within an administrative domain. ISPs do not trust each other sufficiently to accept to give any preferential treatment to some packets over some other packets.

Integrated Service (IntServ)- uses resource reservation for each flow individually. [38] Every application has to reserve the necessary end-to-end resource in order to ensure the QoS. However, applications do not exactly know how much resource they need and it is difficult to implement and used in small networks.

Differentiated Service (DiffServ)- It is an architecture which specifies a simple and scalable mechanism to classify network traffic using different classes to provide the necessary QoS in modern IP networks. DiffServ is mainly concerned with classifying incoming packets as they enter in to the network using different DSCP values. This classification applies to the flow which is defined by 5 tuples, source and destination IP addresses, source and destination ports and transport protocol. Therefore, a flow classified or marked will be treated according its DSCP value. Currently, before 5G architecture is fully specified and implement, we create network slices using RAN and Transport sharing features supported in 4G networks.

This is not truly network slicing but provides the baseline to analyze the network slicing that will be supported in 5G networks.

3. 7 RAN sharing

RAN sharing consists in having the base stations broadcasting more than one PLMN ID at the same time and each PLMN would be associated to a network slice. Transport sharing means sharing the backhaul connection. The eNB has a single slot for a backhaul connection, so connections to different cores must be multiplexed over the same physical cable through VLAN tagging. Each slice is then a combination of

different PLMN ID with own VLAN. The traffic associated to each slice can be assigned different QoS using Differentiated service code Point (DSCP) packet marking.
Therefore, we will evaluate the reliability of 4G-based network slicing with RAN and Transport sharing using DSCP based traffic classification. The eNB in 4G will assign the traffic from each PLMN to a different VLAN with the assigned DSCP packet marking.
In the network switches after the eNB, the network traffic classifier checks every incoming packet for different parameters in the IP header such as source IP address, destination IP address, type of traffic, etc. and assigns it to a specific queuing based of DSCP class value.

3.8 Radio Link Quality of Service

In the radio link 3GPP has defined a mechanism to deploy Quality of Service. The QoS class identifier (QCI) is a scalar value which ranges 1-254 and classifies QoS to which bearer it belongs. Based on the type of resource, the LTE/LTE-advanced supports two types of bearers, the Guaranteed Bit Rate (GBR) and Non-Guaranteed Bit Rate (NGBR). The GBR is the minimum amount of bit rate assigned to the user equipment by the GBR bearer. It needs to be specified for downlink and uplink separately. The GBR bearer can be set and modified on demand and reserves network resources based on the GBR value associated with it.

In the case of Non-GBR, it experiences packet loss due to congestion as it does not reserve network resources.
Each QCI value is associated with asset of QoS attributes such as priority, packet delay budget and acceptable packet error loss.

The QCI values are used to regulate packet forwarding treatments for scheduling weights, queue thresholds, link layer configuration between user equipment and PDN gateway and IP DSCP mapping. [44] [45]

Table 2 Characteristics of LTE standardized QCIs

QCI	Resource Type	Priority	Delay	PELR	Examples
1	GBR	2	100ms	10^{-2}	Conversational Voice
2		4	150ms	10^{-3}	Conversational Video
3		3	50ms	10^{-3}	Real-time games
4		5	300ms	10^{-6}	Non-Conversational Video (Buffered Streaming)
65		0.7	75ms	10^{-2}	Mission critical user plane Push To Talk voice
66		2	100ms	10^{-2}	Non-Mission critical user plane Push To Talk voice
5	Non-GBR	1	100ms	10^{-6}	IMS signaling
6		6	300ms	10^{-6}	Video (Buffered streaming), web, email, ftp
7		7	100ms	10^{-3}	Voice, Video (live streaming), interactive games
8		8	300ms	10^{-6}	Video (buffered streaming), web, email, ftp
9		9			
69		0.5	60ms	10^{-6}	Mission critical delay sensitive signaling
70	Non-GBR	5.5	200ms	10^{-6}	Mission critical data
79		6.5	50ms	10^{-2}	V2X messages

Table 2 shows the standardized QCI values and their corresponding characteristics. 3GPP has already standardized some of the QCI numbers. For example, best effort traffics are assigned QCI value 9. QCI values 1 and 5 are assigned for voice traffics and IMS signaling. QCI values 65 and 69 are for GBR bearers used for Mission Critical user plane (MCPTT Voice) and NGBR bearers used for Mission Critical signaling respectively whereas QCI value 70 for NGBR are used for mission critical data. [46]

4. Mobile backhaul orchestrator design

This chapter mainly discusses the design and architecture of various technologies used to implement the project. It presents an in-depth analysis on the design procedures of the Mobile Backhaul Orchestrator (MBO) as well as the implementation mechanism.

4.1 Machine learning mobile backhaul

In this section we consider the usage of SDN based network slicing as part of the Mobile backhaul orchestrator for managing the mobile backhaul through machine learning algorithms. The end to end solution requires network slicing in the radio access network as well as in the mobile backhaul. However, in this thesis due to the lack of radio network slicing we will focus in mobile backhaul slicing and evaluate the benefits of machine learning.

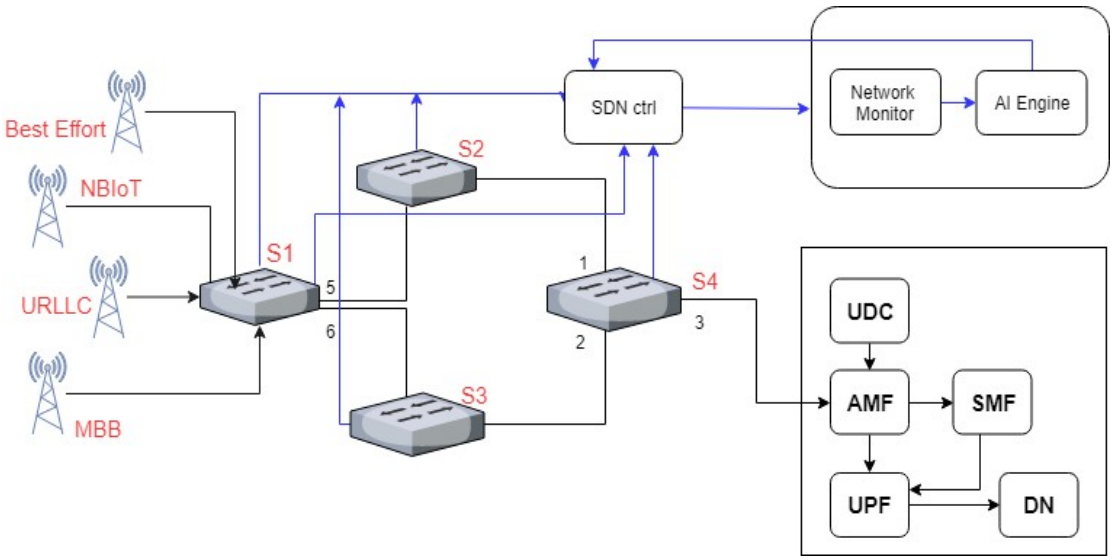


Figure 9 MBO modules

In this thesis we propose a Mobile Backhaul Orchestrator (MBO) that will create and manage network slices based on SDN and machine learning logic to utilize available resources on each slice efficiently. The MBO design includes the modules depicted in Figure 9; 1) Network monitoring to check available resources and 2) Machine Learning engine that will apply different algorithms to estimate optimal routes and suggest them to routing algorithms to decide which routes to use.

