Abstraction of Mobile Network Topology

Albert Codes Morales

Supervised by: Dr. Toktam Mahmoodi

King’s College London

March 2015

Abstract

Using the concept of Software-defined Networking to re-design the control plane in mobile network has been studied in the literature in recent years. This new approach to the design of control plane has some trade-offs that have to be considered. This dissertation analyses the effect that abstracting topology of mobile networks would have on the performance of current mobile networks. . The investigations are carried out in network simulator and by designing the SDN-based control for the LTE network. Results from this simulations are obtained, showing that this abstraction carries a negative impact on the performance of the network, related to certain network parameters. Conclusions to this results are given and some future lines of work to mitigate this effect and analyse it on different environments are suggested.

Acknowledgements

I would like to thank Dr. Toktam Mahmoodi for her support and guidance through all different stages of this project.

Table of contents

Nomenclature

CN	Core Network
eNB/eNodeB	Evolved Node B
EPC	Evolved Packet Core
ForCES	Forwarding and Control Element Separation
HSS	Home Subscriber Server
LTE	Long Term Evolution
MME	Mobility Management Entity
P-GW	Packet Gateway
PLR	Packet Loss Ratio
QoS	Quality of Service
RAN	Radio Access Network
RSRQ	Reference Signal Received Quality
SDN	Software Defined Networking
S-GW	Serving Gateway
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
Wi-Fi	Wireless Local Area Network
WiMAX	Worldwide Interoperability for Microwave Access
s	seconds
m	meter

List of Figures and Tables

1 Introduction

1.1 Overview

Internet traffic in all the network, and especially in wireless and cellular networks this days, is growing in a fast pace. Cisco [1] has declared that traffic in mobile networks will rise from 30 exabytes in 2014 to 290 exabytes in 2019. To deal with this, mobile operators are starting to deploy smaller and smaller cells of different wireless technologies such as Wi-Fi, which is making the network more dense and complex.

To be able to handle all this, and to lower management and expansion costs, a more flexible way to manage the network in the control plane is needed. In this point is where Software Designed Networking (SDN) makes an appearance, which will change how future networks will be designed and managed. The idea of abstracting the control layer, which in older network topologies was user-centric, and creating a new node, a controller, that manages all switches in the network and leaving the data plane on its own, will make the possibility to upgrade and adapt networks much easier.

Cellular networks are the ones mainly analysed in this report. The mobility of the users has a huge impact on the decision on decoupling control and data, since a very strict control on the routing and position of all users is needed for the network to work as intended, and because of that wireless networks are the ones that will be more affected by abstracting the control plane.

1.2 Aims and Objectives

The impact in the performance of the network of having a centralized control plane, decoupled from the data plane is the main focus of this report.

Being more specific, the aim of this report is to model the effect of having the data-plane part of the EPC in a cellular network just being the muscle of it, forwarding following the rules from its flow table, which is periodically updated from an external controller through the control layer of the network.

This update period is going to be a decisive parameter in the design of the network, checking how QoS parameters such as Packet Loss Ratio suffer if we decrease this update rate.

Once this effect has been modelled, the objective is to analyse the results in terms of QoS and discuss the optimal solution depending on the demands of the network, and discuss some ways to deal with this side effect.

1.3 Organization

The paper is organized as follows:

In Chapter 2 (Background), a reference on the concepts used during the system design and simulation is given.

Chapter 3 (System Design) details the network topology on which the study is done, detailing the different nodes in it and all the different important things to considerate towards analysing it.

Chapter 4 (Simulation), introduces to the environment used for simulating, gives a detailed view of the final implementation of the chosen topology of the point of study and shows the results of this simulations, along with comments on them.

Chapter 5 (Conclusions), concludes the report looking at what our objectives where and how we have contributed to analyse them. Also all difficulties and challenges that have been presented during this project are stated, and a couple of possible future lines of work are also noted.

The source code of the simulation in C++ language is listed at the end of the document as an appendix and attached to it in a CD.

2 Background

Before proceeding to show the simulation design and results section, all the concepts and technologies used in the proceedings have to be clearly stated and introduced, to make sure that the final idea is correctly understood.

This section covers this concepts and its purpose is to give the theoretical background needed for this project.

2.1 Long Term Evolution (LTE)

The Long Term Evolution (LTE) [2] in UMTS has been one of the latest advancements in our mobile networks technology. Instead of previous technologies, such as GSM and EDGE, LTE was designed thinking that all communications in the network would only be packet-switched, instead of the circuit-switched technology of the previous ones.

The main key points of LTE and that are its advantages respect older network architectures are hugely increased data transmission rates, more efficiency, a simplified architecture, having parts of the control abstracted in a separated layer, much better mobility management, and reduced delays and latency in transmission.

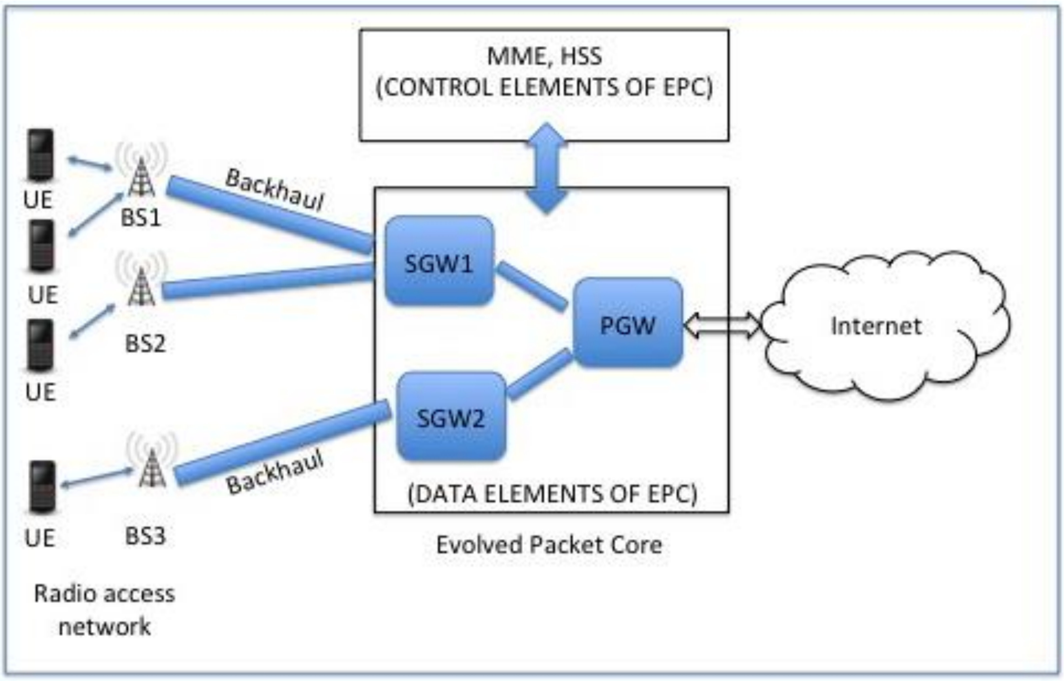

Figure 1 - Architecture of LTE networks [3]

Figure 1 shows a simple approach to the LTE architecture, which is constituted by the following elements:

- **User Equipment (UE):** The end node of the network, the one that starts the communications and asks for different types of applications and services. Can be any type of device that connects to the network using radio access.

