MASTER THESIS

SEAMLESS DISTRIBUTED MOBILITY MANAGEMENT (DMM) SOLUTION IN CLOUD BASED LTE SYSTEMS

Luca Valtulina

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Design and Analysis of Communication Systems (DACS)

SUPERVISORS
Dr.ir. Georgios Karagiannis
Morteza Karimzadeh, M.Sc.
Dr.ir. Geert Heijenk

UNIVERSITY OF TWENTE.

November 2013

Abstract

Distributed Mobility Management (DMM) refers to a set of solutions developed to overcome the limitations posed by currently deployed centralized mobility management schemes. DMM offers to operators a more efficient network deployment driven by a distributed placement of core network entities close to the edge (access) of the network.

The huge appreciation received by Cloud Computing technologies in latest years pushed mobile network operators to plan the adoption of virtualization in their future network aiming at the possibility to share a common infrastructure among them. Innovative DMM solutions can exploit the natural distribution of users and mobility information within these shared virtualized networks envisioning the support of cross-operator mobility management.

This research focuses on the design, implementation and evaluation of three DMM solutions in cloud based LTE systems. A novel architecture has been defined to support DMM by means of redirecting the traffic to the relocated mobility anchor point in order to allow IP address continuity to flows kept active by the UEs upon movement. Traffic redirection occurs in the transport network above the Evolved Packet System (EPS) via encapsulation-free forwarding schemes based on two different technologies: NAT and OpenFlow.

Whereas all proposed solutions have been proven to offer seamless mobility management to UEs which handovers trigger the relocation of their distributed mobility anchor points, the results confirmed the inborn versatility and efficiency of the OpenFlow protocol.

Furthermore, being migration of virtualized entities and functions a prominent feature of cloud based networks, DMM can be also used to support traffic redirection when a virtualized P-GW entity, running on a virtualization platform, is migrated to another virtualization platform and the ongoing sessions supported by this P-GW need to be maintained. Indeed, solutions like mobility prediction systems have been proven to be required to provide seamless mobility to UEs moving to a location where their virtualized mobility anchor has to be migrated.

Acknowledgments

This Master Thesis marks the end of my two years experience at the University of Twente, an inspiring place to learn as a student and grow as a person.

First, I would like to express my gratitude to my supervisor Dr. ir. G. Karagiannis for the useful comments, remarks and encouragement through the entire duration of this research. Besides my supervisor, I would like to thank the rest of my thesis committee: Dr. ir. G. Heijenk and M. Karimzadeh for their insightful comments and support during the thesis development. The value and contribution of this thesis have increased a lot with their comments.

I cannot avoid to thank my partner in crime, Triadimas Arief Satria, for the endless hours of discussions, brainstorming sessions, coding and debugging during these past 9 months. It is always easier to share stress, anger, happiness and success with a person which can understand you and support you being himself in the exact same state of mind.

I thank my past and current employer and colleagues for their immense support, availability and flexibility during the past two years.

Last but definitely not the least I would like to thank my loved ones. My girlfriend for being always at my side and "encouraging me" through the entire duration of my Master program. My family who, despite the distance, helped me overcome all the difficulties of a life abroad. My friends, both in The Netherlands and in Italy just for being there, always. And Google for helping me write these acknowledgements.

Luca Valtulina

Contents

List of Abbreviations

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
AODV	Ad hoc On-demand Distance Vector
API	Application Programming Interface
APN	Access Point Name
BBU	Base Band Unit
CC	Cloud Controller
CDF	Cumulative Distribution Function
CDN	Content Delivery Network
CN	Correspondent Node
	Core Network
CQI	Channel Quality Indication
CSG	Closed Subscriber Group
CSMA	Carrier Sense Multiple Access
CTTC	Centre Tecnològic Telecomunicacions Catalunya
DL	DownLink
DMM	Distributed Mobility Management
E-RAB	E-UTRAN Radio Access Bearer
E-UTRA	Evolved Universal Terrestrial Radio Access
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
EAP	Extensible Authentication Protocol
EARFCN	E-UTRA Absolute. Radio Frequency Channel Number
ECM	EPS Connection Management
EEU	Enterprise End User
eGTP	Evolved GPRS Tunneling Protocol
EPC	Evolved Packet Core
EPCaaS	Evolved Packet Core as a Service
ePDG	Evolved Packet Data Gateway
EPS	Evolved Packet System

FE	Functional Entity
FPGA	Field-Programmable Gate Array
FTP	File Transfer Protocol
GERAN	GSM EDGE Radio Access Network
GPRS	General Packet Radio Service
GRE	Generic Routing Encapsulation
GSM	Global System for Mobile Communications
GTP	GPRS Tunneling Protocol
HA	Home Agent
HeNB	Home eNodeB
HMIP	Hierchical Mobile IP
HSS	Home Subscriber System
HTTP	Hypertext Transfer Protocol
IE	Individual End-user
	Information Element
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IP-CAN	IP Connectivity Access Network
IPsec	Internet Protocol Security
ISP	Internet Service Provider
ISR	Idle state Signalling Reduction
L-GW	Local Gateway
L3	Layer3
LA	Legacy Agent
	Location Area
LIMONET	LIPA Mobility and SIPTO at the Local Network
LIPA	Local IP Access
LMA	Local Mobility Anchor
LTE	Long Term Evolution
MaaS	Monitoring as a Service
MAC	Medium Access Control
MAG	Mobility Anchor Gateway
MCN	Mobile Cloud Networking
MIPv6	Mobile IPv6
MME	Mobility Management Entity

MN	Mobile Node
MOBaaS	Mobility and Bandwidth Availability Prediction as a Service
MPLS	MultiProtocol Label Switching
NAS	Non Access Stratum
NAT	Network Address Translation
NIC	Network Interface Controller
NMS	Network Management System
OLSR	Optimized Link State Routing
P-GW	Packet Data Network Gateway
PBA	Proxy Binding Acknowledgement
PBU	Proxy Binding Update
PCC	Policy and Charging Control
PCEF	Policy and Charging Enforcement Function
PCRF	Policy and Charging Rules Function
PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
PDU	Packet Data Unit
PHY	PHYsical layer
PLR	Packet Loss Ratio
PMIPv6	Proxy Mobile IPv6
PoP	Point of Presence
PPBP	Poisson Pareto Burst Process
QoS	Quality of Service
RAN	Radio Access Network
RANaaS	Radio Access Network as a Service
RAT	Radio Access Technology
RED	Random Early Detection
RLC	Radio Link Control
RRC	Radio Resource Control
RRM	Radio Resource Management
RTT	Round-Trip Time
S-GW	Serving Gateway
SAE	System Architecture Evolution
SDF	Service Data Flow
SIC	Service Instance Component
SIPTO	Selected IP Traffic Offload
SLA	Service Level Agreement
SO	Service Orchestrator

TA	Tracking Area
TCAM	Ternary Content Addressable Memory
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TEID	Tunnel Endpoint IDentifier
TFT	Traffic Flow Template
TLS	Transport Layer Security
TTL	Time To Live
UDP	User Datagram Protocol
UE	User Equipment
UL	UpLink
UMTS	Universal Mobile Telecommunications System
VLAN	Virtual Local Area Network
VM	Virtual Machine
VoIP	Voice over IP
VP	Virtualization Platform
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

List of Figures

List of Tables

Chapter 1

Introduction

In a situation where the number of mobile subscribers accessing wireless networks does not stop increasing, operators are facing a challenge in tackling the huge demand of mobile data services from users. It has been estimated that in dense populated areas the coverage of access technologies such as 3G and WLAN reaches almost 100%. This together with the affordable price of both mobile devices and subscription contracts resulted in users demanding for Internet connectivity everywhere.

Consequentially operators decided to migrate their networks to full IP based networks as the most recent mobile architectures like IEEE 802.16 suite (also known as WiMAX) [8] and the 3GPP Evolved Packet System (EPS) [9], for both voice and data, triggering a real need for a mobility management solution which provides IP address continuity to flows kept active by users upon movement. Most of the current IP mobility solutions standardized by both IETF (Mobile IPv6 [10] and Proxy Mobile IPv6 [11]) and 3GPP (GPRS Tunnel Protocol [12]) rely on a centralized mobility anchor entity which is in charge of both mobility-related control plane and user data forwarding. The presence of this centralized network node makes mobility management prone to several problems and limitations as identified in [13, 14, 15]: suboptimal routing, low scalability, signaling overhead, more complex network deployment, security issues due to the existence of a potential single point of failure, and a lack of granularity on the mobility management service.

In order to address the aforementioned issues, a new paradigm of solution, the so-called *Distributed Mobility Management (DMM)*, is currently being analyzed by both academic and standards communities (IETF and 3GPP). DMM basically develops the concept of a flatter system, in which the mobility anchors are placed closer to the user, distributing the control and data infrastructures among the entities located at the edge (access) of the network.

Furthermore, EU FP7 projects, like the EU FP7 MCN project [16] integrates the use of the Cloud Computing concept in LTE mobile networks in order to increase LTE's performance. This is accomplished by building a shared distributed LTE mobile network that can optimize the utilization of virtualized computing, storage and network resources and minimize communication delays. The use

of DMM can be applied in such environments not only to enhance the LTE mobility management performance and provide session continuity to users across personal, local, and wide area networks without interruption, but also to support traffic redirection when a virtualized LTE entity, like the P-GW, running on a virtualization platform is migrated to another virtualization platform and ongoing sessions supported by this P-GW need to be maintained.

1.1 Background

1.1.1 Towards a cloudified LTE system

Nowadays the Cloud Computing model is used to maximize the utilization of large data storage systems, resulting in clustering and pooling of its resources and the use of virtual machines. The Cloud Computing model could be applied in mobile cellular systems offering decentralized computing, smart storage, on-demand, elastic and pay-as-you-go services to third party operators and users. The process of applying the Cloud Computing concept in networks can be denoted as network cloudification [17] and it consists in the entire (or partial) virtualization of the network elements (hardware and software). When the intended network is a mobile network (e.g. LTE or WiMAX) this process is named Mobile Cloud Networking.