Network Monitoring
The network monitoring is implemented as an application on top of the SDN controller. The SDN controller (e.g. Ryu) receives traffic and collects statistical network information from each device(switch) in the network every 30 minutes. We divide the collected information in two parts, the port statistics and flow statistics.
Flow statistics: indicates the type of packets, packet size, duration, number of packets that match the specific flow. Those flow statistics are gathered as a reply to the flow stat request message sent to the switches.
Port statistics: port statistics of a specific switch can be collected by sending port stat request message to the device under investigation. The reply message includes the number of transmitted and received packets, total size of received and transmitted bytes, time stamp, number of error packets received, port number of the specific switch and its id number, packet drops of both received and transmitted packets.

The network monitoring module is interpolating a specific information such as byte count, time stamps and transfer them to the route optimizer for further processing to identify and optimize the usage of resources.

Machine Learning Engine
This consists of the module that collects information from the network monitoring and predicts optimal routes for each slice. The ML engine is the key component of the MBO, which is designed in this thesis and includes machine learning technology to provide optimal usage of resources.

The MBO will start collecting the network statistics using the network monitoring module after sending stat request messages to every device in the network every 30 minutes. The network devices start to respond their status to the network statistical info which sends it to the Machine Learning engine for further calculations.

4.2 Validation of ML in MBH

This section we validate the proposed solution for managing and guarantee SLA in network slices using machine learning. In order to measure the improvements, we emulate a medium scale mobile backhaul using Mininet.

4.2.1 Mobile backhaul emulation

The validation of the MBO design is performed in a virtual environment using Mininet [47] simulator tool. Mininet is a network simulator tool which helps creating a virtual network on a single machine. The virtual network performs network

operations like real network traffic operations using a single command. Mininet is written using python scripting language and it is easy to implement different scenarios and has many commands to monitor its functionality.

In this thesis we perform the test using i5 Dell latitude laptop with 8GB RAM and 128G HDD. Mininet image version 2.2.2 was loaded in Oracle virtual box. Tools such as Wireshark [48], OVS (Open Virtual Switch) [49], Ryu controller, etc. are also installed in the Mininet image which has 1GB RAM and 10 GB HDD. The test environment mainly consists of three elements; network devices, SDN controller and the MBO.

Network devices: those devices are open flow enabled switches which route traffic coming from the mobile base stations towards the core network and vice versa. The edge switches are connected to one or more eNBs that are identified by their IP addresses. Each base station is connected to several user equipment depending on its capacity. The mobile backhaul network orchestrator will have multiple switches connected to the evolved packet core (EPC).

The network devices are connected to the SDN controller (e.g., Ryu) controller using Open Flow protocol for receiving commands on how to route packets to their destination.
Table 2 shows different open source SDN controllers and the features they support.

Table 3 Comparison of SDN controllers

Features	NOX/POX	Ryu	Flood light	Open-day-light
Language support	Python	Python	Java	Java
Opensource	Yes	Yes	Yes	Yes
Network monitoring	Partial	Yes	Yes	Yes
REST API	No	Yes	Yes	Yes
Traffic engineering	Partial	Partial	Partial	Yes
Platform support	Linux, mac, windows	Linux	Linux	Linux, mac, windows.

SDN controller (Ryu): an SDN controller is considered the brain of the network as it controls and manages the traffic flow to and from the open flow enabled switches. Open source SDN controllers such as Ryu contain different modules that perform various tasks. Those tasks include identifying network devices available in the

network, gathering statistical network information, installing or deleting paths in the network according to the network status. Adding some functionality to the modules by running different algorithms provides additional flexibility and capability of the network.

Ryu is the controller selected in this thesis for running the validation of the MBO optimization because it supports open flow version 1.3 and above and is updated continuously relative to other controllers, as large library support, and it is easy to implement as it is written in Python relatively easy to add new components.

Mobile Backhaul Orchestrator (MBO): is the module that includes Network monitoring and Machine Learning components that utilize SDN controller to create and manage network slices. The MBO interacts with the SDN controller to collect network statistics and utilize Machine Learning to optimize the network resources for the network slices.

4.2.2 MBO validation

In order to validate the proposed design of MBO we will emulate the deployment of network slices and QoS traffic prioritization based on 4G mobile networks. Currently, before 5G architecture is fully specified and implemented, we create network slices using RAN and Transport sharing features supported in 4G networks. This is not truly network slicing but provides the baseline to simulate the network slicing that will be supported in 5G networks. Therefore, to validate the proposed MBO we will utilize the current 4G-emulated network slicing. RAN sharing consists in having the base stations broadcasting more than one PLMN ID at the same time and each PLMN would be associated to a network slice. Transport sharing means sharing the backhaul connection. The eNB has a single slot for a backhaul connection, so connections to different cores must be multiplexed over the same physical cable through VLAN tagging. Each slice is then a combination of different PLMN ID with own VLAN. The traffic associated to each slice can be assigned using Differentiated Service Code Point (DSCP) packet marking. Therefore, we will evaluate the reliability of 4G-based network slicing with RAN and Transport sharing using DSCP based traffic classification. The eNB in 4G will assign the traffic from each PLMN to a different VLAN with the assigned DSCP packet marking.

4G based network slicing

We first simulate a simple network topology to measure the QoS prioritization and network slicing based on 4G mobile networks. Moreover, in this first test case we simplify the topology as shown in Figure 8 where we use two OVS switches (i.e. S1 and S2) and 3 hosts (eNBs). Moreover, we use differentiated services to evaluate how effective the usage of network resources in different network slices. We measure how

the network traffic behaves with various DSCP ToS (Type of Service) values assigned to different network slices.

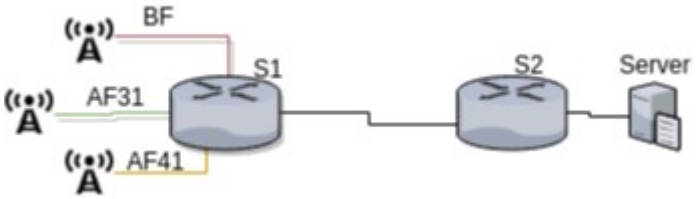


Figure 10 Simple test case topology

The RAN and transport sharing allow a eNB to classify the traffic with different DSCP for each slice. However, for the sake of clarity this first scenario we consider that each eNB provides traffic marked with different ToS classes assigning different DSCP values. We apply the SDN rules on each switch to assign different traffic priorities to each incoming traffic flow. In order to check the impact of congestion with traffic marked with different DSCP priorities, the output link is limited to 1Mbps. The traffic coming from different eNBs is marked with different DCSP values as if they were flowing from different network slices. In this first simulation we check the effect of DSCP associated to different network slices. In the simulation a REST API is used to configure the interface of S2 for adding queue settings, assign DSCP values to the queues, IP addresses, bandwidth requirements of each slice. For example, a sample configuration shown below, assigns queue 1 with a minimum bit rate of 300Kbs and maximum bit rate of 1Mbs to the traffic coming from eNB2 (AF31):

"curl -X POST -d' {"port_name": " s2-eth1","type":"linux-htb","max_rate":"1000000","queues":[{"max_rate":"1000000"}, {"min_rate":"300000"},{"min_rate":"600000"}]}' http://127.0.0.1:8080/qos/queue/0000000000000002".

With the same REST API we configure the switch S1 for marking the DSCP values and the ports to which the S1 is connected to.