- **Evolved Node B (eNodeB):** Also called base stations, this nodes are the link between the UE and the core of the network (CN), which in LTE is the Evolved Packet Core (EPC). They perform control measurements on the UE and keep track of their position and the quality of the received signal and are the ones who decide if a handover is needed. They also allocate the radio access network resources for each UE connected. The eNodeB sends traffic through the SGW over a tunnelling protocol called general packet radio service (GPRS) [4].

- **Serving Gateway (SGW):** When an UE moves between eNodeBs, this node works as a fixed point where all packets going to them are transferred through. It collects data such as the volume of bits sent and received for charging purposes. It also helps in the mobility process when the UE wants to change from a 4G network to an older one, such as EDGE or 3GPP, and permits communication when a user moves from one eNodeB to another one.

- **Packet gateway (PGW):** This node allocates all IP addresses to each UE connected to him through the eNodeBs. It also controls and forces QoS policies on the applications coming from the Internet, and acts as a firewall.

- **Mobility Management Entity (MME):** This node is set apart from the other ones, since it works in the management plane of the network, and processes all information and signalling between the UE and the EPC. Its main functions are managing the connections of the UE when it wants to access the network and during a handover, and some functions to be able to be able to take profit of older networks for some applications, such as voice calls.

- **Home Subscribing Server (HSS):** this part of the EPC contains all information relative to user subscriptions, such as the IMSI or the QoS enforcement needed by the user. It also takes part in generating vectors needed for authentication and security [2].

Following Figure 1, UE is connected to the eNodeB using the RAN (Radio Access Network). This eNodeB connect to the network core through the SGW, which tunnels all his traffic to a PGW, which is the last step between the backhaul and the Internet. Despite the LTE network has a management plane defined by the MME, all data nodes of the network (eNodeBs, SGW and PGW), also help on the control of the network and act in some management plane decisions [4].

2.2 Handover in LTE networks

One of the most critical aspects in mobile networks is the fact that the end nodes of the network, the UE, are subject to position changes, and since the RAN is very susceptible to distance and obstacles, a strong handover management is needed to have a stable and efficient network. Handover is the process in which one UE, because the received signal from its source eNodeB is not good enough to have a correct transmission, changes to another base station without losing connection.

In LTE, eNodeB manage handovers between them using the X2 interface, which is a virtual connection between all eNodeB that pertain to a certain EPC. We can resume the handover process of a UE in three steps:

- Decision: In the first one, the UE and the eNodeB exchange strength signal measurements, and the eNodeB decides if a handover is needed in this case.
- Execution: The source eNodeB asks the target eNodeB for allocation of resources for a handover. If it is accepted, and an ACK is received, then the source eNodeB notifies the UE of the decision and the user detaches from the source cell and joins the new one, and confirms to the new eNodeB that he is correctly connected. During this progress, the source eNodeB takes all buffered packets destined to the UE doing the handover and transfers them to the target eNodeB.
- Conclusion: The eNodeB then asks the PGW to switch the path the packets are being received, which asks the SGW to modify the user plane. If this is acknowledged, the packets are switched and routed to the target eNodeB. The target eNodeB asks the source one to release resources.

When deciding on a handover, the most common algorithms decide upon the results of certain events, related to the strength of the signals between the UE and the source eNodeB or its neighbours. In table X the different LTE measurement events are listed.

Event A1	Serving becomes better than threshold
Event A2	Serving becomes worse than threshold
Event A3	Neighbour becomes offset better than serving
Event A4	Neighbour becomes offset worse than threshold
Event A5	Serving becomes worse than threshold1 and neighbour becomes better than better than threshold2
Event B1	Inter RAT neighbour becomes better than threshold
Event B2	Serving becomes worse than threshold1 and inter RAT neighbour becomes better than threshold2

Table 1 – List of different LTE measurement events

Most handover algorithms use a combination of more than one of this events to make the protocol more efficient. The most typical handover decision used, and the one that is considered in this project, is the A2A4 RSRQ handover algorithm, which follows the following tree to decide whether or not trigger the handover procedure:

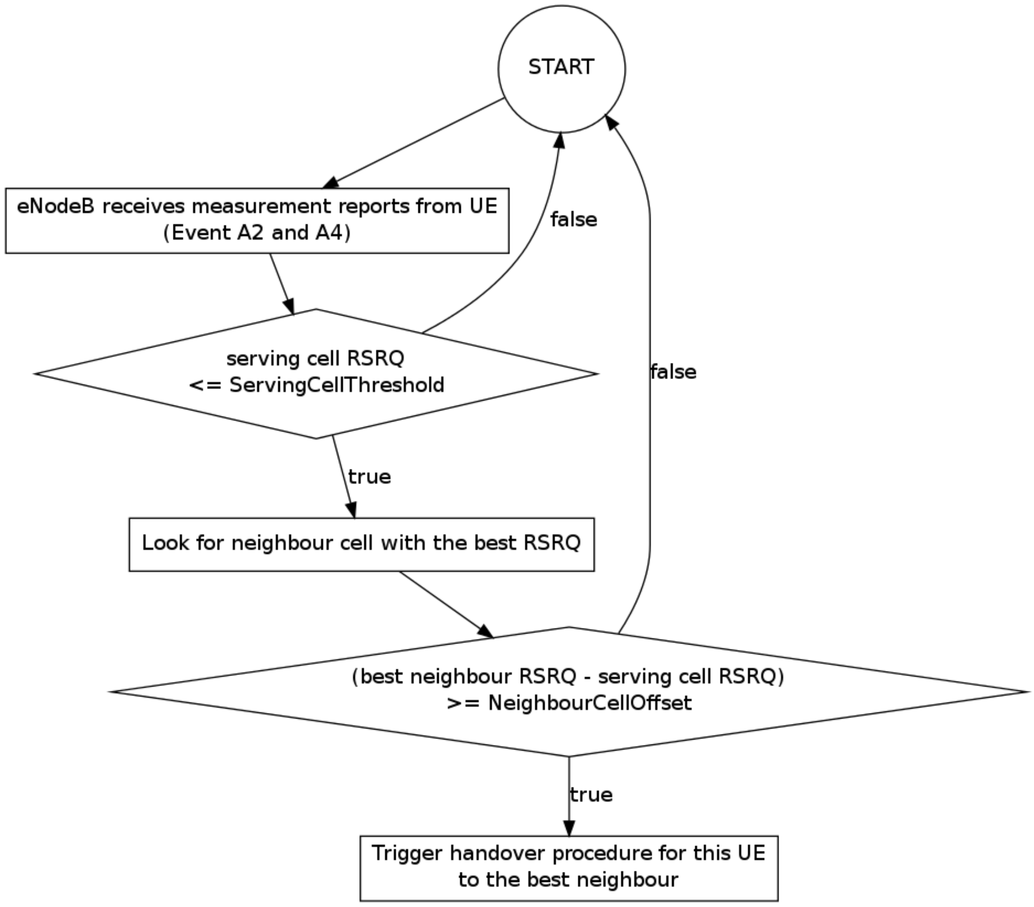

Figure 2 - A2A4 RSRQ Handover Algorithm [12]