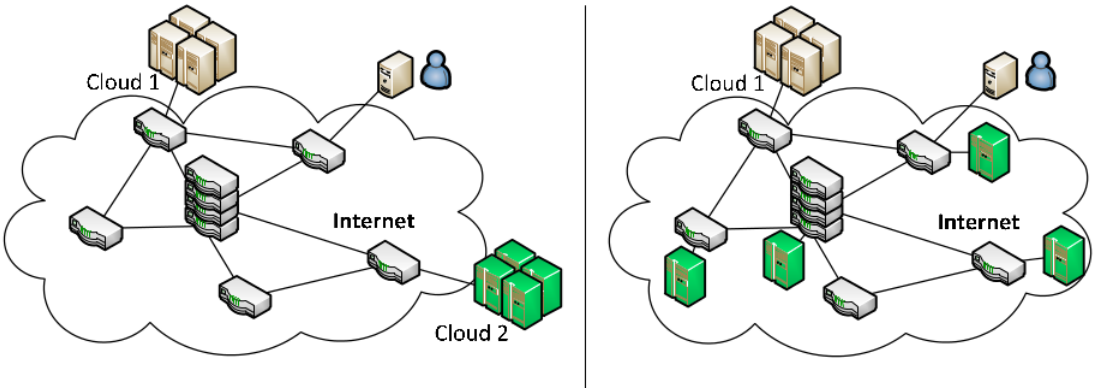


Figure 1.1: Cloud resources spread into the network, copied from [1]

A solution that distributes the cloud (and its main benefits) more deeply into the network and disperses the cloud closer to the end user to reduce latency is needed. Figure 1.1 is copied from [1] and shows how the resources of Cloud 2 are spread much finer and deeper into the network, close to the actual point of usage. An EU FP7 Large-scale Integrating Project named Mobile Cloud Networking (MCN) [16] has been founded in November 2012 with the goal to investigate, implement and evaluate the extension of the Cloud Computing concept beyond the core network data centers towards the mobile end-users. This research has been performed within the context of the MCN project.

Virtualization

Virtualization is a well known technique that has existed for years, especially in the computer world, like the use of virtual memory and virtual operating systems. The new idea is to use virtualization techniques to define a Mobile Cloud Networking solution.

With network virtualization multiple virtual networks, each operating similar to a normal network and unaware of the underlying virtualization process, are enabled to coexist on a common infrastructure. Individual virtual networks can contain totally different protocols and architectures from other co-existing virtual networks, due to the fact that they can belong to different operators.

Besides the possibility to share the same physical infrastructure, network virtualization allows a more efficient usage of the available resources by allowing operators to have networks coexisting in a dynamic and flexible manner. This implies that the physical infrastructure needs to be virtualized into a number of virtual resources being offered to the different virtual networks. The process involves applying the current operating system virtualization experience for network components like routers, links and base stations.

But the virtualization process involves also the wireless part of the network. Virtualization of wireless resources is a complex challenge. First, the wireless resources at the base station have to be fairly shared and assigned to different virtual network operators. Fairness in wireless systems can have different meaning: fairness in terms of spectrum used, or power used, or a product of these two or even fairness of Quality of Service (QoS) delivered to end users. Furthermore, it is not sufficient to look at the resources that are being shared, assigned or scheduled at one base station, but also the interference caused by the utilization of these resources need to be considered as well.

Network virtualization will allow completely new value chains. Smaller players can come into the market and provide new services to their customers using a virtual network. This also allows completely new future networks, e.g. isolating one virtual network (e.g. banking network) from a best effort Internet access network. In addition to all of the previous, the idea of being able to share the network resources among multiple operators is very appealing. This gives operators the flexibility to expand/shrink their networks and the air interface resources they use, and will lead to better overall resource utilization and reduced energy consumption.

Long Term Evolution (LTE)

LTE [9] as defined by the 3rd Generation Partnership Project (3GPP) is a highly flexible radio interface. In contrast to the circuit-switched model of previous cellular systems, LTE has been designed to support only packet-switched services. It aims to provide seamless Internet Protocol (IP) connectivity between User Equipment (UE) and the Packet Data Network (PDN), without any disruption to the end users' applications during mobility.

Often the term "LTE" refers to the evolution of the Universal Mobile Telecommunications System

(UMTS) radio access through the Evolved UTRAN (E-UTRAN) while the evolution of the non-radio aspects is known as System Architecture Evolution (SAE), which includes the Evolved Packet Core (EPC) network. Together LTE and SAE comprise the Evolved Packet System (EPS).

EPS uses the concept of EPS bearers [18] to route IP traffic from a gateway in the PDN to the UE. A bearer is an IP packet flow with a defined Quality of Service (QoS) between the gateway and the UE. The E-UTRAN and EPC together set up and release bearers as required by applications.

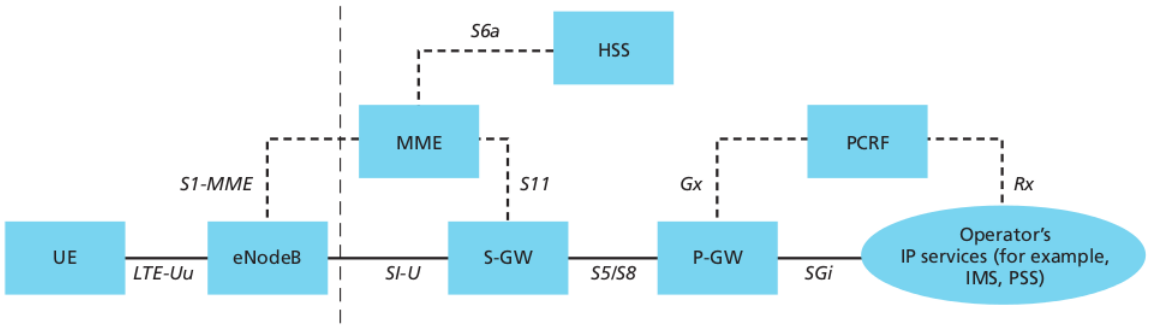


Figure 1.2: The EPS network elements, copied from [2]

Figure 1.2 shows the overall network architecture of LTE Release 8 (2010) [19], including the network elements and the standardized interfaces. At a high level, the network is comprised of the core network (EPC) and the access network (E-UTRAN).

The EPC is composed of five network elements: the S-GW (Serving GW), the P-GW (Packet Data Network Gateway), the MME (Mobility Management Entity), PCRF (Policy and Charging Rules Function) and the HSS (Home Subscriber System). According to [17] and [2], the functions of those elements are as follows:

PCRF – The Policy and Charging Rules Function is responsible for policy control decision-making, as well as for controlling the flow-based charging functionalities in the Policy Control Enforcement Function (PCEF), which resides in the P-GW. The PCRF provides the QoS authorization (QoS class identifier and bit rates) that decides how a certain data flow will be treated in the PCEF and ensures that this is in accordance with the user's subscription profile.

HSS – The Home Subscriber System is a database that contains user-related and subscriber-related information. It also provides support functions for mobility management, call and session setup, user authentication and access authorization.

P-GW – The Packet Data Network Gateway connects the EPC to other external networks. The P-GW routes packets to and from the PDNs. The P-GW also performs various functions such as IP address / IP prefix allocation or policy control and charging.

S-GW – The Serving Gateway supports the transport of the user data between the UE and the external networks. All user IP packets are transferred through the Serving Gateway, which serves as the local mobility anchor for the data bearers when the UE moves between eNodeBs. The S-GW is connected to the P-GW. The two gateways (S-GW and P-GW) are specified independently but they can also be collocated. Both gateways (S-GW and P-GW) support the user plane (i.e., user data).

MME – The Mobility Management Entity is the control node that processes the signaling between the UE and the CN. The protocols running between the UE and the CN are known as the Non Access Stratum (NAS) protocols [20]. The main functions supported by the MME can be classified as: *functions related to bearer management* and *functions related to connection management*. The former includes the establishment, maintenance and release of the bearers and is handled by the session management layer in the NAS protocol; while the latter includes the establishment of the connection and security between the network and UE and is handled by the connection or mobility management layer in the NAS protocol layer.

The evolved NodeB (eNodeB) is the main element of the LTE's access network, E-UTRAN [3], providing the E-UTRA user plane and control plane protocol terminations towards the UE. The eNodeBs are interconnected with each other by means of the X2 interface, as depicted in Figure 1.3. The eNodeBs are also connected by means of the S1 interface to the EPC, more specifically to the MME by means of the S1-MME and to the S-GW by means of the S1-U. The S1 interface supports a many-to-many relation between MMEs / S-GWs and eNodeBs.

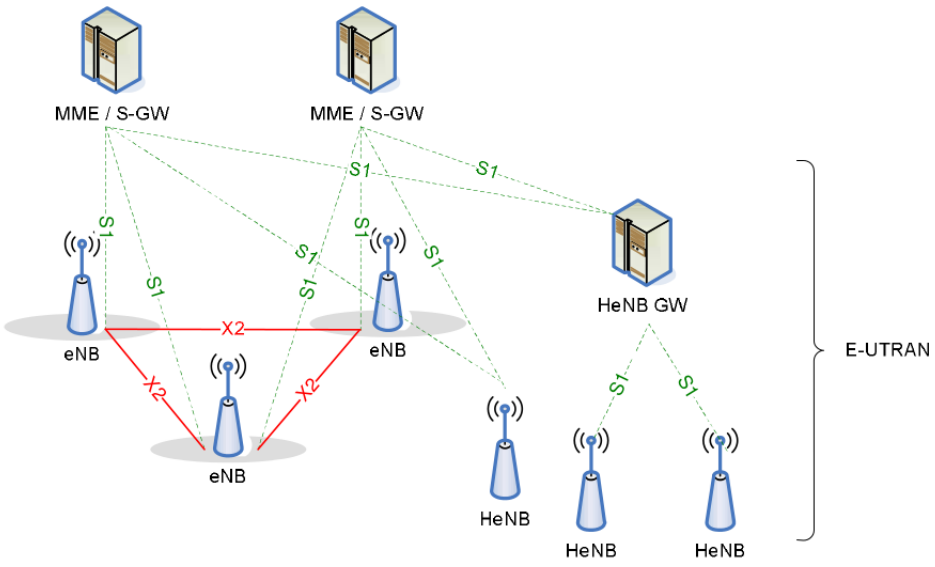


Figure 1.3: Overall E-UTRAN architecture with deployed HeNB, copied from [3]

Release 9 (2010) [21] introduces the definition of Home (e)NodeB Subsystem (HeNB). These systems allow unmanaged deployment of femtocells at indoor sites, providing almost perfect broadband radio coverage in residential and working areas, and offloading the managed, pre-panned macro-cell network. The E-UTRAN architecture may therefore deploy a HeNB Gateway (HeNB GW) to allow the S1 interface between the HeNB and the EPC to scale to support a large number of HeNBs. The HeNB GW serves as a concentrator for the control plane, specifically the S1-MME interface. The S1-U interface from the HeNB may be terminated at the HeNB GW, or a direct logical user plane connection between HeNB and S-GW may be used.

The E-UTRAN is responsible for all radio-related functions, which [2] briefly summarized as:

- Radio Resource Management (RRM): covers all functions related to the radio bearers, such as radio bearer control, radio admission control, radio mobility control, scheduling and dynamic allocation of resources to UEs in both uplink and downlink.
- Header Compression: helps to ensure efficient use of the radio interface by compressing the IP packet headers that could otherwise represent a significant overhead, especially for small packets such as VoIP.
- Security: all data sent over the radio interface is encrypted.
- Connectivity to the EPC: consists of the signaling toward MME and the bearer path toward the S-GW.

On the network side, all of these functions reside in the eNodeBs/HeNBs. Each eNodeB/HeNB can be responsible for managing multiple cells, thus reducing latency and improving efficiency.