"curl -X POST -d' {"match": " {"nw_dst":"ip_address","nw_proto":"udp","tp_dst":"5002"},"actions": {"mark":"26"}}]}' http://127.0.0.1.:8080/qos/queue/0000000000000001"

Once the configuration of the queues is ready, we run the Mininet with the topology shown in Figure 10 and the Ryu controller module simultaneously. Iperf is used to generate traffic and measure performance in IP networks. For each Iperf test, we measure the bandwidth, jitter, throughput, time stamps and other parameters. Iperf

indicated the statistics on the number of packets transmitted in a given time interval through a given link. As shown in Figure 8, the eNBs are connected to S1 and we run Iperf client on each eNBs(hosts). The S1 switch is connected to S2 where we run an Iperf server.

Table 4 DSCP values for different slices for simple topology

Slice name	Queue ID	Max rate	Min rate	DSCP	Port number
Best effort (BE)	0	1Mbps		0	5001
Low drop slice (AF31)	1	1Mbps	300k	26	5002
Low drop (AF41)	2	1Mbps	600k	34	5003

Traffic is generated from the eNBs to the server based on the specifications given in Table 4 above. The traffic from each eNB is assigned to different network slice in the switch S1 which delivers the traffic to the switch S2 through the same link but using different ports. We run Iperf clients and server using the following commands.

Client side:
Iperf -c 172.16.20.10 -p 5002 -u -b 300k
Iperf -c 172.16.20.10 -p 5003 -u -b 600k
Iperf -c 172.16.20.10 -p 5001 -u -b 1M

Server side:
Iperf -s -u -i 1 -p 5001
Iperf -s -u -I 1 -p 5002
Iperf -s -u -I 1 -p 5003

The S2 switch where we have the Iperf server listening to different slices on different ports. Traffic flows coming through ports 5002 and 5003 have the minimum bit rate assigned to AF31 and AF41 traffic respectively and must be given priority over the best effort traffic coming through port 5001 with a maximum bit rate.

As shown in Table 4, traffic marked with ip_dscp value 26(AF31) is sent through port 5002 with minimum rate 300kbps in a 1M link capacity and traffic marked with 34 (AF41) is sent through port 5003 with a minimum link capacity of 600k.Traffic marked with ip_dscp value 0 which means best effort traffic is sent through port number 5001.
We perform the same measurement in different scenarios to see how the traffic behaves for different network parameters.

Scenario1: generating traffic with minimum rate to AF31, AF41and best effort traffic. In this scenario, we generate 300k and 600k traffic for AF31 and AF41 respectively

which is the minimum rate assigned and 1M traffic for the best effort traffic and observe the maximum throughput achieved for each slice.

Figure 11 shows that traffic flow marked with DSCP is enforced in the switches and each slice is guaranteed to transfer the minimum amount of bitrates assigned to it.

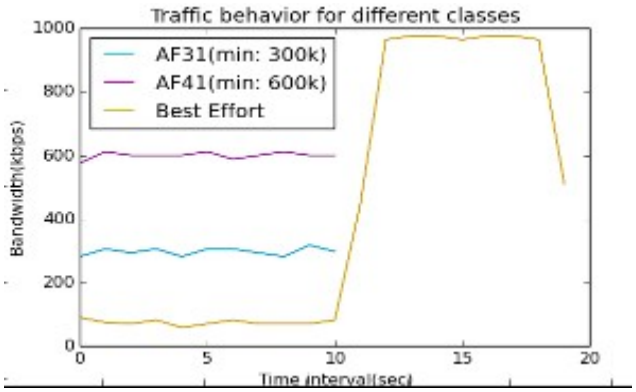


Figure 11 Classifying traffic using DiffServ

Figure 11 shows that the minimum bitrate (600Kbps) which is assigned to traffic marked with AF41, is guaranteed. The same results are also observed for traffic flows marked with AF31, where minimum 300kbps bit rate is achieved. However, in the case of default traffic (best effort), it shows that despite 1Mbs bit rate is assigned, it only gets the remaining capacity of the link. The slice with best effort traffic regains its maximum bit rate only after the AF traffic flows have finished. This shows that in a 1M link capacity, we are trying to send traffic size of 1.9M in total which is beyond the link capacity. In such situation, priority is given to traffic marked with DSCP values which is 900k in total and we are left with only 100k free bandwidth and this free bandwidth is used by the best effort traffic until extra free bandwidth is available.

Scenario 2: the target of this test is to measure end to end delay variation, so we generate different traffic flows in each slice and observe the delay variation. For example, for AF41, we generate the minimum bit rate assigned i.e, 600k first and we then increase the incoming traffic bit rate gradually to 1M, 5M and 20M. We start observing the traffic flows and check the delay variation in each case. The same procedure is performed to check the relationship between delay variation and throughput.

The results are shown in Figure 12. The graph clearly shows the delay variation when changing the amount of data traffic for a network slice marked with guaranteed bit rate.

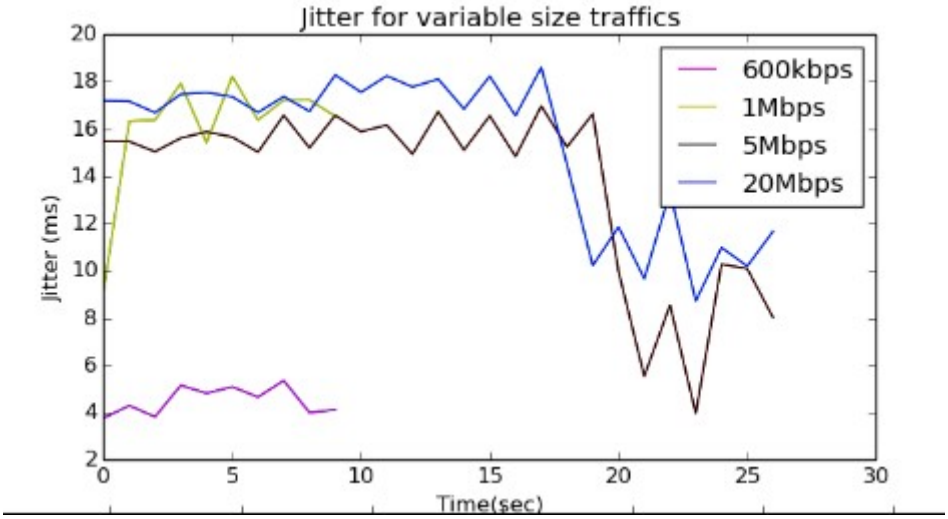


Figure 12 AF41 Network performance for variable data size

In Figure 12, the minimum bit rate assigned for AF41 is assured with low jitter as shown in the pink color. However, if we increase the traffic in the same slice to 1Mbps, which is higher bitrate than the guaranteed by the DCSP marking, the jitter increases exponentially but still can deliver all the packets without any packet drop. If the traffic further increased to 5M and 20M the jitter is similar as 1M traffic but after some time packet drop gets above 90% and packet delivery terminated.
A similar measurement is performed for the network slice with traffic marked with DSCP AF31. The minimum rate assigned for AF31 is 300Kbs, which is assured with low jitter.