Using this handover algorithm, the network makes sure that there are no unnecessary handovers by making sure that the source eNodeB is giving an enough bad coverage to have to consider it and at the same time that there is an eNodeB to which it would be worth the effort of doing the handover. RSRQ measurements both take into account the strength of the signal from the UE to the corresponding eNodeB and also the delays and throughput available in that eNodeB, due to how loaded is that cell in that particular moment.

2.3 Software-Defined Networking (SDN)

Standard state of the art computer networks nowadays are an incredible complex group of network nodes and devices like switches, routers or gateways. Managing this networks supposes an incredibly difficult challenge for any company or operator, because of its complexity and the limited access to the network they usually have. This difficulty comes from the fact that routers and other network devices are hardly modifiable, and have very few control interfaces.

Software Defined Networking (SDN) [5] is expected to be a radical change of how networks are controlled. It introduces a new networking idea in which the control plane is decoupled from the data plane of the network, releasing it from control decisions and just making it forward traffic. This control plane is centralized and installed in software-based controllers. Network devices can be programmed from this controllers using interfaces like ForCES [6] or OpenFlow [7]. This ability to program all devices in the network will grant users and operators a lot of network control, and will make networks much more flexible and updatable.

The OpenFlow protocol consists of a switch with one or more flow tables which communicates securely with an external controller through the OpenFlow Protocol [8]. Flow tables determine how every flow that enters the switch will be served. This tables contain rules that can be more complex than usual IP routers. They can have many different actions upon a flow arrival.

Looking at cellular networks, the next mobile generation, 5G, is being designed having SDN and similar technologies in mind, but SDN is also thought to be used in current 4G/LTE networks. In figure 3 is shown the LTE classical architecture with the addition of a SDN controller in the EPC part of the network:

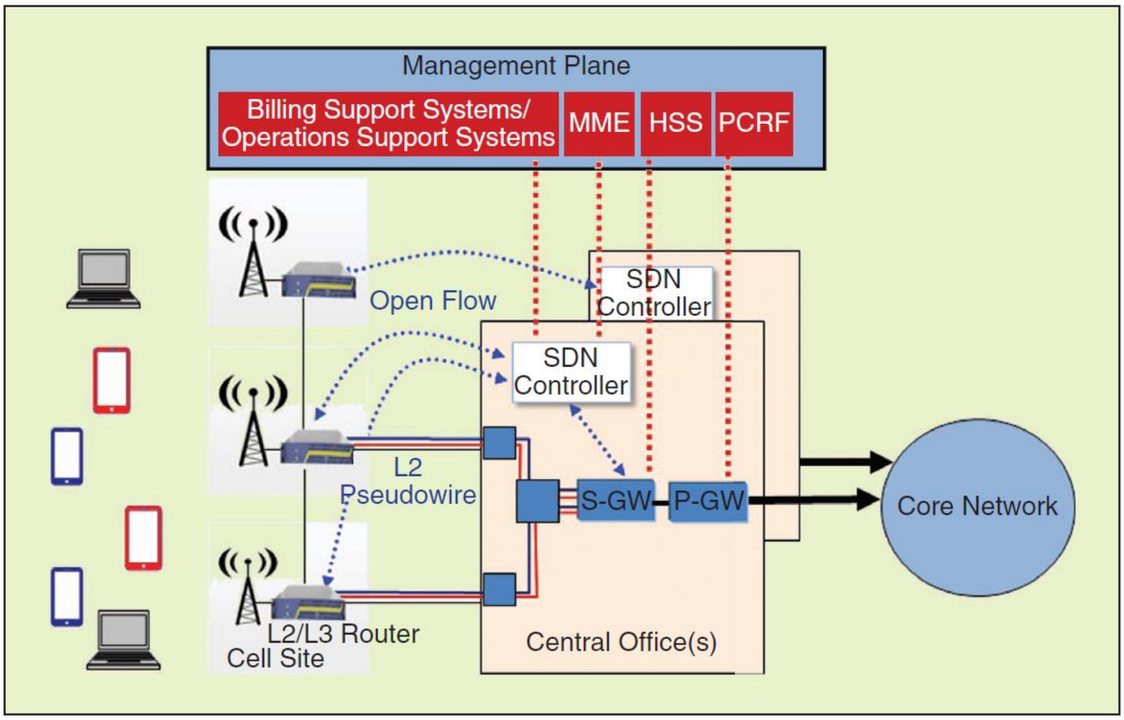

Figure 3 - LTE network using SDN Technology [4]

In normal LTE architecture, there was communication between the eNodeBs and the EPC, but now eNodeBs talk directly to the SDN controller, and the controller notifies the S-GW in case. This absence of talking between the eNodeBs and their respective S-GW creates a time gap between the eNodeBs decide a rule and the device which forwards packets to them, the S-GW, is aware of this rule.

In conventional LTE networks, S-GW store all information relative to the changes in user's positions, because even if they change eNodeBs, they maintain the same IP [9]. In the SDN scenario, all this information has to be held by the controller, which has to notify the S-GW of any user change. The controller also directs the eNodeBs to release resources once the cell change has been successfully stablished.

Li et al. expose in CellSDN [9] that applying SDN to current cellular networks would indeed give the network much more flexibility, make it easier to control and to expand, and give operators and users more control in an application level. It would make the mobility management much easier to deploy, informing several nodes at the same time of the modifications and push the new forwarding rules at the same time to all switches affected. After listing the benefits SDN will bring to cellular networks, they also define some extensions that could be given to the actual

SDN architecture to make it even more round, like installing what they call agents on local switches, which would be able to perform simple operations and release the controller from a lot of signalling, but still refer to the controller for new flows or unknown patterns.