The data plane of E-UTRAN works as follow: an IP packet for a UE is encapsulated in an EPC-specific protocol and tunneled between the P-GW and the eNodeB for transmission to the UE. Different tunneling protocols are used across different interfaces. A 3GPP-specific tunneling protocol called Evolved GPRS Tunneling Protocol (eGTP) [12] is used over the EPC interfaces, S1-U and S5/S8. Alternatively Proxy Mobile IP [11] can be used in the S5/S8 tunnel and it is currently deployed in case of non-3GPP access [22].

Applying network virtualization to the LTE network means to virtualize the infrastructure of the LTE system (that is eNodeBs/HeNBs, routers, Ethernet links etc.) and let multiple network operators create their own virtual network depending on their requirements and goals, while using a common infrastructure. The challenges of that are mainly how to virtualize the physical infrastructure to support such scenarios, and what kind of changes are required to be introduced to the LTE system.

Since network virtualization is receiving immense attention in the research community all over the world, there have already been different approaches to virtualize different aspects of the LTE network:

some are focusing on resource virtualization like eNodeBs [23, 24], server [25] and router virtualization [26, 27], others are focusing on building a framework to set up virtual networks on the fly based on different virtual resources [28] and lately some researches focused on the virtualization of the LTE wireless medium [29, 30, 31].

Architecture of cloud components

Currently the MCN project [16] has designed an architecture of cloud components, [4], to support the on demand provision of cloudified mobile networks, from the access part to the service platforms including the core network, to enterprise end users (EEUs).

The access part is provided through the Radio Access Network as a Service (RANaaS) which architecture reference model is depicted in Figure 1.4.

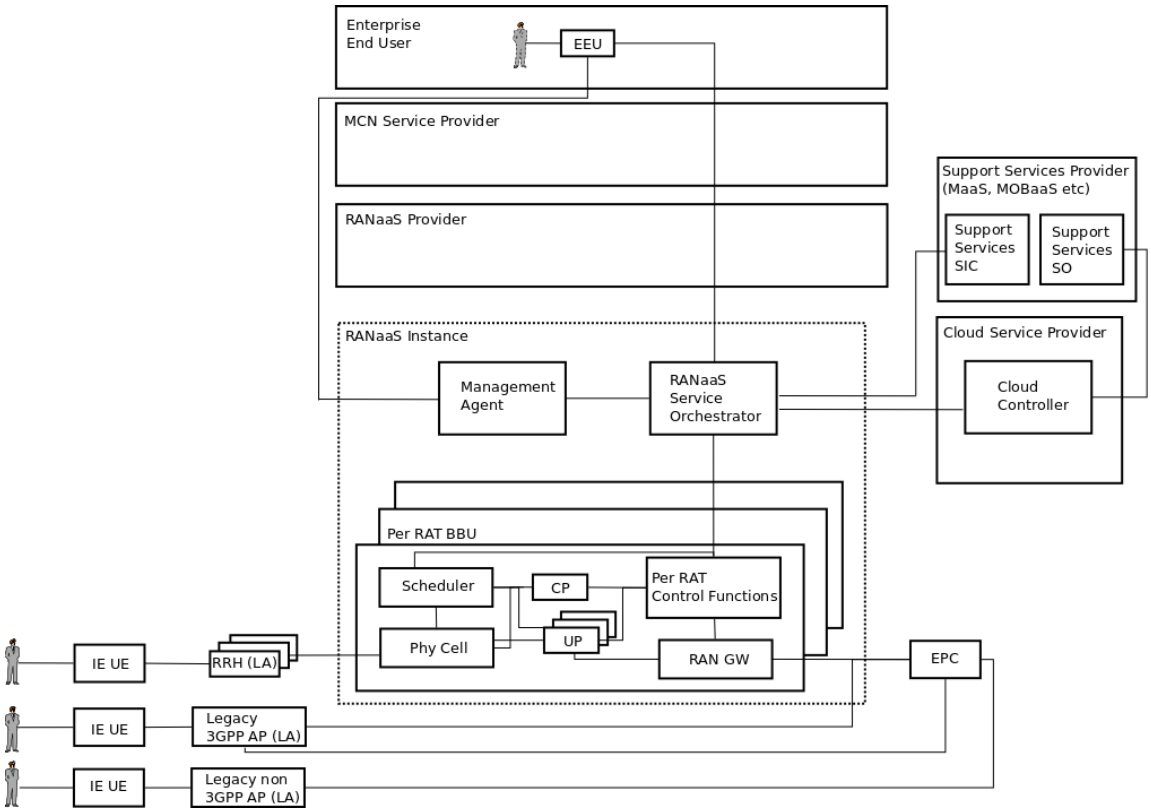


Figure 1.4: RANaaS architecture reference model, based on [4]

The main tasks of each architectural components have been defined in [32] and they are summarized below:

- CP: processes common control information (independent of UEs) in uplink and downlink and includes part of the air interface radio control plane stack.

- Legacy Agents (LAs): support function that interworks with RANaaS related support services instances (e.g. Monitoring as a Service or MaaS). The Legacy Agents are managed and controlled by the Management Agent.
- Management Agent: authenticates and authorized EEU requests. If the AAA successfully completes the Management Agents forwards requests to the Service Orchestrator. The Management Agent is also responsible for configuring the Legacy Agents.
- Monitoring Agent: extracts information (based on regular updates, triggers or on request) from each MCN service and exposes such information in a constant manner to a logically centralized monitoring system.
- Per Radio Access Technology (RAT) Control functions: covers all the eNodeB control functions regarding the S1-MME, X2, and air interface.
- Phy Cell: constitutes the air interface Layer1 processing part of a cell. Its functionalities are controlled by the scheduler.
- Radio Access Network Gateway (RAN GW): is the only node seen to the outside world. Its role is to offer security between the destination network node and the RANaaS service instance and to route downlink packets to the relevant RANaaS functional element. The RAN GW can be placed inside or outside the Radio Access Technology's Base Band Unit (RAT's BBU).
- Scheduler: dynamically controls the access to the shared radio resources according to QoS parameters.
- RANaaS Service Orchestrator (SO): can make decisions (based on monitoring and policies) about the usage of virtual resources, and acts towards the Cloud Controller (CC) to execute them. It is able to configure Service Instance Components (SICs), either directly or through the CC upon request from the EEU or in reaction to internal RANaaS triggers.
- Individual End-user User Equipment (IE UE): is a device used by an end-user to have access to services provided by the network.
- UP: corresponds to the dedicated user processing that is required per user radio bearer in uplink and downlink. More specifically, it includes S1-U termination stack (Layer1, Layer2, UDP/IP, GTP), air interface radio user plane stack and part of air interface radio control plane stack (Layer1 and Layer2).

The EPCaaS can be seen as a cloudified implementation of 3GPP EPC architecture, which can be provided on demand to EEUs. The EPCaaS architecture reference model can be seen in Figure 1.5.

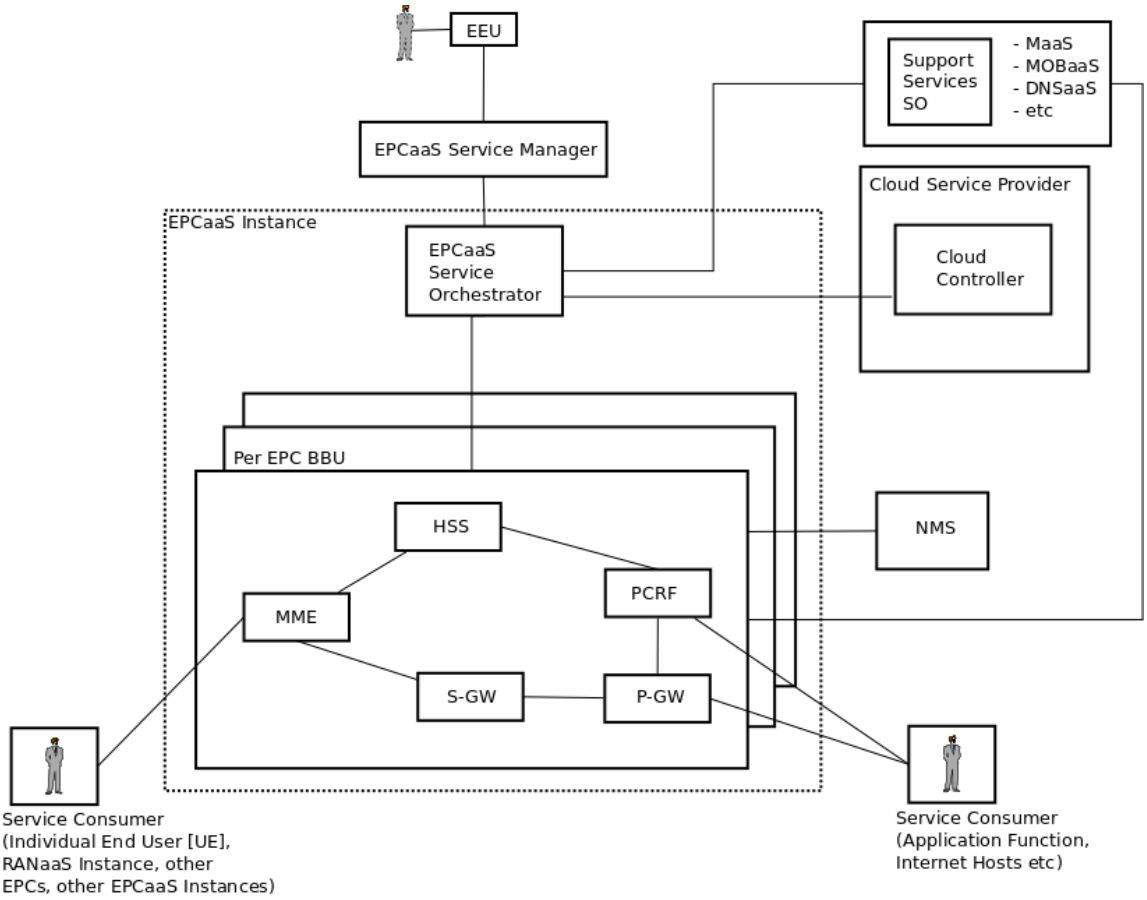


Figure 1.5: EPCaaS architecture reference model, based on [4]

Figure 1.5 describes a particular implementation option for the EPCaaS in which each 3GPP standard functional entity is mapped to one Service Instance Component (SIC) which is ultimately mapped to one virtual machine (VM). Although other implementation options exist, this particular model is used as the reference architecture model for EPCaaS in the remainder of this report.

Having already described the functionalities of the MME, S-GW, P-GW, HSS and PCRF entities, the remaining functional elements of a EPCaaS Instance are shortly described below. For a complete description of each entity's tasks the readers are advised to refer to [33].