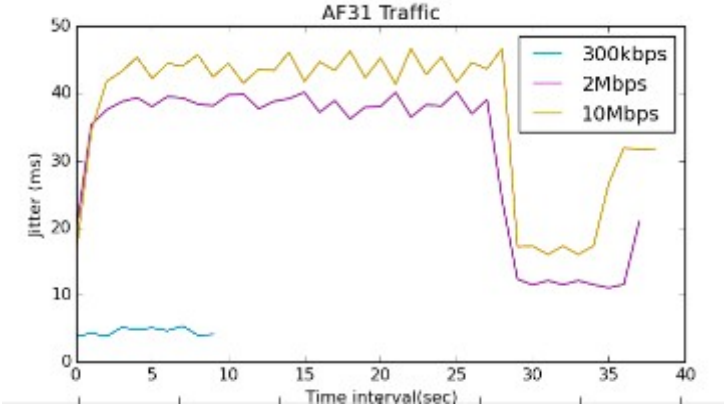


Figure 13 AF31 traffic

However, when the rate of data transfer increases, the same delay variation behavior is observed similar to the case of AF41 measurement. As the amount of data

transferred increases the packet drop and jitter also increases and after a certain interval of time, the packet drop reaches more than 80%.

4G best effort network slice

We measure the network behavior in best effort network slices. Data packets which are not marked with any DSCP value are treated as best effort or default traffic. Those packets marked with DSCP value get higher priority than those with best effort or no DSCP marking at all as shown in Figure 14.

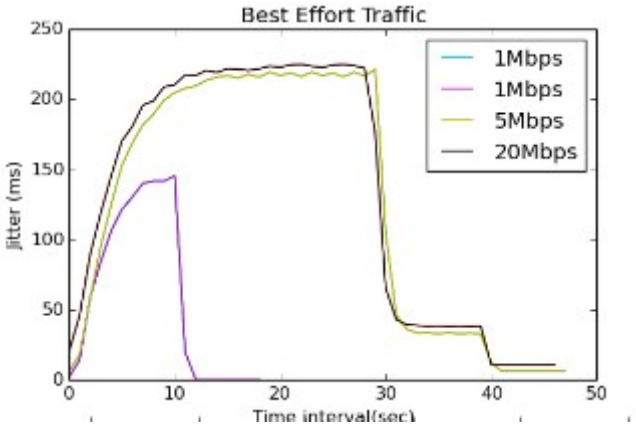


Figure 14 Best-effort traffic

In Figure 14, the pink color shows that jitter is quite high for best effort and suddenly become very low from above 100ms to single digit values for 1Mbps traffic. This means priority is given to other classes and minimum bandwidth is assured to traffic flows assigned with AF classes than the best-effort.

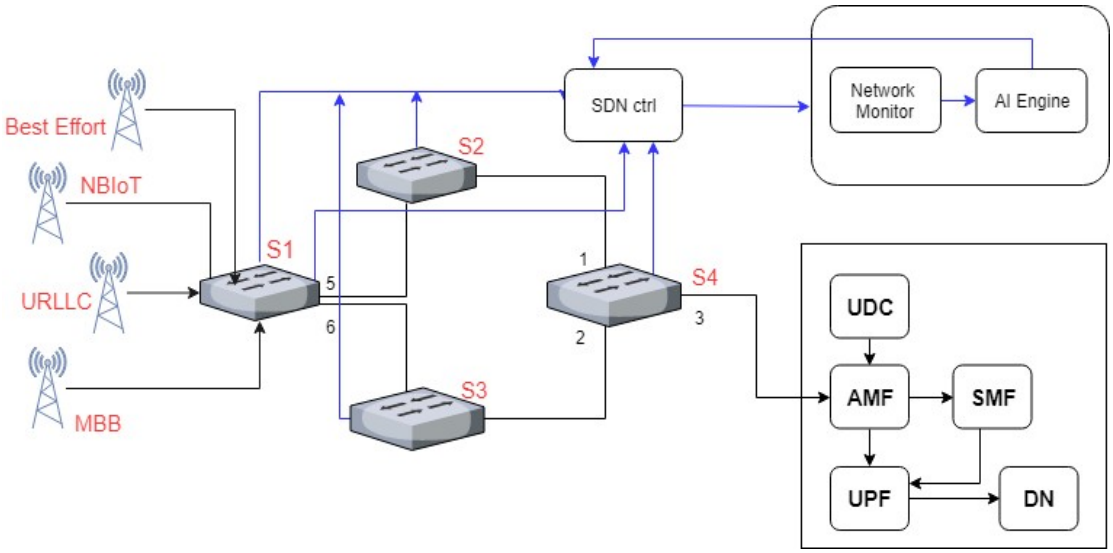


Figure 15 4G slices with large network topology

The same trend is also observed for higher bitrate data transfer of Best effort traffic. If we increase the traffic volume, the jitter also increases, and the amount of packet drop

is very high relative to other classes. Therefore, from all the above tests and observations one can say that DSCP value can be used to configure slices with different parameters such as minimum bit rate, delay, packet loss, etc.

4G network slicing with enhanced network topology

Next objective is to measure the impact of larger scale network topology in the 4G based network slicing. Figure 15 shows the new topology used for the simulation.

Figure 15 shows a larger topology with four switches and four slices: URLLC (Ultra Reliable Low Latency communication), NBIoT, Best Effort traffics and Mobile Broad Band slices (MBB) where each slice is represented with the traffic coming from different eNB. We assign different bit rates for each slice as shown in the table below.

Table 5 DSCP values for different slices for advanced topology

Slice name	IP address	Queue ID	Max rate	Min rate	DSCP	Port number
Best effort (BE)	172.16.10.10	0	10Mbps		0	5001
NBIoT	172.16.10.20	1	10Mbps	1Mbps	AF31	5002
MBB	172.16.10.30	2	10Mbps	1 Mbps	AF41	5003
URLLC	172.16.10.40	3	10Mbps	1Mbps	AF22	5004

As shown in Figure 15, the first edge switch S1 is connected to 4 eNBs which are identified by their IP addresses. The traffic coming from those eNBs is routed through two ports in switch S1. Best effort traffic and the URLLC traffic are routed through port 5 and NBIoT and Mobile Broad Band traffics are routed through port 6 in switch S1. The same route division was also made for the return traffic coming from the server (core network). As shown Figure 15, packets destined to URLLC and BE will be routed through port 1 and MBB and NBIoT packets will route through port 2.

The switches are configured by the Ryu SDN controller using REST API. With this API we configure the queues (see Appendix A) and minimum rate assigned for each queue. Next, we start to observe how the network behaves by varying the traffic volume across each slice. We use Mininet custom topology for testing and run it using the following command:

Sudo mn –custom mbh/MBH_topo.py –topo mbh_topo – controller remote –mac –link tc,bw=10

This command tells the Mininet emulator to start the custom topology written in python script and assign 10Mbps port speed for each link and use remote controller at port 6653. The following command starts the Ryu controller module which contains routing configurations of the switches.

Ryu-manager ryu.app.rest_qos mbh/qos_mbh_controller.py ryu.app.rest_conf_switch

Scenario 3: This scenario measures the impact of the different bitrate in the network slices based on DSCP priorities. We generate traffic using iperf from the eNBs to be sent to the server (Core network) through the different switches in the network. We first generate traffic using the minimum bit rate assigned in the table above and observe how the maximum throughput is achieved using the following iperf commands.

Client side:

URLL: Iperf -c 172.16.10.50 -p 5004 -u -b 1M

MBB:Iperf -c 172.16.10.50 -p 5003 -u -b 1M

LBW-IoT:Iperf -c 172.16.10.50 -p 5002 -u -b 1M

BE: Iperf -c 172.16.10.50 -p 5001 -u -b 10M

Server side:

Iperf -s -u -i 1 -p 5001

Iperf -s -u -i 1 -p 5002

Iperf -s -u -i 1 -p 5003

Iperf -s -u -i 1 -p 5004

Those commands show that the clients generate the specified amounts of traffic and the server is listening for incoming traffic in a specific port listed above and the server sends statistics information every second.