Some good alternatives to the standard SDN structure have been exposed in technical papers, such as ProCEL [3], which argument that installing a switch between the eNodeBs and their S-GW, with its own controller, would release the controller of smaller decisions when the update time is really quick, because that would make the controller have to bear with an incredible amount of signalling information. This switches would be also directly connected to the internet and they would be able to discern all data that is received from their users. They could they decided depending on the type of traffic received. If this traffic is heavy QoS demanding, like voice calls or video it would still use all the LTE network and all its benefits of a stable mobility model, but if the traffic could handle small interruptions, it would be treated differently and forwarded directly through a wired connection to the internet, avoiding the mobile network and releasing it from a huge part of traffic.

Both this two alternatives would clearly help on mitigating the effect that is taken into consideration in this report.

3 System design

In the previous chapter, a little background on SDN and its application on current cellular networks is given. Particularly, figure 3 shows how the LTE architecture is modified to work using SDN protocols, and is exactly the topology that is going to be studied, simulated and analysed in this report.

In this scenario, each user is connected to the network through one base station using radio access. Then the eNodeB have a wired connection to the SGW on which they forward all data packets, and a direct connection to the SDN controller to which they forward all information relative to user position, signal measurements and control policies. The S-GW recaps all packets from the eNodeBs and connects them to the remote hosts through the P-GW. The forwarding rules that the S-GW follows are dictated by the controller. All this nodes have the functionalities described in the background section, corresponding to an LTE network.

The SDN controller is the main point of interest in this design. It abstracts the topology of the network, meaning he knows where every node of the network is located, and keeps updating it every certain period of time. To store this information, for fixed nodes, it simply knows the geographical location, which is preinstalled in it. For mobile users, it stores its relative position to well-known points, the eNodeBs, using graph theory.

It computes the distance between each eNodeB and the UE and stores it in a matrix. In each row of the matrix the controller stores the last known relative position of a UE to each of the eNodeB under his control. The controller also stores all information relative to which eNodeB each user is connected, and all parameters that the MME controlled.

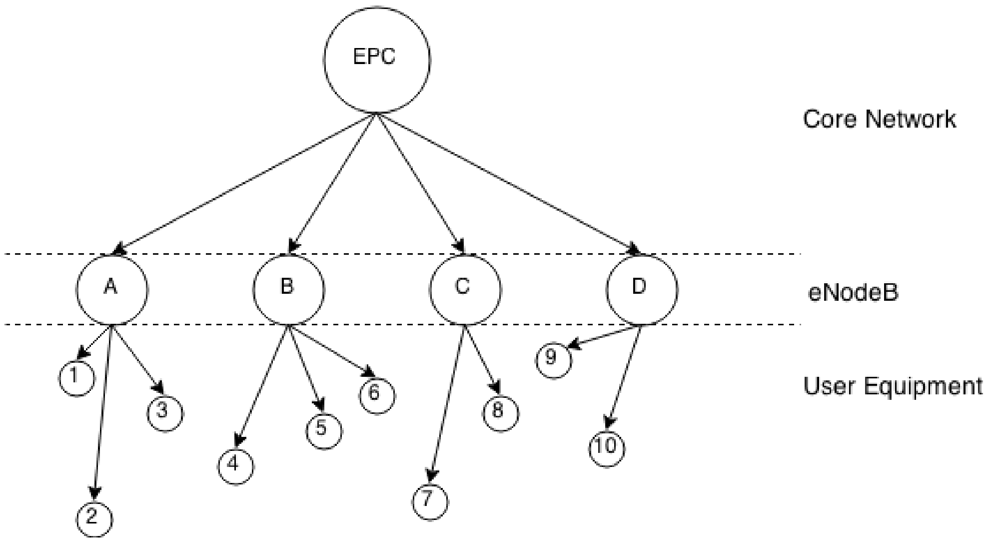

Figure 4 – Graph of the extracted network topology.

In figure 4 is shown the final simulated topology seen from the point of view of the controller. A bigger node implies that it is the parent of the following smaller one. The one on top represents the core of the mobile network, which has a constant position and is not relevant in the mobility scenario. The four middle ones are the eNodeBs of the network, which we need to know its geographical position to compare it against the small numbered nodes, which are the different UEs. The arrows that connect the nodes represent that there is an existing link between them, and in the eNodeB-UE link case, the link's distance is the parameter used in the weight comparison. In the weight matrix, the rows are determined by the eNodeB number, and the columns are defined by the UE number.

Using this weighting technique, the controller discerns different areas around the eNodeBs depending on the distance from the UE to its source eNodeB. There is a first area where it is geographically closer enough to don't even think about triggering a mobility procedure, and the controller just updates its position.

After that, a second area, where if a user enters it, the controller on each update asks the eNodeB for information relative to the signal quality and QoS parameters between the UE and the base station. If this information is below a certain threshold, the controller will check if the eNodeB has triggered a handover due to this results.

And finally, a third area where if the user is located there it will ask the eNodeB to manually trigger a handover procedure in the eNodeB to the one which is located closer to the actual UE position. In the graph shown before, when the controller is informed of a new handover or it triggers it, the link between the UE and the eNodeB is closed and a new one is created.

When the eNodeB that is procuring radio access to a particular UE decides to handover this user to a neighbour base station due to the results of some measurements, it sends a message to the controller through the secure channel letting he know of this handover.

Apart from mobility management, the controller also has to correctly update all flow routes in the core of the cellular network. If a handover has been triggered, the controller has to alert the EPC that it has to reroute all traffic to the convenient final eNodeB. Since this update is not real-time, some packets will be lost in the lapse between the handover is made and the switch has updated its flow table.

4 Implementation and results

4.1 Simulation Overview

The chosen software to run the simulations on is NS-3 [10], because of its importance and references in the research area and because of its many complete, complex and different models available to use. Version 3.21 of the simulator has been used to run the simulations, and then the NetAnim [11] and Flow Monitor models of NS-3 have been used to extract and analyse the results.

In order to simulate the effect of the delay in the update of the flow table in an SDN enabled switch in terms of packet loss ratio, the scheme displayed in the previous chapter was the first approach. A direct implementation of this idea was not possible in NS-3, because of the closed structure of the LTE model of NS-3, which is strictly tied to the SGW/PGW node that is automatically created, and all the tunnelling interfaces and protocols that are automatically created between the eNodeB and the SGW/PGW.

To show this point, in figures 5 and 6 we can see how the LTE model is defined in NS-3 and how the packets flow between the UE and the Internet.