- Network Management System (NMS): can be considered as a traditional, legacy management platform which provides the operator with an aggregated view on all EPC entities deployed in the network. Furthermore the NMS collects EPC-specific service counters from the devices and manages the alarms in case of faults. Although the NMS is not part of the EPCaaS, if the SICs expose proper interfaces towards this legacy element, it can be used to manage them, as if they were "real" component. Nevertheless, if NMS exposes API which can be used by external applications, it could be used as a source of information for e.g. Monitoring as a Service (MaaS) and eventually for the Service Orchestrator (SO), to run its decision algorithms.

- EPCaaS Service Orchestrator (SO):
  - chooses the initial placement of the SICs, both in terms of geographical location of the data center and physical host. This decision is especially important for both S-GW and P-GW, because user data traffic is routed through them and an optimized placement is therefore required.
  - Triggers a scaling out/in operation of the EPC entities and in which geographical location, to closely follow the capacity demanded by the IE UEs.
  - Triggers a migration of a running SIC to another physical host or data center, to support energy saving or maintenance operations and possibly to move P-GW closer to the subscribers.

  To run its decision algorithms, the EPCaaS SO collects information from several sources. Particularly important is the interaction with Mobility and Bandwidth Availability Prediction as a Service (MOBaaS), which can provide the SO with predictions about the bandwidth requested in a certain geographical area at a certain time by an aggregated group of IE UEs.
  Additionally, the EPCaaS SO is in charge of the initial configuration of the various SIC during the provisioning phase, which includes the configuration of service parameters such as 3GPP-specific EPC parameters, like GPRS timers, identifiers, QoS policies etc. and the provisioning of subscriber data in both HSS and PCRF entities.

1.1.2 Limitation of currently deployed mobility management schemes

Mobility management provides the mechanisms for maintaining active session continuity while a user switches across personal, local, and wide area networks. Due to the mobile nature of the UEs and the introduction of femto- and pico-cells from Release 9, handovers occur very often. In LTE, network-controlled UE-assisted handovers are performed. A centralized mobility management approach based either on eGTP or Proxy Mobile IPv6 has been deployed since LTE Release 8 where the P-GW acts as mobility data plane anchor.

When the UE first connects to the RAN via an eNodeB or HeNB, the data traffic to/from the UE is anchored to the P-GW (via the current S-GW). UE's traffic is encapsulated in an eGTP tunnel between the eNodeB/HeNB and the S-GW as well as in another (Proxy Mobile IPv6 tunneling can also be used here) between the Serving and PDN Gateways.

When the UE moves from one eNodeB/HeNB to another, to maintain the interested IP flow(s) active, the data traffic to/from the UE remains anchored to the same P-GW while the S-GW entity may need to be relocated. If the S-GW is relocated, a S5/S8 tunnel is established between the new S-GW and the P-GW and a S1-U tunnel is established between the new S-GW and the target eNodeB/HeNB. Only the latter is established in the case when the S-GW remains unchanged upon handover.

The procedure described above shows that the LTE's data plane is highly centralized and hierarchically organized requiring the management of either one or several hierarchical tunnels, in order to maintain the data path between the centralized anchor point (P-GW) and the UE. Although very simple to deploy and manage, tunneling introduces data overhead due to the necessary encapsulations, as well as data processing at the tunnel end-points to perform encapsulation/de-capsulation functions. In a wide area network like LTE, the centralized anchors need to maintain a considerable number of per-user tunneling contexts, in the range of millions for a nationwide network, which may cause scalability issues. The aggregated traffic is also huge and expected to grow exponentially in the future posing a higher concern on well known data path centralization issues such as the presence of a single point of failure and the creation of bottlenecks.

Furthermore routing via a centralized anchor often results in a longer route that increases the delay in the communication between two end-points and wastes network resources. Two different cases can be identified. In the first case, the transmitting and the receiving UEs are close to each other but are both far from the mobility anchor. Packets destined to the UE need to be routed via the P-GW, which is not in the shortest path. The second case involves a Content Delivery Network (CDN) [34]. A user may obtain content from a server, such as when watching a video. As such usage becomes more popular, resulting in an increase in the core network traffic, service providers may relieve the core network traffic by placing these contents closer to the users in the access network in the form of cache or local CDN servers. Yet as the UE is getting content from a local or cache server of a CDN, even though the server is close to the UE, packets still need to go through the core network to route via the centralized mobility anchor. This can lead to very high bandwidth requirements on core network equipment.

In order to tackle some of these issues Local IP Access (LIPA) and Selected IP Traffic Offload (SIPTO) [35] have been developed within the 3GPP and introduced in Release 10 [36]. LIPA enables an IP capable UE connected via a HeNB to access other IP capable entities in the same residential/enterprise IP network without the user plane traversing the mobile operator's network core. SIPTO, on the other hand, enables an operator to offload certain types of traffic at a network node close to the UE's point of attachment to the access network, by selecting a set of GWs (S-GW and P-GW) that is geographically/topologically close to the UE current position.

Unfortunately both SIPTO and LIPA have a very limited mobility support. In the latest Release 11 [37] and Release 12 [38], there is currently a work item on LIPA Mobility and SIPTO at the Local Network (LIMONET) [39] that is studying how to provide SIPTO and LIPA mechanisms with some additional, but still limited (only local), mobility support.

1.2 Motivation

Following the architectural innovations introduced by mechanisms such as LIPA and SIPTO and driven by the distributed nature of future cloud based mobile networks, the deployment of a Dis-

tributed Mobility Management solution is currently being analyzed by both academic and standards communities (IETF and 3GPP). DMM implements the concept of a flatter LTE system in which the mobility anchors are placed closer to the access network. A hierarchical layer used in data plane forwarding can be removed from the network by allowing the co-location of S-GW and P-GW entities. Besides overcoming the limitations and problems posed by the deployed centralized scheme, DMM, can aim at the development of a cross-operator mobility management system which is currently unsupported in LTE standard.

The integration of the Cloud Computing concept in LTE mobile networks increases LTE's performance by building a shared distributed LTE mobile network that can optimize the utilization of virtualized computing, storage and network resources and minimize communication delays. The use of DMM can be applied in such environments also to support traffic redirection when a virtualized network function running on a virtualization platform is migrated to another virtualization platform and ongoing sessions need to be maintained in the UEs' current mobility anchor.

1.3 Problem statement

In current LTE systems, UEs change their mobility anchor points (P-GW) rarely and, if they remain attached to the same operator's access network, often as a consequence of long range movements (e.g. using vehicles). When this happens a *PDN Disconnection* procedure will be triggered by the network for each IP flow initiated at the previous P-GW. The same occurs when movements cause UEs to attach to the eNodeB belonging to a different operator, i.e. inter-operator roaming. A P-GW serving the newly accessed network will be used to anchor UEs' re-initiated traffic.

Distributed Mobility Management and the introduction of the Cloud Computing concept in LTE mobile networks will lead to the deployment of flatter systems in which mobility anchor points are placed closer to the end users. It is therefore expected that the relocation of UEs' mobility anchors upon handover will happen far more often.

DMM will have to provide the network with the possibility to keep ongoing sessions active also upon handovers that require mobility anchor relocation (in both intra- and inter-operator scenarios). Furthermore, when a virtualized P-GW entity is migrated to a new location, DMM can be used to support traffic redirection in order to maintain ongoing sessions supported by this P-GW.

As a consequence, IP address continuity needs to be supported in the network to allow the anchoring of UEs' previously initiated IP flows to the relocated or migrated P-GW although the flows' destination IP addresses are topologically unrelated to the location of the relocated or migrated P-GW.

1.4 Research questions

The objective of this research is to demonstrate the possibility to implement a DMM solution which can provide seamless change or migration of a virtualized mobility anchor to cloud based LTE systems. Therefore the main research question is:

How can seamless DMM be implemented and evaluated in cloud based LTE system?

In order to answer this question the following sub-questions have been defined:

1) Which requirements need to be satisfied by a DMM solution when applied in cloud based LTE systems?

2) Which architecture/framework can be used for the support of DMM in cloud based LTE systems?

3) Which of the existing DMM solutions can be applied in cloud based LTE systems?

4) How can the DMM solution be implemented in cloud based LTE systems?

5) How can the seamlessness of the DMM solution be assessed and verified?

1.5 Organization of the report

The remainder of this report is structured as follows:

Chapter 2 introduces the concept of Distributed Mobility Management. The requirements to design a DMM scheme for a cloud based LTE system are presented together with a functional framework that will be used by the proposed entity. The last Section of this Chapter specifies in details the modifications required to the current 3GPP's X2- and S1-based handover and PDN disconnection standard procedures to support IP address continuity in the EPS when the UE's mobility anchor point is relocated. This Chapter answers research sub-questions 1 and 2.

Chapter 3 proposes two DMM solutions which can be applied in a cloud based LTE system. A preliminary comparison is also carried out in this Chapter. Research sub-questions 3 and 4 are fully addressed in this Chapter.

Chapter 4 introduces the simulation environment which will be used to evaluate the DMM solutions proposed in the previous Chapter. The topology and parameters of the simulation will be discussed here followed by the description of the selected performance metrics. The experiment scenarios for each of the proposed DMM solution will be presented also in this Chapter.

Chapter 5 describes in detail the simulation experiments and evaluations of the simulation results. This Chapter together with Chapter 4 provide the answer to research sub-question 5.

Chapter 6 concludes the report and describes recommendations for future works. Additional information can be found in the documentations listed in the Bibliography.

Chapter 2

Distributed Mobility Management

The goal of this Chapter is to explain why DMM is considered an enhancement to current telecommunication network deployments, what are the requirements that need to be fulfilled by a DMM solution, which architectural design must be deployed to implement such solution in a cloud based LTE system and finally what are the modifications required to current 3GPP's LTE standard procedures to support IP address continuity in the EPS.

2.1 Integration in cloud based LTE systems

Centralized mobility management approaches have been proven to be prone to several problems and limitations [13, 14, 15]: suboptimal routing, low scalability, signaling overhead, more complex network deployment, introduction of a potential single point of failure and a lack of granularity on the mobility management service.

For this reason several mobile operators are now looking for alternative mobility solutions that are more distributed in nature, allowing cheaper and more efficient network deployments. This new paradigm of solutions is defined as Distributed Mobility Management (DMM). DMM implements the concept of a flatter system in which the mobility anchors are placed closer to the user distributing the control and data infrastructures among the entities located at the edge (access) of the network.

DMM may be partially or fully distributed, whether in the former the distribution scheme is applied only to the data plane while in the latter to both the data and control planes. With a partially distributed scheme a centralized entity is needed to provide locations information regarding the mobile nodes to the distributed anchor points. In a fully distributed approach this information is stored at every distributed anchor point which shares it with its peers when needed. It is important to notice that in the fully distributed approach data and control planes needs to be decoupled although they are both handled by the distributed anchor points.

While the partially distributed approach introduces a single point of failure and increases the signal-

ing load in the core network, the fully distributed approach comes with a major issue in the lack of knowledge that each distributed anchor point has of its peers and their connected nodes. For this reason handover procedures may introduce latency and overheads. Furthermore a scheme to address this issue needs yet to be researched whereas possible solutions are: layer-2 mechanisms with capability to retrieve the IP addresses configured in the UE, a peer-to-peer communication service between the anchor points or a distributed scheme that allows IP discovery (either unicast or multicast).