The results in Figure 16 show similar trends as the previous scenarios performed using simple network topology of two switches. In this scenario we extend the network topology to use more switches as shown in Figure 15. The measurement results show that the minimum bit rate assigned for each slice marked with DSCP value is guaranteed.

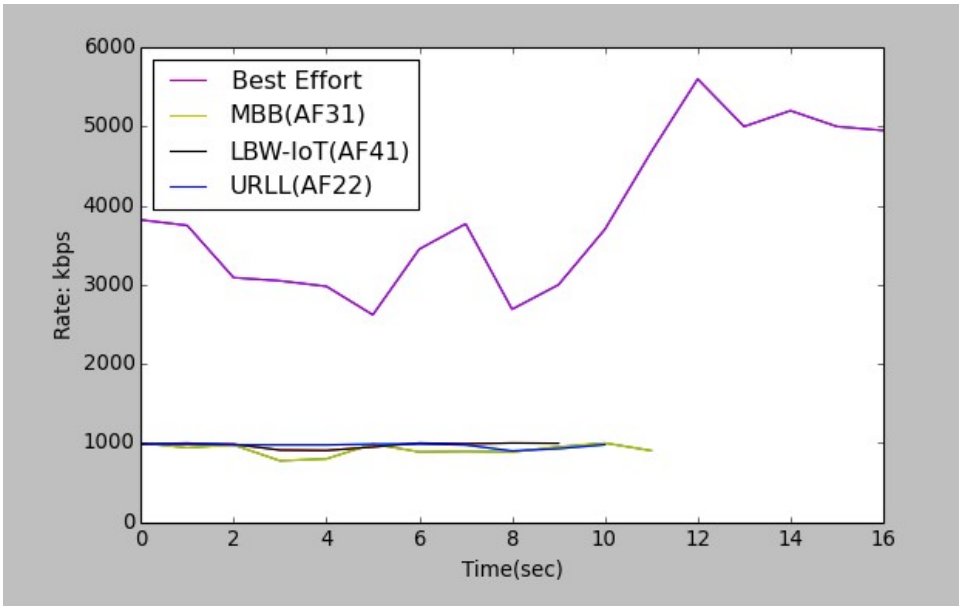


Figure 16 Minimum achievable throughput

In Figure 16, we can see that minimum bit rate is guaranteed for traffic coming from URLLC, LBW-IoT and MBB which is 1Mbps. However, the best effort traffic, uses the remaining bandwidth available. Once URLLC traffic finish its transmission, the best effort traffic starts using the remaining free bandwidth and the throughput increases.

Scenario 4: In this scenario we observe end to end delay if we increase the traffic volume. We performed the test in this scenario by increasing the traffic volume for the URLL but keeping other traffic bitrates constant. Hence, we can observe that traffic is blocked totally because of traffic bitrate is beyond the link capacity.

Figure 17 shows that if the rate increases beyond the minimum assigned rate, for example when performing tests with 600k,1M,3M and 5M rates, we observe delay variation increases as the traffic volume increases beyond 1Mbps. In this scenario we take the URLLC traffic as an example to show the delay variation by increasing the data bitrate but keeping other traffics on their minimum rate (1M). Figure 17 shows, it is possible to guarantee 600k -blue line and 1Mbps, red line, with lower jitter and with no packet drop.
Increasing the rate to 3Mbps, green line, shows no packet drop but the jitter increases reasonably

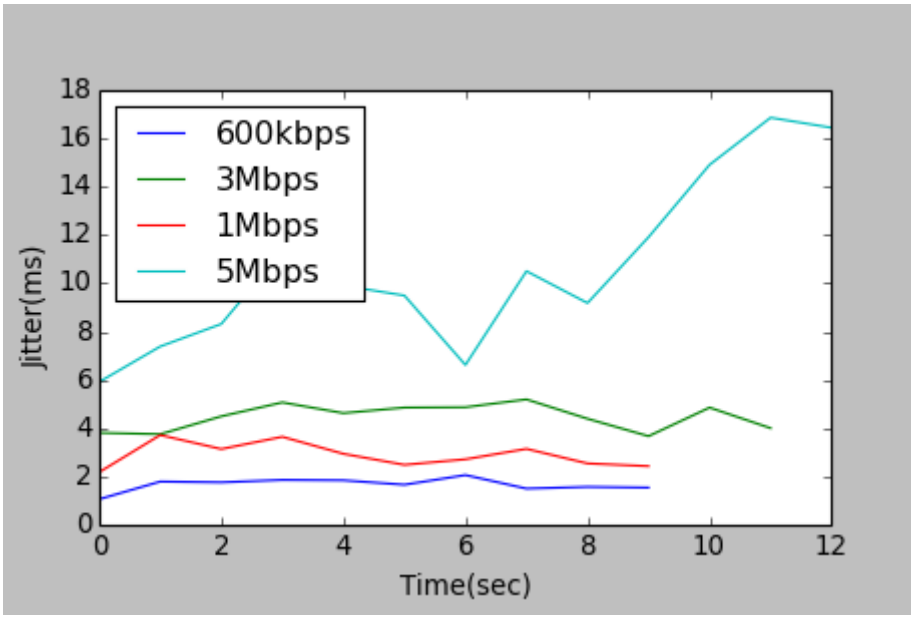


Figure 17 URLL traffic behavior

However, increasing the rate to 5M as shown in the light blue line, the jitter increases tremendously, and packet drop is observed and as time goes on, the packet drop also increases until it reaches to the point where all packets are dropped.

MBO network slicing

As explained in the previous tests, it was possible to observe how different traffic flows can be configured to meet specific requirements. It has also been seen how the traffic behaves if someone transfers excess traffic volume beyond the specified bit rate. We observed in the previous tests that trying to use excess rate causes congestion on the network, delay and ultimately leads to network blockage. The problem is how to avoid congestion in case we are interested to add a new slice while ensuring the minimum bit rate assigned to each slice without disrupting or with minimum disturbance to the existing slices. The aim of the proposed MBO is to answer such a question. The objective of this section is to compare the efficiency of MBO to manage available resource to guarantee the minimum resources assigned to each slice.

The MBO includes the AI engine that will utilize machine learning algorithms to estimate optimal paths based on different features or attributes. Those attributes include link capacity, packet loss, latency, hop count etc. The AI engine takes those attributes and uses different machine learning algorithm to estimate the optimal paths and guide the controller to decide which path to take to deliver packets with minimum delay. In this case we validate the network optimizer using Machine learning techniques to evaluate and predict the congestion level of a link.

From the different types of machine learning techniques reviewed in previous sections, we selected the supervised learning. The idea behind supervised learning is that, for some inputs, we want to have certain value as an output. The machine learning algorithms run based on the inputs until getting output values close enough to the target value. We use Artificial Neural Network (ANN) to perform the tests. Artificial Neural Networks is a network made of multiple neurons. A neuron is a building block which takes one or more inputs and pass through some mathematical functions, in our case sigmoid function to produce an output. A sigmoid function is used to unbound the input to output. For implementing Neural Network (NN), we use a powerful python library called NumPy. A Neural Network has three layers: Input layer, Hidden layer and Output layer. We use 3 attributes BW, packet loss and hop count as an input. The hidden layer is a layer between the input and output layers. There could be a single or multiple hidden layer. For simplicity we use a single hidden layer.