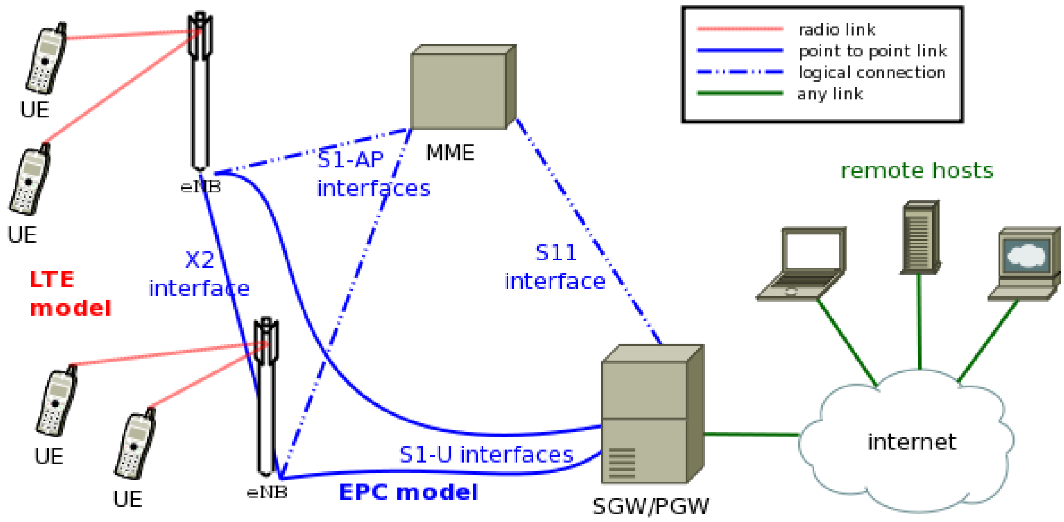

Figure 5 - LTE topology in NS-3 [12]

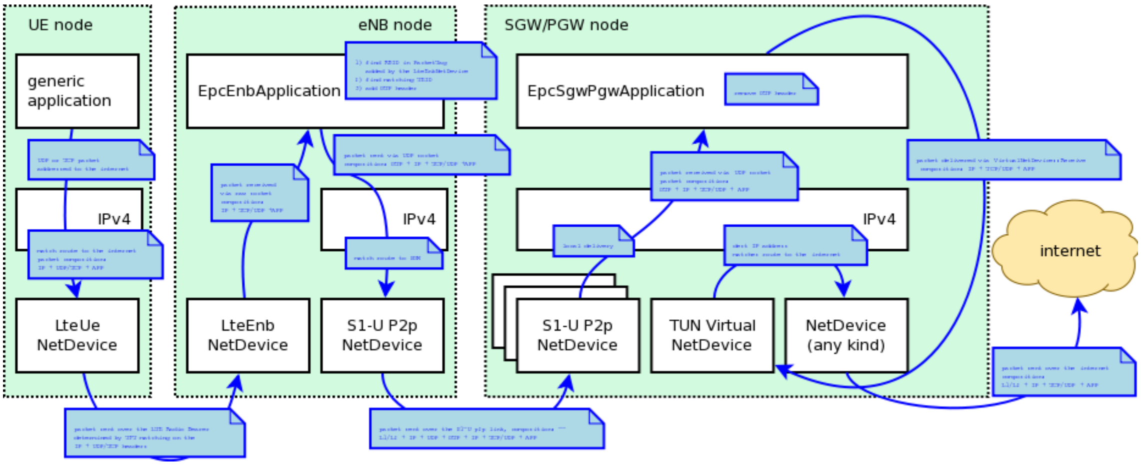

Figure 6 - Path packets follow between UE and the Internet in NS-3 models [12]

During the handover process in an LTE network, when the source and target eNodeB agree the exchange, the S-GW is alerted of this and reroutes all traffic from one eNodeB to another. In a case of abstracting this control and putting it in a server, the SDN enabled switch would keep sending packets through the port where the source eNodeB is, until the controller alerts of this handover and modifies the rules in the flow table.

In order to emulate the effect of having the control plane abstracted from the data layer, since we can't recreate it directly, the idea of port switching problem in the SDN enabled switch is moved to a handover problem.

The designed scenario is the following: each eNodeB has a first area where in normal situations you wouldn't trigger a handover to another station, even if it was geographically closer, because your RSRQ measurements are good enough to need a handover. After this area, following the RSRQ handover mechanism decision explained in the background, a handover would be triggered, but in this scenario, to emulate the desired effects, a big attenuation is inserted in the medium, which increases the packet loss ratio to almost 100%, like if the forwarding rule in the flow table of the SGW/PGW node had not been changed, which in this case simulates the effect of having an SDN controller.

After a certain period of time, the algorithm updates the position of all UE and eNodeB and triggers handover procedures to users who are losing packets because they are outside of the coverage area of their eNodeB and attaches them to the closest one which has coverage, and the user recovers the connection and stops losing packets.

The time that the UE is disconnected is exactly the same as if the user was connected in a centralized control topology but the switch wasn't aware of this new condition, so this simulation method will achieve the same results. In the following tree is detailed the procedure that the controller in our simulation bench follows to know the location of each user and if a handover is necessary:

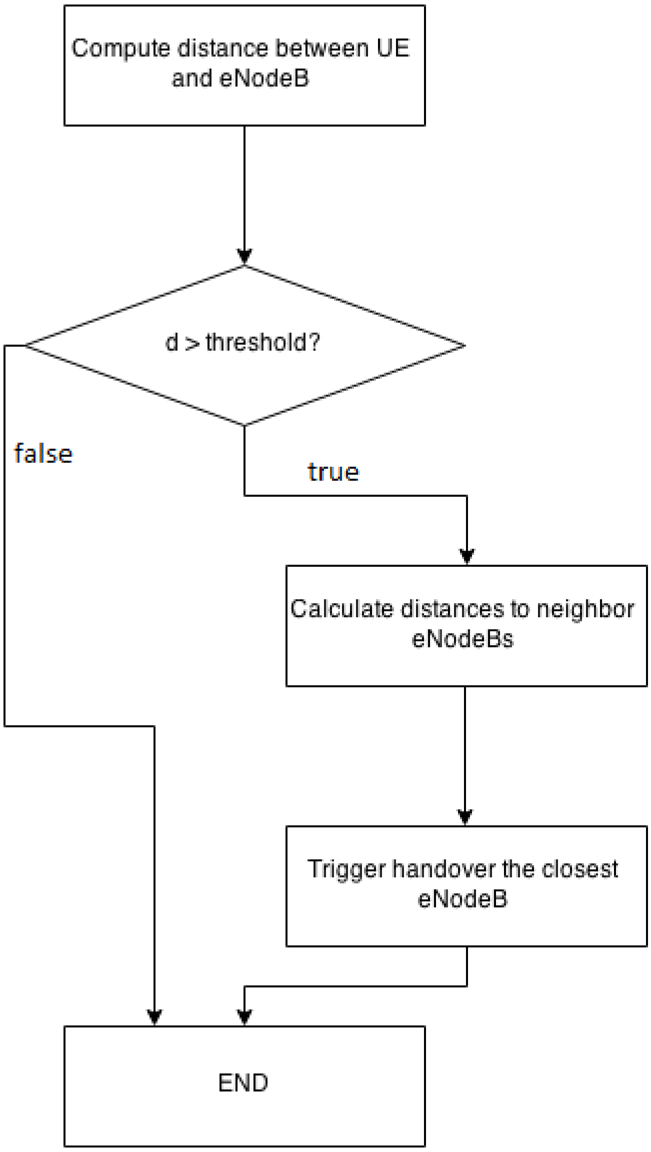

Figure 7 - Simulation handover algorithm flow chart

That mentioned period of time is going to be the main point of study, to determine how often would the flow tables need to be updated to maintain a correct functionality of the network but at the same time not overload it with a lot of SDN signalling.