Following the current LTE standard, this research focuses on the deployment of a partial distributed scheme. In the remainder of this Section the requirements of a DMM solution when applied in cloud based LTE system will be presented together with the functional entities framework and architecture that can be used to support DMM in cloud based LTE system.

2.1.1 Requirements

In August 2010 a new IETF non-working group called Distributed Mobility Management [40] has been rechartered with the goal to extend current IP mobility solutions for flat networks architectures.

[41] defines the IETF requirements for DMM in IPv6 networks deployments. In the following for each requirement a correspondent set of features that needs to be supported by the future EPC is presented. Some additional requirements are added at the end of this subsection.

Distributed deployment *IP mobility, network access and routing solutions provided by DMM must enable distributed processing for mobility management so that traffic does not need to traverse centrally deployed mobility anchors and thereby avoid non-optimal routes.*

The distribution of the mobility anchors will improve scalability and robustness of the mobility infrastructure. Placing the mobility management closer to the edge of the network (e.g. just above the access network) will attain routing optimality and lower delays. Beside, offloading near the edge of the network would become possible, to the benefit of the core network load.

Since Release 8 and 9 the RAN of LTE has become flattened to one serving node (i.e. the eNodeB/HeNB). This helped the distribution of the control plane functionalities to the edge of the network. The same needs to be done for the data plane introducing a new logical entity that embodies functionalities of both S-GW and P-GW and which is placed on the next hierarchical level above the access network.

Transparency to upper layer when needed *DMM solutions must provide transparent mobility support above the IP layer when needed. Such transparency is needed, for example, when, upon change of point of attachment to the Internet, an application flow cannot cope with a change in the IP address. However, it is not always necessary to maintain a stable home IP address or prefix for every application or at all times for a mobile node.*

In other words, it shall be possible to offload selected traffic (e.g. Internet) by moving IP flows from one access to another. A solutions to maintain active the flows that a user has initiated on a previous anchor point needs to be deployed in the EPC. It is fundamental that the UE remains unaware of the occurred handoff procedure.

IPv6 deployment *DMM solutions should target IPv6 as the primary deployment environment and should not be tailored specifically to support IPv4, in particular in situations where private IPv4 addresses and/or NATs are used.*

IPv6 is supported by LTE networks.

Existing mobility protocols *A DMM solution should first consider reusing and extending IETF-standardized protocols (MIPv6 and its extensions) before specifying new protocols.*

PMIPv6 has been adopted by the 3GPP EPS network and therefore solutions derived from this scheme will need to be considered between the first choices to implement distributed mobility management in LTE networks.

Co-existence *The DMM solution must be able to co-exist with existing network deployments and end hosts. For example, depending on the environment in which DMM is deployed, DMM solutions may need to be compatible with other deployed mobility protocols or may need to interoperate with a network or mobile hosts/routers that do not support DMM protocols. The mobile node may also move between different access networks, where some of them may support neither DMM nor another mobility protocol.*

Furthermore, a DMM solution should work across different networks, possibly operated as separate administrative domains, when allowed by the trust relationship between them.

The recent introduction of mobile devices with multiple network interfaces (e.g. 3G/4G, WLAN and Bluetooth) together with their capability to be simultaneously connected to more than one access network enforces the need of interoperability between 3GPP and non-3GPP mobility solutions across different access networks.

Security considerations *A DMM solution must not introduce new security risks or amplify existing security risks against which the existing security mechanisms/protocols cannot offer sufficient protection.*

The EPC needs to provide two different security levels: first, access network security that only allows a legitimate UE/router to access the DMM service; second, end-to-end security that protects signaling messages for the DMM service. Access network security is required between the UE/router and the entity where the DMM is performed (e.g. P-GW). End-to-end security is required between UEs that participate in the DMM protocol.

It is necessary to provide sufficient defense against possible security attacks, or to adopt existing security mechanisms and protocols to provide sufficient security protections. For instance, EAP-based authentication can be used for access network security, while the IP-nature of LTE makes IPsec suitable for end-to-end security.

Flexible multicast distribution *DMM should consider multicast early so that solutions can be developed not only to provide IP mobility support when it is needed, but also to avoid network inefficiency issues in multicast traffic delivery (such as duplicate multicast subscriptions towards the downstream tunnel entities). The multicast solutions should therefore avoid restricting the management of all IP multicast traffic to a single host through a dedicated (tunnel) interface on multicast-capable access routers.*
Multicast needs to be considered so that solutions can be developed to overcome performance issues in multicast distribution scenario.

In addition to the above, the following requirements also need to be taken into consideration in the definition of DMM solutions for a cloudified LTE system.

Dynamicity The dynamic use of mobility support by allowing the split of data flows along different paths that may travel through either the mobility anchor or non-anchor nodes, even though no specific route optimization support is available at the correspondent node. This requirement will tackle the lack of fine granularity of the centralized mobility management approaches.

Control and data plane separation Separating control and data planes by splitting location and routing anchors. Keeping the control plane centralized while distributing the data plane (partially distributed mobility management) is a possible solution to minimize the signaling overhead between the mobility anchors due to the lack of knowledge that a distributed anchor point has of its peers and their connected UEs.

Network-based Not burdening the UE with extra signaling and keeping the user unaware of the on-going handoff procedure within the same domain are fundamental aspects that need to be provided by the DMM solutions deployed in LTE networks. For this reason is legit to prefer a network-based mobility management solution over a client-based one.

2.1.2 Functional framework

Also in the context of IETF DMM WG, M. Liebsch et. al [42] define a functional framework for DMM and introduce a set of Functional Entities (FEs) which are required to support IP address continuity in a network with distributed mobility anchors.

Based on their work the functional entities and their tasks needed to support IP address continuity in a DMM solution for cloud based LTE system are defined as follow:

- FE_MCTX (Functional Entity Mobility Context Transfer): the task of this function is to export relevant mobility context information from the UE's previous mobility anchor (source P-GW) and to import this information on the UE's current mobility anchor (target P-GW) to enable IP address continuity after mobility anchor relocation. Furthermore the FE_MCTX can provide mobility context information also to a control function (FE_IEC) to allow forwarding of packets to the UE's currently used mobility anchor.

  This function is co-located with the EPC MME entity.

- FE_I (Functional Entity Ingress to DMM plane): this function establishes the forwarding of the UE's packets to the appropriate DMM Egress function (FE_E). Information to implement this traffic redirection can be retrieved from a control function (FE_IEC) in a reactive fashion or they can be proactively delivered to the Ingress function by the control function upon change of the UE's mobility anchor.

- FE_E (Functional Entity Egress of DMM plane): terminating the DMM data forwarding plane, this function receives downlink packets forwarded by the DMM Ingress function (FE_I). The task of this function is to identify a UE's received packets and deliver them to the UE's current mobility anchor (target P-GW). The state of the DMM Egress function will be established through the DMM Ingress/Egress Control function (FE_IEC) either in a proactive or reactive manner.

  This function can be co-located with the EPC P-GW entity or it can be placed closer to it (e.g. next hop on the SGi interface). If the DMM Egress function is not co-located with the EPC P-GW entity, forwarding techniques can be used to deliver UE's downlink data packets to the currently used mobility anchor.

- FE_IEC (Functional Entity for Ingress/Egress Control): the task of this function is to establish, update and remove policies in the DMM Ingress (FE_I) and Egress (FE_E) functions either in a proactive or reactive fashion to allow forwarding of UE's data packets towards the currently used mobility anchor (target P-GW).

  UE's mobility context information are delivered to this function by the FE_MCTX function of the UE's serving MME upon triggering of the specific handover procedure.

After UE's handover with mobility anchor relocation, the IP address (or prefix in case of IPv6) carried in the source address field of the uplink packet is topologically incorrect. For the uplink packets to be assumed as routable, IP routers of the mobility domain must not apply filtering according to the source addresses. Therefore DMM traffic redirection is needed only for downlink traffic towards the

UE that kept his previous IP address active, thus the Ingress function will always need to be placed further north in the transport network than the Egress function.

Downlink traffic forwarding by the DMM Ingress function can be for example accomplished by an IP tunnel to the Egress function, address translation to a routable IP address or other means.

2.1.3 Architecture of functional entities

[42] depicts and describes two DMM architecture deployment variants using the functional entities introduced above.

The first variant, namely Mobility Protocol Centric Solution, is depicted in Figure 2.1 and it aims to extend available IP-based mobility protocols (e.g. Mobile IP, Proxy Mobile IP) to enable DMM, without being dependent on external component and protocol. The control plane functionalities (FE_IEC and FE_MCTX) are implemented as extensions to the existing mobility protocol and therefore are co-located with the mobility anchor. IP address continuity on the data plane is achieved establishing a forwarding tunnel from the previous mobile anchor to the currently used one.

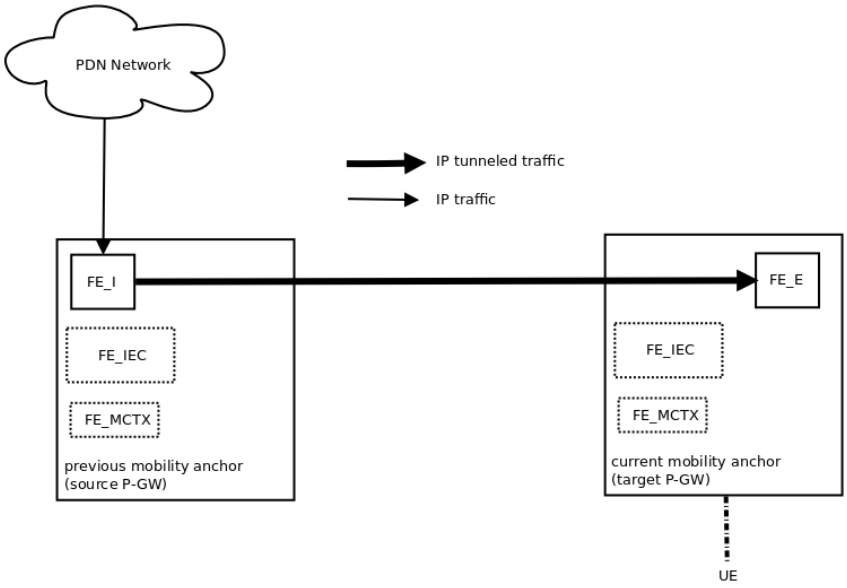


Figure 2.1: Mobility Protocol Centric Solution

The main drawback of this solution is the sub-optimal routing path due to the fact that downlink traffic needs to traverse the location of the IP address topologically correct mobility anchor.

Differently from the first variant, in the second architecture (depicted in Figure 2.2) the functional entities are distributed and protocol operations are implemented between them. This solution, namely Distributed Architecture, is considered to be independent of the mobility architecture and protocol.