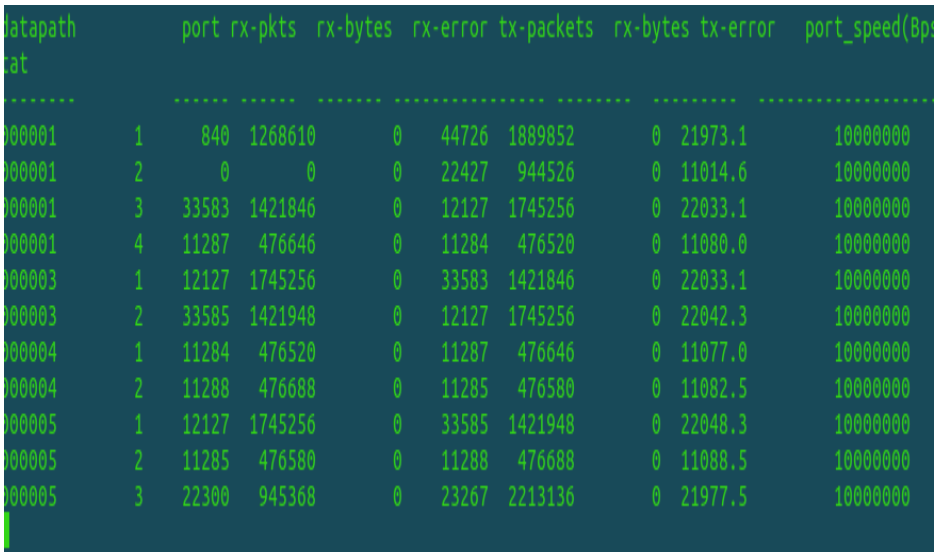
Initially each input is multiplied with some random number called weight and are summed up and pass through an activation function to generate a neuron in the hidden layer. The generated hidden layer is now an input to the output layer or to another hidden layer and pass through an activation function and finally get the output. The process of getting an output value from a given inputs is called Feed Forward (FF) process. However, the output might not meet or close enough to the expected output, hence we perform the same process but backwards to find and replace the random weights with some reasonable values using partial derivatives and this process is called Backward Propagation (BP). Once we find new weights, we perform the same

Feed Forward (FF) process to find the final output. In this case we might perform multiple FF and BP processes until we get the desired output before going to testing. The output value from the network features (inputs) is somewhere between 0 and 1. Value close enough to zero means the links in question is less congested and value close to 1 means there is high probability that the link is congested.

Therefore, using this technique we evaluate if a given link is congested or not based on the network information collected from the network. About 1500 sample data was collected for one and half hour. From the data collected 30% of it is used for training and the rest 60% of the data for testing.

To perform the tests, we identify which ports are more exposed to congestion and apply traffic engineering on those switches connected to those ports. Figure 15 shows that there are two routes from S1 towards the server through port 5 and port 6 and two routes for the reverse traffic from the server (core network) through port 1 and 2 at switch 4. Therefore, we implement traffic engineering on those ports. The MBO network monitoring collects information from the traffic flow at both switches and the Machine Learning engine predicts how the network traffic behaves in different situations and performs different tasks in case of congestion happens to those ports. As packets start to arrive at the switches the Network monitor module starts to collect traffic information periodically by sending port and flow stat request method to the Ryu controller every 5 seconds. The Network monitoring engine generates traffic statistics which describes the status of the ports of each switch.

A sample screenshot of the network parameters collected from each switch is shown in figure 18.



datapath	port	rx-pkts	rx-bytes	rx-error	tx-packets	rx-bytes	tx-error	port_speed(Bp
cat								
000001	1	840	1268610	0	44726	1889852	0	21973.1	10000000
000001	2	0	0	0	22427	944526	0	11014.6	10000000
000001	3	33583	1421846	0	12127	1745256	0	22033.1	10000000
000001	4	11287	476646	0	11284	476520	0	11080.0	10000000
000003	1	12127	1745256	0	33583	1421846	0	22033.1	10000000
000003	2	33585	1421948	0	12127	1745256	0	22042.3	10000000
000004	1	11284	476520	0	11287	476646	0	11077.0	10000000
000004	2	11288	476688	0	11285	476580	0	11082.5	10000000
000005	1	12127	1745256	0	33585	1421948	0	22048.3	10000000
000005	2	11285	476580	0	11288	476688	0	11088.5	10000000
000005	3	22300	945368	0	23267	2213136	0	21977.5	10000000

Figure 18 Sample screen shot of network parameters

Figure 18 shows that parameters such as packet count, transmitted and received packet size of each port, time duration during which packets are transferred, also the

hardware information such as port capacity, round trip time and number of packet errors are collected by the Network Monitoring module through the SDN controller. The Network monitor module sends all the collected data to the AI engine module. The AI engine starts to monitor the port status of each device and evaluate against the set-up threshold values. For example, once statistical information of each device is collected then the AI engine starts to calculate how much bandwidth is utilized so far and how much underutilized bandwidth is available in each port, packet loss, jitter. Those parameters are calculated using the following formulas.

$B_T = (T_{xi}+T_{ri})/t_i \quad B_{T-1} = (T_{xi-1}+T_{ri-1})/t_{i-1}$
Where T_{xi} = Transmitted Bytes at time t_i, T_{ri} = Received Bytes at time ti
$BW_{ratio} = ¿B_T - ¿B_{T-1}¿ / maxBW$, Eq (3)
where B_T is bandwidth usage at time T, BW is Bandwidth.

$P_{loss} = (P_{trans} - P_{reci}) / P_{total}$ Eq (4)
where P_{loss} = Packet loss, P_{Trans} = Transmitted packets and P_{reci} = Received packets

$Latency = TByte/Tr_{Rate}$ Eq (5)
where TByte =Total Byte and Tr_{Rate} = Transmission Rate
The packet loss is calculated using the difference between the numbers of transmitted packets and received packets at the corresponding switch port of the path. Transmission latency is the time spent to transfer data from the source to its destination.
The AI engine is set in such a way that if a link uses more than 50 percent of the available bandwidth at port 5 of switch 1 and at port 1 of switch 4 then the AI module assign this port as congested and suggest the controller to install new path and reroute the new slice requests to port 6 and reverse traffic to port 2 , which are normally used by best effort traffic.

Switch ID	Port	Available BW
1	1	8970.00
1	2	10000.00
1	3	6534.00
1	4	10000.00
1	5	5464.00
1	6	10000.00

Figure 19 Bandwidth utilization measurement
The Figure 19 shows the sample screen shot below shows the calculated available bandwidth in kbps in a specific port at switch 1.

Scenario 5: we generate the minimum rate (1M) traffic volume assigned for URLLC slice and 10M traffic for the best effort traffic which is the maximum capacity of the

link. We observe the bandwidth utilization at port 5 but in this case the link was not congested enough to force the AI module to inform the controller about the level of congestion of the port. As shown in the screen shot on Figure 19 above, the available bandwidth at port 5 is not below the minimum threshold. Therefore port 6 is still 100% free as there is no traffic going through the link. We purposely did not generate any traffic for the other two slices at port 6 in order to see the full availability of the link. Next, we start to increase the traffic volume gradually and the AI module is still following the traffic every 5 seconds and calculates the necessary parameters. The AI engine evaluates against the minimum requirement already set. It then informs the controller about the high possibility of the port being congested and the routing algorithm reroutes new slice requests to less congested paths which in our case port 6.

The results show that initially, the link was not congested with a minimum 1 M rate traffic for the URLL slice and 10M for the best effort traffic.