4.2 Simulation Scenario Design

The proposed scenario to simulate the effect of varying this update time is a squared area with four eNB placed in a way that there is no “black area”, area where no eNodeB has coverage, to avoid them affecting the results. In this area, ten UE have been deployed, with a constant speed mobility model in order to emulate a real area and trigger the handovers. The connections between the UE and the eNodeB are automatically created by the LTE model in NS-3, and the propagation loss model of the RAN has been adjusted to emulate the effect of losing packets as explained before. To simplify the algorithm and the implementation, the controller only discerns within two areas depending on the distance of the UE to the source eNodeB, instead of the three areas described in the system design.

All four eNodeB are connected to the PGW/SGW node and this one is connected to a remote host who will work as the "Internet". All connections between the eNodeB, the PGW/SGW and the remote host nodes are defined using a simple point to point link model, which has a maximum capacity of 100 Gbit/s, and introduces an almost negligible delay to the transmission. With this assumption, we can assume that this links are completely stable, and will not affect the results of the simulations. Figure 8 draws the final chosen scenario for the simulations:

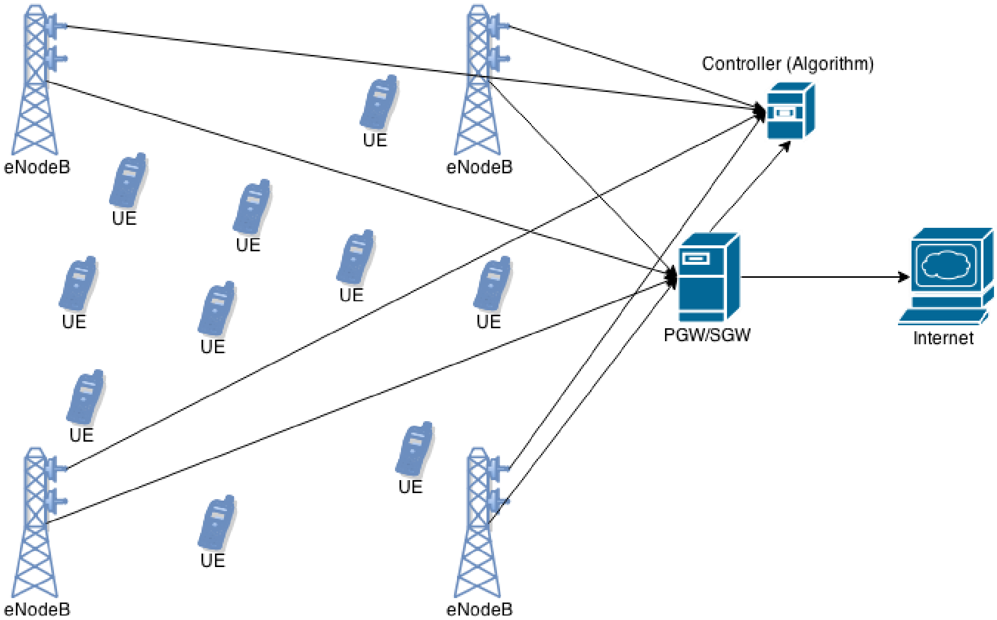

Figure 8 - Simulation network topology

In each of this UE an UDP client has been installed to send data to the remote host, and the remote host has the same application to answer this packets. Since to create a data flow an UDP protocol is used in the simulations, it can be stated that no retransmissions are being made of this packets, so the Packet Loss ratio can be computed using the following formula:

$$PLR\ (\%) = \frac{\#\ lost\ packets}{(\#\ received\ packets\ + \#\ lost\ packets)}$$

To give mobility to the UE to make it realistic, a mobility model has been installed in all of them which makes them walk in a certain direction until they reach one of the boundaries of the area, then it takes a random new direction and starts moving again. All users have a constant speed of 15 m/s, which simulates the average speed of a car moving in an urban environment. This speed was chosen because the effect on a low mobility node, like a pedestrian walking, of the pattern we want to model may not describe it correctly, since a user may not even handover during the entire simulation time if it just walks, and it would shadow the final results.

In telecommunications, when designing a topology for a wireless network, the design must be tested for a different variety of conditions, since the number of users, their position and the application they use can vary creating so many different environments. In this case, Monte Carlo method [13] is used to replicate the same environment for different values of this variables to be able to evaluate the topology in a reliable way. Monte Carlo methods when used in simulations that contain random variables helps reducing the error by a factor of $1/\sqrt{N}$ [13], where N is the number of independent simulations run on the same topology.

Since both initial placement and direction are decided using random parameters, with only one simulation the results may not show a correct effect of what we are trying to see, so to be able to rely on them several simulations have to be made, and then take an average of this simulations to achieve a proper result. In this study, 20 independent simulations have been carried away, for each update time to be analysed.

It is noted that the handover procedure in NS-3 works as the normal LTE handover once you ask for a handover, with the only difference that if the source eNodeB had packets queued to transmit to the UE and you trigger a handover, the packets are discarded since the connection between the source eNodeB and the UE is closed, so this effect may alter the final results in a minimum way. The X2 interface is needed to have handover capabilities, since the model uses them to transmit handover data, but doesn't reroute queued packets.

4.3 Simulation Results

In the following table and graph are displayed the results of the simulations in the topology detailed before, for each of the ten UE deployed in the simulation area. Each of this numbers are the average of 20 independent simulations:

Update Time	UE 1	UE 2	UE 3	UE 4	UE 5	UE 6	UE 7	UE 8	UE 9	UE 10
5 seconds	10,46	10,33	10,41	11,51	10,37	11,10	10,40	11,28	10,08	11,98
10 seconds	11,58	11,63	11,61	12,42	12,08	12,80	11,77	12,24	11,60	13,67
30 seconds	18,26	18,00	17,50	19,06	17,90	19,62	19,14	19,36	17,78	20,45
60 seconds	23,86	24,61	24,76	22,55	25,15	26,91	26,51	24,38	23,91	26,10