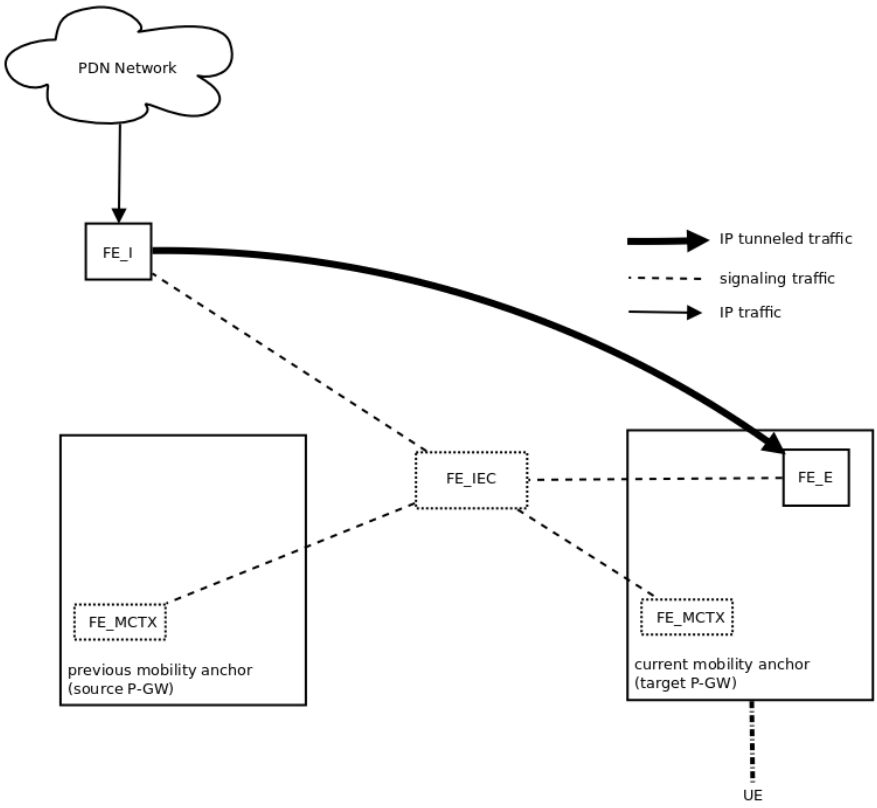


Figure 2.2: Distributed Architecture

Optimal routing is achieved placing the DMM Ingress function in a location further north than the DMM Egress function in the transport network. IP tunneling is still used to forward the downlink traffic to the currently used mobility anchor but without traversing the IP address topologically correct mobility anchor.

The functional entities architecture used in the definition of the DMM solutions that will be proposed further in this report is based on the Distributed Architecture introduced above and it is depicted in Figure 2.3.

To avoid encapsulation overhead, alternative forwarding techniques will be used instead of IP tunneling to deliver the UE's downlink traffic to the currently used mobility anchor. Furthermore the FE_MCTX function has been removed from the mobility anchor and it has been co-located with the EPC MME. A direct signaling path is needed between the MME and the entity where the Ingress/Egress Control function is located. The Ingress/Egress Control function can also be co-located with the MME which will then need to be extended to support signaling toward the entities that implement the Ingress and Egress functions.

Differently from what depicted in Figure 2.1 and 2.2, in Figure 2.3 the two DMM Egress functions have different positions. This is done to show the possibility to choose whether the Egress function can be co-located with a P-GW entity or not but the chosen positions are just an example. The only

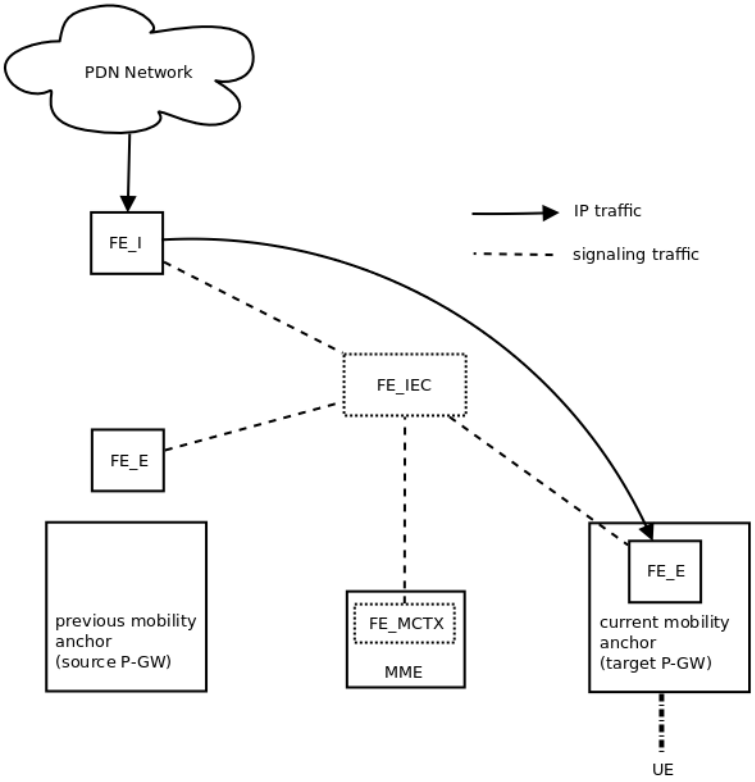


Figure 2.3: Functional entities architecture without IP tunneling

constraint on the placement of the Egress functions is that they must be placed further south than the Ingress function in the transport network. When the serving Egress function is located on an entity placed closer to the mobility anchor point (e.g. next hop on the SGi interface, as shown for the source P-GW in Figure 2.3), forwarding techniques can be used to deliver the traffic to the mobility anchor point when needed (i.e. destination IP address topologically unrelated to the location of the currently used P-GW).

2.2 Modifications required to support IP address continuity in the EPS

When users change their EPS traffic anchor point, IP address continuity is not supported in current 3GPP's LTE standard. The data plane network has been designed in hierarchical fashion, leading to a placement of the traffic anchor points (P-GW) deeper in the core network. Following this model, UE traffic remains anchored to a single P-GW until the user moves out of the access network served by the specific P-GW. For instance this happens during inter-operator roaming procedure. In the minds of LTE developers this sort of movement was (rightly) expected to happen very rarely, but being the distribution of mobility anchor points closer to the edge of the network a requirement for a DMM solution, this thought might need to be revised. In fact users are expected to change their traffic anchor

point far more often.

Currently when upon handover a UE attaches to a different anchor point, flows initiated at the previous P-GW will be stopped. This is due to the fact that no standard that support the continuity of bearers after P-GW relocation exists yet. Moreover no handover procedure with P-GW relocation capability is available in 3GPP's technical specifications.

The purpose of this Section will be to present requirements and possible solutions to support inter-P-GW IP address continuity in the EPS. This Section requires an advanced knowledge of the LTE standard to be understood. Moreover it can be skipped, if desired, since not strictly related with the remainder of this document.

The main obstacle in the implementation of such an IP address continuity scheme for the current EPS is given by the fact that no signaling nor data forwarding scheme is available between two different P-GWs entities. But if P-GW and S-GW entities are combined into a single entity, the procedures and messages used to implement handover with S-GW relocation can be revised and used to support the usage of one IP address in different P-GW domains.

When required, a P-GW allocates an IP address for a UE from its pool of addresses. These IP addresses are topologically anchored to the P-GW and are meant to be used exclusively in its domain. A mechanism to support the continuity of an IP address when the users move from one P-GW domain (source P-GW) to another (target P-GW) needs to be implemented. This solution will consist of two steps:

1. Signal the target P-GW to implement a bearer for the moving UE without requiring a new IP address allocation. Bearer will be established for the IP address of flows kept active by the moving UE after handover. If more IP addresses require continuity, more bearers will be implemented at the target P-GW.

2. Signal the source P-GW to momentarily remove from their pool of addresses, the IP address(es) of flows kept active by the moving UE after handover. This means that until the address(es) are not re-inserted in the pool, they cannot be allocated to users requiring PDN connectivity at the source P-GW. So in this case, some modifications are needed also to the PDN disconnection procedure, since the source P-GW needs to be signaled to restore the IP address in its address pool.

Signaling procedure to both P-GWs is managed by the MME entity.

There are three kinds of bearers in LTE: Radio bearers, S1 bearers and EPS bearers. Their architecture is shown in Figure 2.4. Currently only Radio bearers and S1 bearers are fully portable in case of intra-E-UTRAN handover. Moving towards a flatter EPC architecture will lead to the unification of S-GW and P-GW into a single entity which will be referred in the following as S-/P-GW. The S5/S8 bearer

is therefore unnecessary and a direct map from External bearer to E-RAB (S1 bearer) is required. The downlink mapping scheme in the S-/P-GW will be performed from DL TFT to DL S1-TEID.

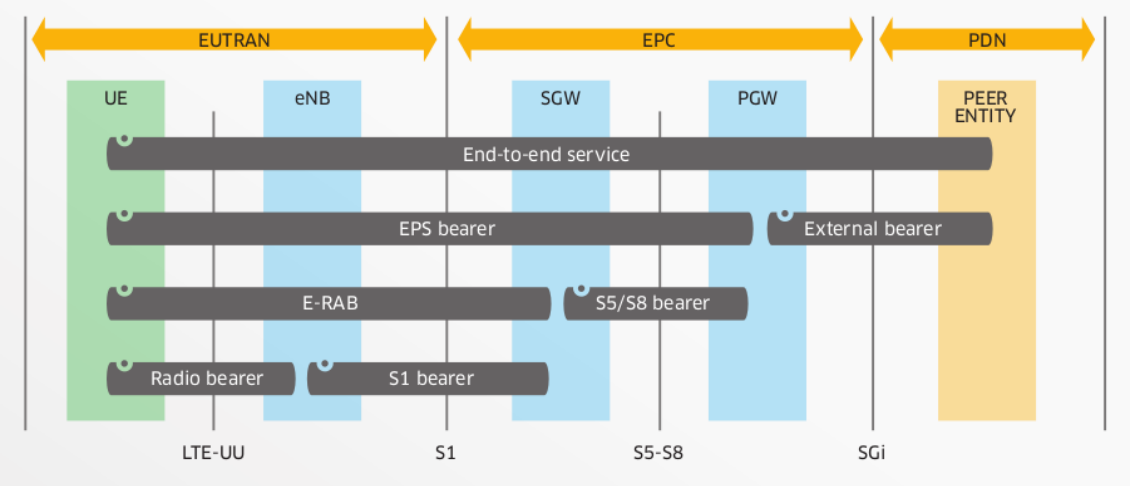


Figure 2.4: Bearer architecture, copied from [5]

A procedure to adjust the DL S1-TEID after handover with and without S-GW relocation have been already specified in [43] and it will be used as a reference to define a new model of handover where also the P-GW functionalities of a S-/P-GW entity will be relocated. Continuity of previously established bearers will need to be provided with the control plane being managed by the MME.