However, as the bit rate increases the jitter and latency also increases while the available bandwidth starts to get lower and lower and drops below 50% of the link bandwidth and at this point the routing algorithm is forced to act based on AI suggestion and reroute the new slice arrival to port 6 and squeeze the best effort traffic as shown in the Figure 20 below.

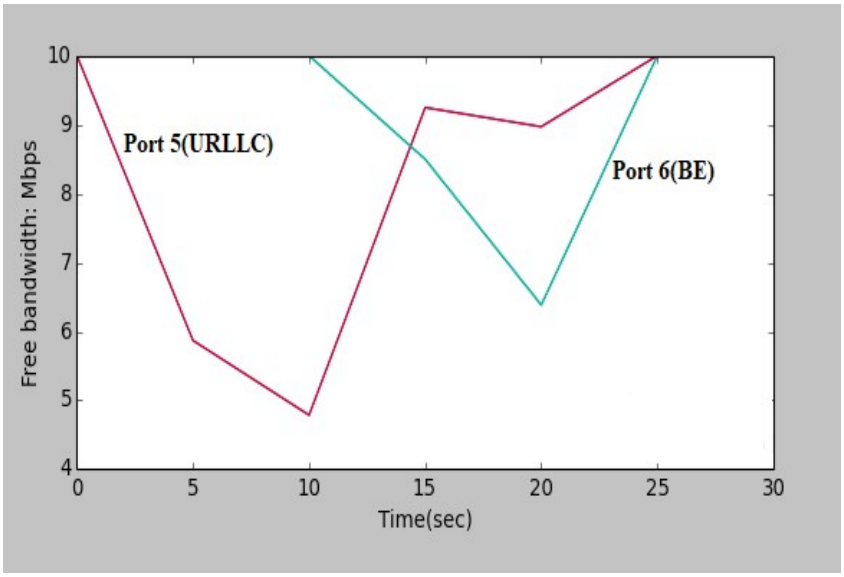


Figure 20 Bandwidth utilization

From Figure 20, we observe that both ports connected to switch S1 are underutilized initially and they have 100% available bandwidth. When packets start to pass through port 5 represented with the red line in Figure 20, the available bandwidth starts to get lower and lower meanwhile port 6 represented with the green line, is not utilized at all. When traffic increases, the available bandwidth at port 5 reaches to its lowest level which is below 50% of link capacity. The SDN controller start to reroute new

slice arrivals to port 6 at the 10^{th} second based on the AI engine results. Therefore, we can conclude that the AI module can perform optimization of resource utilization using machine learning.

5 Discussions and limitations

This thesis studies the current limitations of networking technologies when applied to 5G mobile backhaul. It tries to use technologies such as SDN and machine learning to efficiently manage network slices on top of basic IP routing. Currently, before 5G architecture is fully specified and implemented, we create network slices using RAN and Transport sharing features supported in 4G networks. This is not truly network slicing but provides the baseline to simulate the network slicing that will be supported in 5G networks. Hence, we make an assumption that slice requests with different quality parameters are coming to the backhaul network and based on those parameters traffic is identified to which slice it belongs and routing is performed.

The System is mainly working on residual capacity routing that means it checks the available capacity in each of the alternative paths and takes the better route. However, residual capacity routing does not take other factors such as packet loss, delay, hop count etc. in to consideration but in our system those factors are also taken into account and new routes are applied upon certain parameters passing the threshold.

We use Mininet for simulation purposes because it is easy to manage and can perform network operations like real network traffic operations do. I performed the tests using a laptop computer with a specification presented in the previous chapter. Mininet simulation tool is easy to install and create OVS switches and hosts using a single command but in real hardware this might not be easy to perform in the same way. The other problem was with the OVS. Tests are performed using open flow based OVS-switches using Open Flow protocol. However, the current OF version (1.6) does not yet support GTP (GPRS Tunneling Protocol) which is used for transporting packets in the mobile network between the eNodeB and SGW.

Therefore, packets which belong to different slices might come from the same user equipment (source address) and route to the internet and vice-versa. In such scenarios there are two options, the first option is to parse the incoming packet and identify the TEID (Tunnel End point Id) and map it to which slice it belongs. This is relatively easy but it creates overhead to the network. The other option is to make configuration changes in the Linux kernel to identify the TEID which is complex but still doable.

During testing measurements were taken every 5 seconds as I was using iperf to measure the bandwidth usage and the VM machine I was running the simulator had limited capacity. Random timing interval was chosen just to collect data from the network devices but it would have been better had it been taken in a bigger gap for example every 15 min or 30 min. However, it does not have any influence on the

responsiveness to congestion as the system reacts only when there is congestion. It just collects resource usage information every 5 seconds.

6 Conclusion

The main objective of the thesis was to develop and test a system which measures the use of network resources in the mobile backhaul. The system further redistributes available resources dynamically to different slices either existing or new slices to make sure that each slice requirements are guaranteed without disturbing or with minimum disturbance to the operation of other slices.

Furthermore, the thesis analyses technologies such as SDN and ML to measure the available resources in each link on the network. The ML is used to estimate optimal paths based on different features or attributes collected from the network devices. Those attributes include link capacity, packet loss, latency, hop count etc. Based on those attributes the system estimates the optimal paths and guides the controller to decide which path to take to deliver packets with minimum delay.

The system developed as part of the thesis was designed to be efficient to measure and manage resources. The implementation included different test case scenarios to show how network slices behave in different environments. Those different test cases proves the feasibility of the system and how network resources are measured and managed in an efficient way. It also shows the limitations such that it was performed in a simulation environment and was not tested in a real network environment but part of the thesis was also tested for NBIoT slices in a real network environment and works fine.

The system developed in this thesis can be improved further using real data collected with multiple features from mobile networks and integrating real mobile front and backhaul systems with a real network topology.

Reference

[1] B. Bhargava, Traffic Engineering with Routing protocols using SDN and NFV [online]. California State University, CA; May 2017. Available at: https://pqdtopen.proquest.com/doc/1916517761.html?FMT=AI. Accessed September 7,2018.

[1] Trimponias, G., Xiao, Y., Xu, H., Wu, X. and Geng, Y., 2017. On traffic engineering with segment routing in sdn based wans. arXiv preprint arXiv:1703.05907.

[2] Mendoza, F., Ferrús, R. and Sallent, O., 2017, December. SDN-based traffic engineering for improved resilience in integrated satellite-terrestrial backhaul networks. In 2017 4th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM) (pp. 1-8). IEEE.

[3] Chengiz Alaettinoglu. Overcoming traffic engineering challenges with SDN [online]. South Brisbane, AU; May 2017. Available at: https://blog.apnic.net/2017/05/04/overcoming-traffic-engineering-challenges-sdn/. Accessed June 30 2018.

[4] Biswas, I., Abu-Tair, M., Morrow, P., McClean, S., Scotney, B. and Parr, G., 2017. A Dynamic Approach to MIB Polling for Software Defined Monitoring. Journal of Computer and Communications, 5, pp.24-41.

[5] Greg Fero. “OpenFlow applications work where network management tools fail” [online]. Newton, MA; November 2011. Available at: https://searchnetworking.techtarget.com/news/2240111241/OpenFlow-applications-work-where-network-management-tools-fail. Accessed October 2018.

[6] Hamad, Diyar Jamal, Khirota Gorgees Yalda, and Ibrahim Tanner Okumus. "Getting traffic statistics from network devices in an SDN environment using OpenFlow." Information Technology and Systems (2015): 951-956.