Table 2 – Simulation results per UE

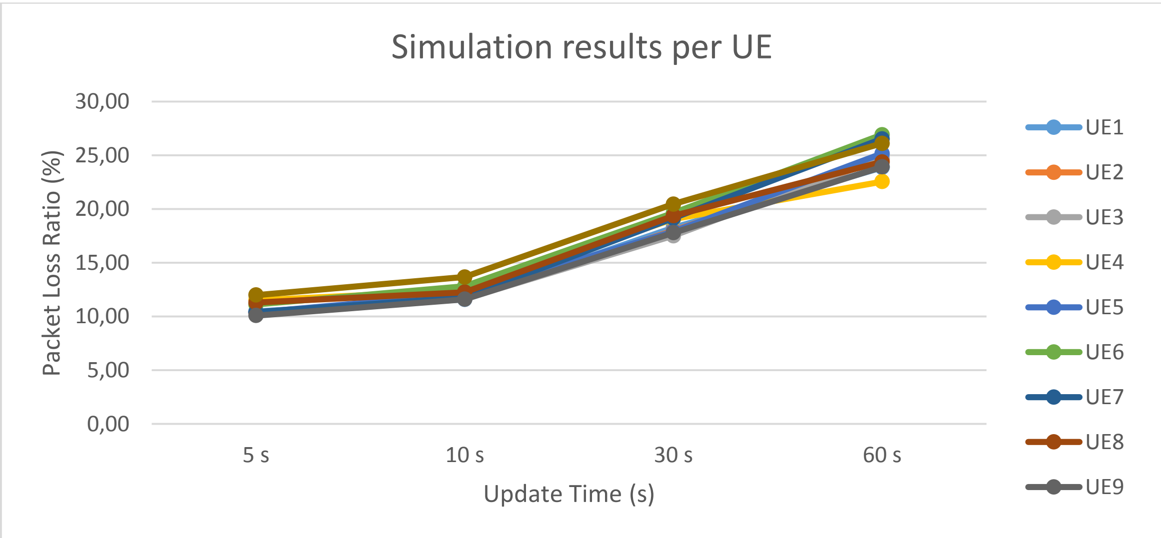

Figure 9- Simulation Results per UE

The results show what was predicted at the beginning, that the increase in the update time would make the packet loss ratio increase. All UE follow a very similar pattern, and since they are all randomly positioned and their direction is also randomly decided, it means that this numbers have some pattern behind.

The little deviations between UE correspond to the fact that every user follows a completely different trajectory, and this changes, between many other things, the number of times that a user needs to trigger a handover because he is out of his eNodeB coverage area. If one of this individual users is given a random trajectory which makes him stay around his eNodeB for a longer time, he will lose less packets than one that is constantly leaving the coverage area and having to trigger a base station change.

To analyse this pattern more clearly, in table 3 and figure 10 are shown the results taking an average of all users simulated:

Update Time	Packet Loss Ratio (%) (UE average)
5 s	10,79
10 s	12,14
30 s	18,71
60 s	24,87

Table 3 – Simulation results in average

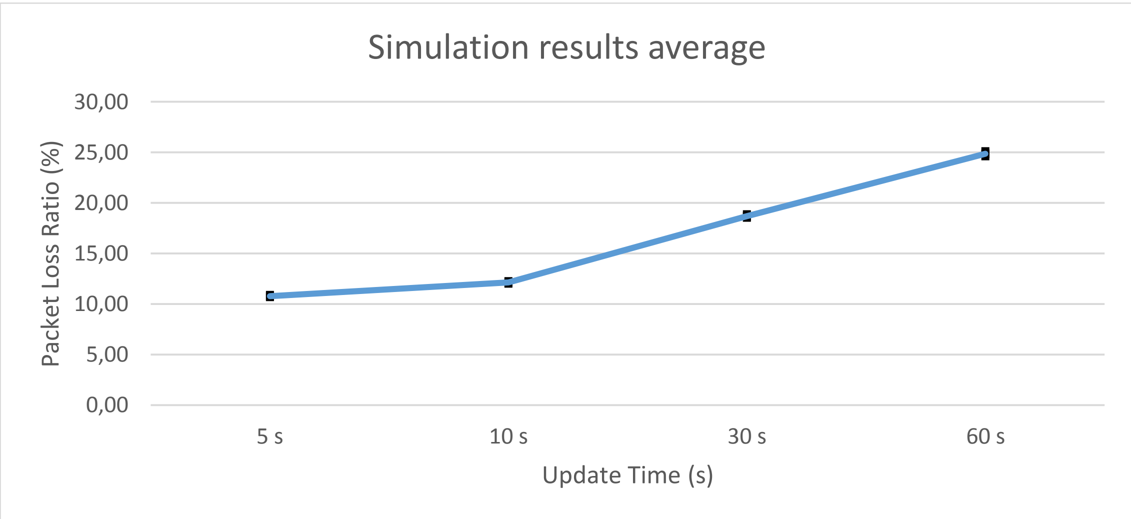

Figure 10 – Simulation results in average with 95% confidence interval bars

The average follows a clear incremental pattern when the update time is increased. Looking closely at the numerical values extracted from the table:

- When the update time is changed from five seconds to ten seconds, the increase in the packet loss ratio is 1.35%.
- In the next step, changing from ten seconds to thirty seconds, the increase is 6.57%, and since the time increase in this step is four times bigger, the relative increase if we did the same step size as in the first one is 1.64%.
- In the last step, from thirty seconds to sixty seconds, the increase is 6.16%, and the relative increase in a 5 seconds hop is 1.02%

From this analysis can be extracted that the increase in the packet loss ratio doesn't follow a linear pattern, and that there is an area where increasing the update time has a more detrimental effect on the QoS parameter than in others. In fact, it is clear that in the interval between ten and thirty seconds it is important to decide whether or not to increase the update time, while between thirty and sixty seconds the decrease in the packet loss ratio is almost 50% as it was in the previous section.

Strictly talking about QoS maximization, the better approach would be to reduce the update time to make it almost real time, but it is obvious that this solution can't be optimal because of the huge amount of load that it would make. To be able to find a reasonable optimal update time point, all parts that are affected by the update time have to be taken in consideration. In figure 11 a simple chart has been drawn to represent the number of updates that the flow table needed during the simulation time:

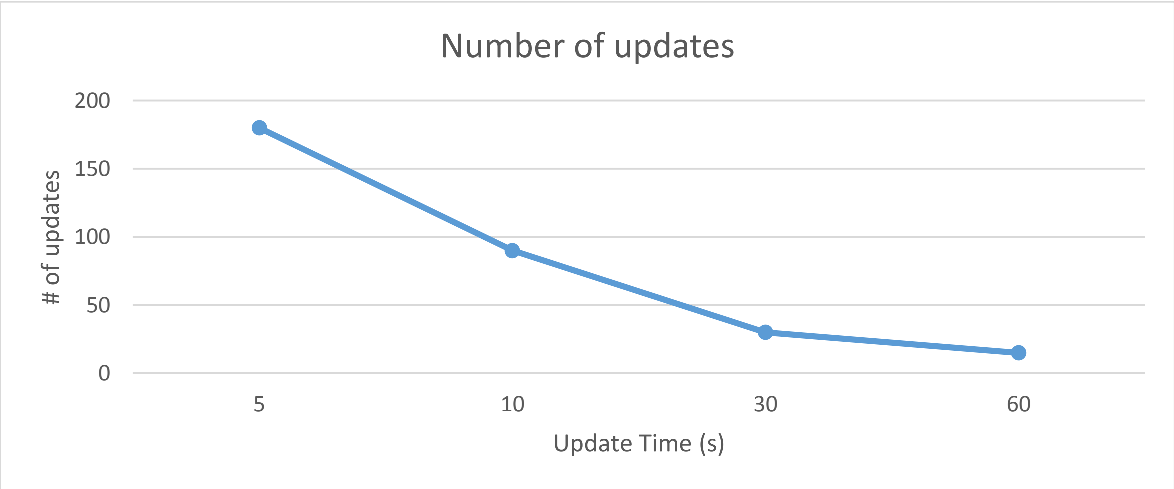

Figure 11- Number of updates in a 15 minute simulation

From this last picture it can be extracted that the quicker the updates are scheduled, the more updates we have to do. This is quite obvious, but it is important to mark since for each update, an amount of traffic between the controller and all the switches and eNodeB has to be made to know the actual status of the network.

And in this simulation case this effect would be negligible since we are dealing with only a few users and base stations, but in a gigantic cellular network, all this data flow that appears when we want to update the network topology on the controller has to be taken into account as some way of traffic overhead. Looking at figure 11, this effect increases exponentially as we reduce the update time, so this effect should be considered especially when a low update time is desired.

Another interesting point to consider in order to make a decision of what would be a reasonable update time is the type of application that is used in the network.

If the type of application can hold a reasonable packet loss ratio, like video on demand, then a larger update time can be used without affecting the overall performance of this applications and at the same time having very little overhead. Instead, for applications that are interactive, like web browsing, gaming or even more background applications like electronic mailing or IM, packet loss ratio is a very crucial QoS parameter, and then it has to be kept to a minimum, and in our study case, a very low update time would be crucial to deliver a good service.

Taking into account all this different points made, we can extract some conclusions:

- In networks where the packet loss ratio is not a crucial factor, where the dominant applications used are the ones not extremely affected by this loss, a longer update time is the best choice, since this way we reduce the SDN signalling without affecting the network, and the controller would not have such a huge load of work.
- If the network to manage is, instead, very susceptible to packet losses, and cannot depend on retransmissions, a quicker update rate may be the optimal choice at first glance, but overloading the network with a lot of signalling between the controller and all the middle nodes may affect it in an undesirable way, so a trade-off between QoS and signalling load has to be made.

The optimal proposed scenario is a variable update time in the margin between ten and thirty seconds, because the gain obtained from reducing the update rate lower than ten seconds may not be worth since it increases massively the controller-nodes traffic for the low PLR reduction, and making it update longer than 30 seconds would just increase the traffic loss while not incrementing the overall control overhead of the network. Only the case of having a network so huge that the controller could have problems dealing with would be a good scenario to lower it even more.

5 Conclusion

In this report, the effect of extracting the network topology and decoupling the control plane by using Software-defined networking on the EPC of current mobile networks is analysed. After testing the designed model in different independent scenarios, the negative effects of this centralised control are defined and a possible solution to counter them is suggested as a trade-off between the benefits and the side effects of this model, optimised depending on the type of network to control or the demands of the network.

To this end, it can be stated that SDN is going to be a huge leap forward in future network design. However, this concept is not perfect and some modifications and adjustments need to be done to ensure a good performance. The idea of installing an agent on each middle point in the network would greatly reduce the side effect of centralizing the control shown in this document, since it would be able to store some extra information relative to the position of the users, and be able to react to certain events without the precise order of its superior control point.

As a future line of work, two main points arise that could potentially lead to interesting results:

- Since the controller knows the network topology and in each update it refreshes the weighted graph of the network, it could be possible to implement a predicting algorithm on the controller that could foresee the future position of a UE based on his trajectory, and notify the switches that a handover is going to be triggered in-between updates so it updates the rule accordingly, could help reduce the packet loss ratio, especially when the update time is quite high.
- Analysing this same effect described in this report but not only focusing on LTE networks and taking into account heterogeneous environments, such as cellular networks that co-work with wireless networks like Wi-Fi or WiMAX, and see how the controller responds to both handovers in the cellular networks and handovers between the LTE network and one of this wireless networks. Some early research on how to implement SDN in heterogeneous networks has been already done on [14], so the theoretical aspects of implementing SDN in this type of networks have already been discussed, but its performance still has to be tested.

Overall, this project has been successful and the research community may benefit from this results.

6 References

[1] Cisco Virtual Networking Index: Global mobile IP traffic forecast update, 2014 to 2019. February 2015. http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html

[2] S. Sesia, M. Baker and I. Toufik*, LTE – the UMTS Long Term Evolution: From Theory to Practice*, John Wiley & Sons, 2011.

[3] K. Nagaraj, S. Katti, “ProCel: Smart Traffic Handling for a Scalable Software EPC”. HotSDN, August 2014

[4] T. Mahmoodi, S. Seetharaman, “Traffic Jam: Handling the Increasing Volume of Mobile Data Traffic,”IEEE Veh. Technol. Mag., Sept. 2014.

[5] ONF, “Software-Defined Networking: The New Norm for Networks”, white paper. April 2012

[6] A. Doria, J. Hadi Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern. Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810 (Proposed Standard), March 2010.

[7] ONF, “OpenFlow™-Enabled Mobile and Wireless Networks”, white paper. September 2013

[8] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, “A Survey of Software-Defined Networking: Past, Present and Future of Programmable Networks”, IEEE Commun. Surv. & Tut. , Vol.16, No.3, 2014

[9] L. E. Li, Z. M. Mao and J. Rexford, “CellSDN: Software-Defined Cellular Networks”, Univ. Princeton, TR-922-12, April 2012.

[10] NS-3 Network Simulator, available online: http://www.nsnam.org/, last update March 2015

[11] NetAnim offline animator, available online: https://www.nsnam.org/wiki/NetAnim, last update March 2015.

[12] NS-3 LTE Module Design, available online: http://www.nsnam.org/docs/release/3.22/models/html/lte-design.html, last update March 2015.

[13] Monte Carlo Method, Wikipedia: , http://en.wikipedia.org/wiki/Monte_Carlo_method, last update February 2015.

[14] B. A. A. Nunes, M. Mendonca, K. Obraczka, and T. Turletti, “Software Defined Networking for Heterogeneous Networks”, IEEE Comsoc MMTC E-Lett., Vol.8, No.3, March 2013.

7 Appendices

7.1 Simulation Source Code

The source code used to simulate the network environment is provided in the CD included with this document. I verify that I am the sole author of the programs contained in the CD.