It is important to describe how downlink traffic is mapped to EPS bearers in the P-GW. At the P-GW downlink traffic is grouped into specific packet flows denoted as Service Data Flows (SDFs) which are then mapped into EPS bearers using a set of filter rules called downlink traffic flow templates (DL TFT). DL TFT were initially only used to filter traffic belonging to dedicated bearers but now is possible to assign a TFT to the default bearer too. The PCRF is the entity entitled to create these special filters while the PCEF of a P-GW uses the DL TFT to map traffic to an EPS bearer in the downlink direction. TFT are created by the PCRF after that the P-GW's PCEF performs an IP-CAN Session Establishment procedure as defined in [44]. Therefore to support the migration of an EPS bearer to a completely new EPS data forwarding plane, a new DL TFT has to be created for the moving UE and given to the target P-GW. One type of the information provided by the PCEF to the PCRF during the IP-CAN Session Establishment procedure is the UE IP address which therefore needs to be signaled to the target P-GW by the MME. A mechanism to do this will be explained in the following subsection.

2.2.1 X2-based handover with S-/P-GW relocation

This procedure is used to hand over a UE from a source eNodeB to a target eNodeB using X2 when the MME is unchanged and the MME decides that the S-/P-GW is to be relocated. The presence of IP connectivity between the source S-/P-GW and the source eNodeB, between the source S-/P-GW and

the target eNodeB, and between the target S-/P-GW and target eNodeB is assumed. (If there is no IP connectivity between target eNodeB and source S-/P-GW, it is assumed that the S1-based handover procedure in Section 2.2.2 shall be used instead.)

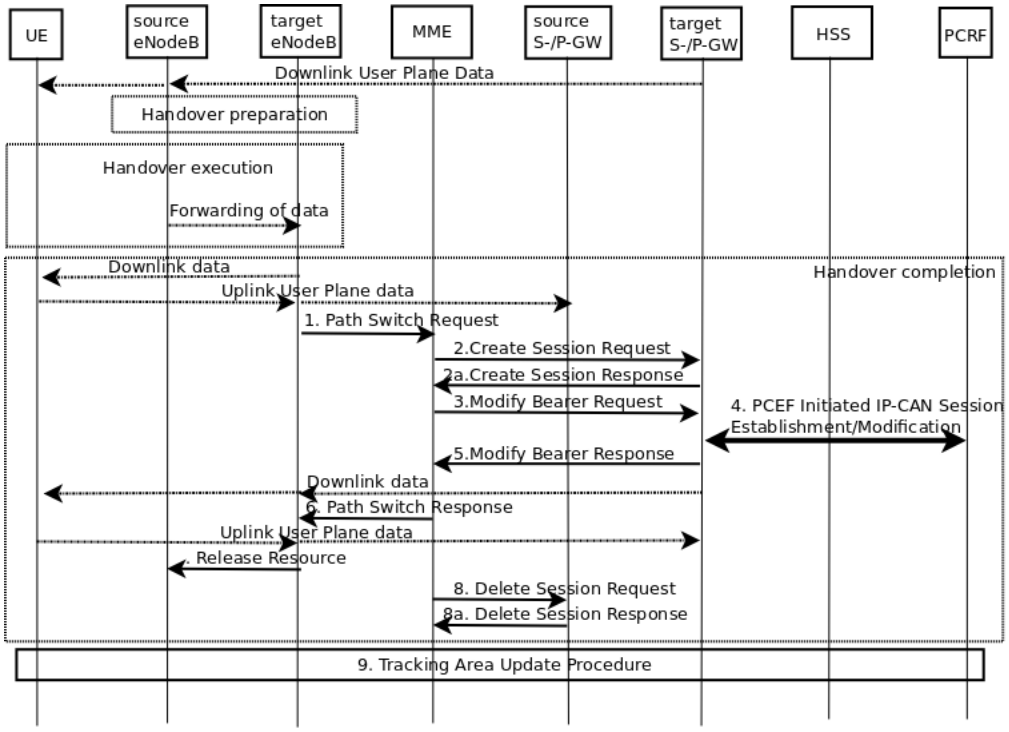


Figure 2.5: X2-based handover with S-/P-GW relocation

1. This step is the same as step 1 in Section 5.5.1.1.3 of [43] (S-GW replaced by S-/P-GW).

2. The MME stores the IP address and TEID used for signaling by the source S-/P-GW together with the EPS bearer identity(s) of the EPS bearer context(s) into a Initiator S-/P-GW Table. This table will be used to signal the correct S-/P-GW when the EPS bearer will be deactivated.

   The MME sends a Create Session Request (EPS bearer context(s), Serving Network, UE Time Zone) message per PDN connection to the target S-/P-GW. An EPS bearer context includes UE IP address, Serving Network and UE Time Zone. The target S-/P-GW allocates the S-/P-GW addresses and TEIDs for the uplink traffic on S1-U reference point (one TEID per bearer). The target S-/P-GW sends a Create Session Response (S-/P-GW addresses and uplink TEID(s) for user plane) message back to the MME. The MME starts a timer, to be used in step 8. If the source S-/P-GW continues to serve the UE, no message is sent in this step. In this case, the target S-/P-GW is identical to the source S-/P-GW. If the target S-/P-GW requested UE's location info, the MME also includes the User Location Information IE in this message.

   The MME uses the list of EPS bearers to be switched, received in step 1, to determine whether

any dedicated EPS bearers in the UE context have not been accepted by the target eNodeB. The MME releases the non-accepted dedicated bearers by triggering the bearer release procedure as specified in clause 5.4.4.2 of [43]. If the target S-/P-GW receives a DL packet for a non-accepted bearer, the S-/P-GW drops the DL packet and does not send a Downlink Data Notification to the MME.

If the default bearer of a PDN connection has not been accepted by the target eNodeB and there are multiple PDN connections active, the MME shall consider all bearers of that PDN connection as failed and release that PDN connection by triggering the MME requested PDN disconnection procedure (see Section 2.2.3 for more details). If none of the default EPS bearers have been accepted by the target eNodeB, the MME shall act as specified in step 6.

3. The MME sends a Modify Bearer Request (eNodeB address and TEID allocated at the target eNodeB for downlink traffic on S1-U for the accepted EPS bearers, ISR Activated) message to the target S-/P-GW for each accepted PDN connection. If the UE Time Zone has changed, the MME includes the UE Time Zone IE in this message. For the case that neither MME nor S-/P-GW changed, if ISR was activated before this procedure MME should maintain ISR. The UE is informed about the ISR status in the Tracking Area Update procedure. When the Modify Bearer Request does not indicate ISR Activated the S-/P-GW deletes any ISR resources by sending a Delete Bearer Request to the other CN node that has bearer resources on the S-/P-GW reserved.

4. If dynamic PCC is not deployed and the S-/P-GW is relocated, the target S-/P-GW executes a PCEF Initiated IP-CAN Session Modification procedure with the PCRF, as specified in [44], to report the new IP-CAN type. Depending on the active PCC rules, the establishment of dedicated bearers for the UE may be required. The establishment of those bearers shall take place in combination with the default bearer activation as described in Annex F of [43]. This procedure can continue without waiting for a PCRF response. If changes to the active PCC rules are required, the PCRF may provide them after the handover procedure is finished. The UE IP address provided to the PCRF function needs to be the same as received by the MME in the EPS bearer context(s) included in the Create Session Request message sent at step 2.

   If dynamic PCC is not deployed, the S-/P-GW may apply local QoS policy. This may lead to the establishment of a number of dedicated bearers for the UE following the procedures defined in clause 5.4.1 of [43] in combination with the establishment of the default bearer, which is described in Annex F of [43].

   If the S-/P-GW is not relocated, but has received the User Location Information IE and/or UE Time Zone IE and/or User CSG Information IE from the MME in step 2 or 3, the S-/P-GW shall takes into consideration these information that e.g. can be used for charging.

   If the S-/P-GW is not relocated and it has not received User Location Information IE nor UE Time Zone IE nor User CSG Information IE from the MME in step 2 or 3, nothing needs to be

done in this step and downlink packets from the S-/P-GW are immediately sent on to the target eNodeB.

5. The S-/P-GW creates a new entry in its EPS bearer context table and generates a Charging ID. The new entry allows the S-/P-GW to route user plane PDUs between the E-UTRAN and the Packet Data Network, and to start charging. The way the S-/P-GW handles Charging Characteristics that it may have received is defined in [44].

   The target S-/P-GW sends a Modify Bearer Response message to the target MME. The message is a response to a message sent at step 3.

   If the S-/P-GW does not change, the S-/P-GW shall send one or more "end marker" packets on the old path immediately after switching the path in order to assist the reordering function in the target eNodeB.

6. This step is the same as step 5 in Section 5.5.1.1.3 of [43] (S-GW replaced by S-/P-GW).

7. This step is the same as step 6 in Section 5.5.1.1.3 of [43].

8. When the timer started in step 2 expires the MME deletes the EPS bearer resources by sending Delete Session Request (Cause, LBI and EPS bearer context(s) of bearers accepted by the target S-/P-GW) message to the source S-/P-GW. Cause indicates to the S-/P-GW that the S-/P-GW changes and EPS bearer context(s) will be stored in the Transferred EPS Bearers Table at the S-/P-GW.

   The S-/P-GW acknowledges with Delete Session Response message. If ISR has been activated before this procedure, the Cause also indicates to the S-/P-GW that the source S-/P-GW shall delete the bearer resources on the other old CN node by sending Delete Bearer Request message(s) to that CN node.

9. This step is the same as step 8 in Section 5.5.1.1.3 of [43].

2.2.2 S1-based handover with S-/P-GW relocation

The S1-based handover procedure is used when the X2-based handover cannot be used. The source eNodeB initiates a handover by sending Handover Required message over the S1-MME reference point. This procedure may relocate the MME and/or the S-/P-GW. The source MME selects the target MME. The MME (target MME for MME relocation) determines if the S-/P-GW needs to be relocated. If the S-/P-GW needs to be relocated the MME selects the target S-/P-GW, as specified in clause 4.3.8.2 of [43].

A draft of the message flow of an S1-based handover procedure with MME and S-/P-GW relocations is shown in Figure 2.2.2. The S-/P-GW entity is defined as an entity embodying functionalities of both

S-GW and P-GW. This entity is considered as a single physical entity and when S-GW functionalities are relocated, P-GW functionalities will need to be relocated as well. S5/S8 bearer indicated in Figure 2.4 will be removed from the architecture and the EPS bearer will therefore be identical to the E-RAB. The name EPS bearer is kept anyway.

In the description of this procedure the virtualized EPC service instance components are assumed to be already in place and running when the handover is triggered.

The procedure is a modification of the S1-based handover (normal) procedure specified in Section 5.5.1.2.2 of [43].