[7] Jason Brownlee. “Your First Machine Learning Project in Python step-by-step” [online]. Vermont Victoria, AU; June 2016. Available at: https://machinelearningmastery.com/machine-learning-in-python-step-by-step/ Accessed at: September 2018.

[8] Latah, M. and Toker, L., 2016. Application of artificial intelligence to software defined networking: A survey. Indian Journal of Science and Technology, 9(44), pp.1-7.

[9] Hartung, M. and Körner, M., 2017. SOFTmon-Traffic Monitoring for SDN. Procedia Computer Science, 110, pp.516-523.

[10] Kimmerlin, M., 2014. Caching in LTE networks using Software-Defined Networking.

[11] Jeong, S., Lee, D., Hyun, J., Li, J. and Hong, J.W.K., 2017, September.

Application-aware traffic engineering in software-defined network. In 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 315-318). IEEE.

[12] Liyanage, M., Okwuibe, J., Ahmed, I., Ylianttila, M., Pérez, O.L., Itzazelaia, M.U. and de Oca, E.M., 2017, June. Software defined monitoring (sdm) for 5g mobile backhaul networks. In 2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN) (pp. 1-6). IEEE.

[13] Bill Kauffmann, "SDN goes mobile:Core concept for Mobile Backhaul Implementation" [Online]. Available: http://telecoms.com/opinion/sdn-goes-mobile-core-concepts-for-mobile-backhaul-implementation/. [Accessed Dec, 2018]

[14] Fan, Z. and Liu, R., 2017, August. Investigation of machine learning based network traffic classification. In 2017 International Symposium on Wireless Communication Systems (ISWCS) (pp. 1-6). IEEE.

[15] Helsinki University, "Elements of AI," [online]. Available at: https://course.elementsofai.com/. [Accessed Dec 2018]

[16] Loakim K.Samaras. "Carrier Ethernet 2.0 services in mobile backhaul utilizing OpenDaylight and OpenFlow" [online] available at: https://schd.ws/hosted_files/opendaylightsummiteuropeanm2016/c7/CE%202.0%20services%20in%20Mobile%20Backhaul%20utilizing%20ODL%20%26%20OF.pdf. [accessed November 2018]

[17] David Stokes. "4G, 5G Mobile Backhaul: What is The Difference For The Consumer?" [online]. November 2016. Available at: https://blog.ecitele.com/4g-5g-mobile-backhaul-what-is-the-difference-for-the-consumer. [accessed February 20 2019]

[18] Moy, J.T., 1998. OSPF: anatomy of an Internet routing protocol. Addison-Wesley Professional.

[19] Medhi, D. and Ramasamy, K., 2017. Network routing: algorithms, protocols, and architectures. Morgan Kaufmann.

[20] Raimo Kantola. "Routing and SDN" [online]. Course material 2017. Available at: https://mycourses.aalto.fi/course/view.php?id=20962. [Accessed December 2018]

[21] De Ghein, L., 2016. MPLS Fundamentals: MPLS Fundamentals ePub _1. Cisco Press.

[22] Πάτρας, Χ., 2015. A study on software defined networks: traffic engineering (Master's thesis, Πανεπιστήμιο Πειραιώς).

[23] Haile Magicho, R., 2014. Application of SDN Concept in Mobile Backhaul for Traffic Optimization.

[24] Juniper Networks. "Mobile Backhaul Reference Architecture" [online]. Available at: http://www.ictnetworks. com.au/pdf/8030008-en.pdf. [accessed November 20 2018]

[25] Braun, W. and Menth, M., 2014. Software-defined networking using OpenFlow: Protocols, applications and architectural design choices. Future Internet, 6(2), pp.302-336.

[26] C. Perera, P. Jayaraman, A. Zaslavsky, P. Christen and D. Georgakopoulos, "Dynamic configuration of sensors using mobile sensor hub in internet of things paradigm," *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, VIC, 2013, pp. 473-478.

[27] P. Waher, and R. Klauck, “Internet of Things-Discovery,” XEP-0347, Sep. 2017.

[28] Github. Available at: https://github.com/pranityadav7/Python-Load-Balancer-Application. [Accessed August 2018]

[29] Github. Available at: https://github.com/pranityadav7/Python-Load-Balancer-Application. [Accessed August 2018]

[30] Nagios. “Nagios Network Monitoring tool” [online]. Availabel at: https://www.nagios.com/solutions/network-monitoring/. [Accessed March 20 2018]

[31] Arnold, M., 2017. Predictive networking and optimization for flow-based networks. arXiv preprint arXiv:1707.06729.

[32] Taniguchi, Y., Tsutsumi, H., Iguchi, N. and Watanabe, K., 2016. Design and evaluation of a proxy-based monitoring system for openflow networks. The Scientific World Journal, 2016.

[33] Friedman, J., Hastie, T. and Tibshirani, R., 2001. The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.

[34] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

[35] Michelle Messenger, “What Exactly is Tensor Flow,” [Online].Available: https://medium.com/datadriveninvestor/what-exactly-is-tensorflow-80a90162d5f1. [Accessed January 2, 2019]

[36] Zahid G., “5G NFV Network Slicing,” [Online]. Available at: https://blog.3g4g.co.uk/2015/11/5g-nfv-and-network-slicing.html. [Accessed March 24, 2019]

[37] Peter Ashwood-Smith,” Why end-to-end Network Slicing will be Important for 5G” [online]. Available at: https://news.itu.int/why-end-to-end-network-slicing-will-be-important-for-5g/. [Accessed January 2, 2019]

[38] Tom Nolle, “Mobile backhaul ofload could affect Evolved Packet Core design” [online]. CIMI Corporation. Available at: https://searchnetworking.techtarget.com/tip/Mobile-backhaul-offload-could-affect-Evolved-Packet-Core-design. [Accessed December 3, 2018]

[39] White paper, “End to End Network Slicing” [online]. Available at: https://www.wwrf.ch/files/wwrf/content/files/publications/outlook/white%20pager/203-End%20t0%20End%20Network%20Slicing.pdf. [Accessed December 3, 2018]

[40] Huang Yan, “End-to-End Network Slicing: Key to Digital Transformation” [online]. ZTE Technologies. Available at: https://www.zte.com.cn/global/about/magazine/zte-technologies/2018/1/Special-Topic/5G-network-Slicing.

[Accessed February 2019]

[41] Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J.J., Lorca, J. and Folgueira, J., 2017. Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. IEEE Communications Magazine, 55(5), pp.80-87.

[42] Open Networking Foundation,"Applying SDN Architecture to 5G Slicing"[online]. 2016. Available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Applying_SDN_Architecture_to_5G_Slicing_TR-526.pdf. [Accessed March 3, 2019]

[43] IETF Trust, "Network Slicing Use Cases:Network Customization and Differentiated Services"[online]. Available at: https://tools.ietf.org/id/draft-netslices-usecases-02.html. [Accessed: February 3, 2019]

[44] GSM Association, "An Introduction to Network Slicing" [online]. Available at: https://www.gsma.com/futurenetworks/wp-content/uploads/2017/11/GSMA-An-Introduction-to-Network-Slicing.pdf. [Accessed February 10, 2019]

[45] Fatima Furkan, "Quality of Service in 4G Wireless Networks" [online]. ARN Laboratory, iNext Research Center, University of Technology Sydney. Available at: https://opus.lib.uts.edu.au/bitstream/10453/36963/2/02whole.pdf . [Accessed April 2, 2019]

[46] Mininet.org. Available at: http://mininet.org

[47] wireshark.org. Available at: http://wireshark.org

[48] ovs.org. Available at: http://ovs.org