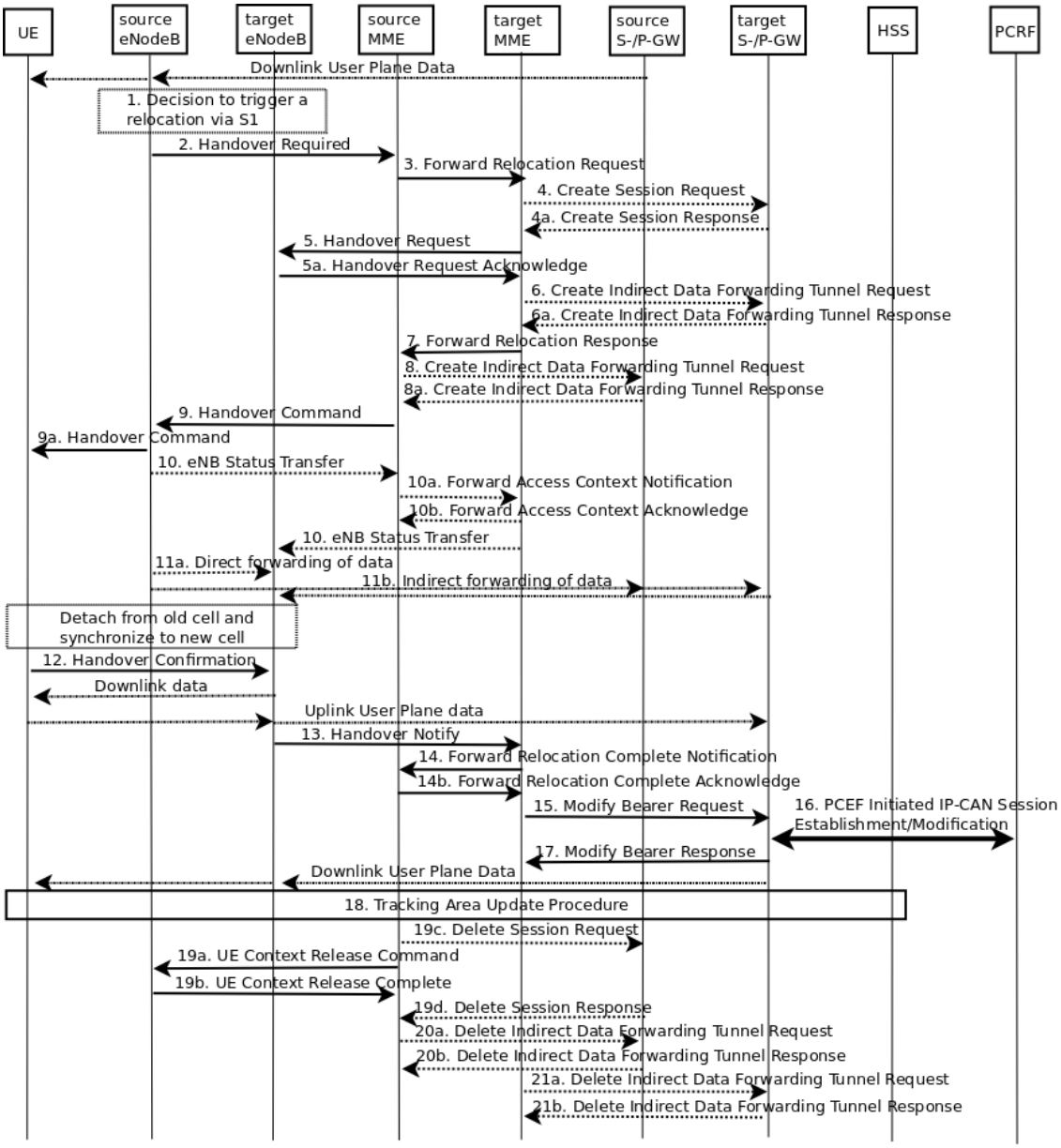


Figure 2.6: S1-based handover with S-/P-GW relocation

1. This step is the same as in Section 5.5.1.2.2 of [43].

2. This step is the same as in Section 5.5.1.2.2 of [43].

3. This step is the same as in Section 5.5.1.2.2 of [43] with the difference that in EPS bearer context(s), sent from MME to S-/P-GW, P-GW addresses and S5/S8 TEIDs are replaced by the UE IP address (UE IP address is sent to the MME by the source S-/P-GW during the E-UTRAN Initial Attach Procedure inside a Create Session Response message where it also replaces the PDN Type and PDN Address fields).

4. If the MME has been relocated, the target MME verifies whether the source S-/P-GW can continue to serve the UE. If not, it selects a new S-/P-GW. The "Serving GW Selection Function" described in clause 4.3.8.2 of [43] can be reused for this purpose.

   If the MME has not been relocated, the source MME decides on this S-/P-GW re-selection. The target MME stores the IP address and TEID used for signaling by the source S-/P-GW together with the EPS bearer identity(s) of the EPS bearer context(s) into a Initiator S-/P-GW Table. This table will be used to signal the correct S-/P-GW when the EPS bearer will be deactivated. Information regarding S-/P-GW signaling addresses and EPS bearer context(s) are retrieved by the target MME from the Forward Relocation Request message sent by the source MME. If no MME relocation is deployed the MME will still have to store this information into his Initiator S-/P-GW Table.

   If a new S-/P-GW is selected, the target MME sends a Create Session Request (EPS bearer context(s), Serving Network, UE Time Zone) message per PDN connection to the target S-/P-GW. An EPS bearer context includes UE IP address, Serving Network and UE Time Zone. The target S-/P-GW allocates the S-/P-GW addresses and TEIDs for the uplink traffic on S1-U reference point (one TEID per bearer). The target S-/P-GW sends a Create Session Response (S-/P-GW addresses and uplink TEID(s) for user plane) message back to the target MME. If the source S-/P-GW continues to serve the UE, no message is sent in this step. In this case, the target S-/P-GW is identical to the source S-/P-GW.

5. This step is the same as in Section 5.5.1.2.2 of [43] with the correction that S-GW addresses are intended as S-/P-GW addresses.

6. This step is the same as in Section 5.5.1.2.2 of [43] with the correction that S-GW addresses are intended as S-/P-GW addresses.

7. This step is the same as in Section 5.5.1.2.2 of [43] with the correction that S-GW addresses and S-GW change indication are intended as S-/P-GW addresses and S-/P-GW change indication.

8. This step is the same as in Section 5.5.1.2.2 of [43] with the correction that S-GW addresses are intended as S-/P-GW addresses.

9. This step is the same as in Section 5.5.1.2.2 of [43].

10. This step is the same as in Section 5.5.1.2.2 of [43].

11. This step is the same as in Section 5.5.1.2.2 of [43].

12. This step is the same as in Section 5.5.1.2.2 of [43]. Both uplink and downlink path are different though, being data forwarded from target S-/P-GW to target eNodeB.

13. This step is the same as in Section 5.5.1.2.2 of [43].

14. This step is the same as in Section 5.5.1.2.2 of [43] (S-GW replaced by S-/P-GW).

15. This step is the same as in Section 5.5.1.2.2 of [43] (S-GW replaced by S-/P-GW).

16. If dynamic PCC is not deployed and the S-/P-GW is relocated, the target S-/P-GW executes a PCEF Initiated IP-CAN Session Modification procedure with the PCRF, as specified in [44], to report the new IP-CAN type. Depending on the active PCC rules, the establishment of dedicated bearers for the UE may be required. The establishment of those bearers shall take place in combination with the default bearer activation as described in Annex F of [43]. This procedure can continue without waiting for a PCRF response. If changes to the active PCC rules are required, the PCRF may provide them after the handover procedure is finished. The UE IP address provided to the PCRF function needs to be the same as received by the MME in the EPS bearer context(s) included in the Create Session Request message sent at step 4.

    If dynamic PCC is not deployed, the S-/P-GW may apply local QoS policy. This may lead to the establishment of a number of dedicated bearers for the UE following the procedures defined in clause 5.4.1 of [43] in combination with the establishment of the default bearer, which is described in Annex F of [43].

    If the S-/P-GW is not relocated, but has received the User Location Information IE and/or UE Time Zone IE and/or User CSG Information IE from the MME in step 15, the S-/P-GW shall takes into consideration these information that e.g. can be used for charging.

    If the S-/P-GW is not relocated and it has not received User Location Information IE nor UE Time Zone IE nor User CSG Information IE from the MME in step 15, nothing needs to be done in this step and downlink packets from the S-/P-GW are immediately sent on to the target eNodeB.

17. The S-/P-GW creates a new entry in its EPS bearer context table and generates a Charging Id. The new entry allows the S-/P-GW to route user plane PDUs between the E-UTRAN and the Packet Data Network, and to start charging. The way the S-/P-GW handles Charging Characteristics that it may have received is defined in [44].

The target S-/P-GW sends a Modify Bearer Response message to the target MME. The message is a response to a message sent at step 15.

If the S-/P-GW does not change, the S-/P-GW shall send one or more "end marker" packets on the old path immediately after switching the path in order to assist the reordering function in the target eNodeB.

18. This step is the same as in Section 5.5.1.2.2 of [43].

19. When the timer started in step 14 expires the source MME sends a UE Context Release Command message to the source eNodeB. The source eNodeB releases its resources related to the UE and responds with a UE Context Release Complete message. When the timer started in step 14 expires and if the source MME received the S-/P-GW change indication in the Forward Relocation Response message, it deletes the EPS bearer resources by sending Delete Session Request (Cause, LBI and EPS bearer context(s) of bearers accepted by the target S-/P-GW) message to the source S-/P-GW. Cause indicates to the S-/P-GW that the S-/P-GW changes and EPS bearer context(s) will be stored in the Transferred EPS Bearers Table at the S-/P-GW.

    The S-/P-GW acknowledges with Delete Session Response message. If ISR has been activated before this procedure, the Cause also indicates to the S-/P-GW that the source S-/P-GW shall delete the bearer resources on the other old CN by sending Delete Bearer Request message(s) to that CN node.

20. This step is the same as in Section 5.5.1.2.2 of [43] (S-GW replaced by S-/P-GW).

21. This step is the same as in Section 5.5.1.2.2 of [43] (S-GW replaced by S-/P-GW).

2.2.3 UE or MME requested PDN disconnection

The UE or MME requested PDN disconnection procedure for an E-UTRAN is depicted in Figure 2.7. UE or MME requested PDN disconnection. The procedure allows the UE to request for disconnection from one PDN. Bearers including the default bearer of this PDN shall be deleted during this procedure. The procedure also allows the MME to initiate the release of a PDN connection.

This procedure is not used to terminate the last PDN connection. The UE uses the UE-initiated Detach procedure in Section 5.3.8.2 of [43] to disconnect the last PDN connection. The MME uses the MME-initiated Detach procedure in Section 5.3.8.3 or [43] to release the last PDN connection.

The procedure is a modification of the UE or MME requested PDN Disconnection procedure specified at Section 5.10.3 of [43]. The Initiator S-/P-GW is the S-/P-GW entity which has established the PDN connection which is required to be closed. This entity can be different from the current UE's S-/P-GW because S-/P-GW relocation is also supported.

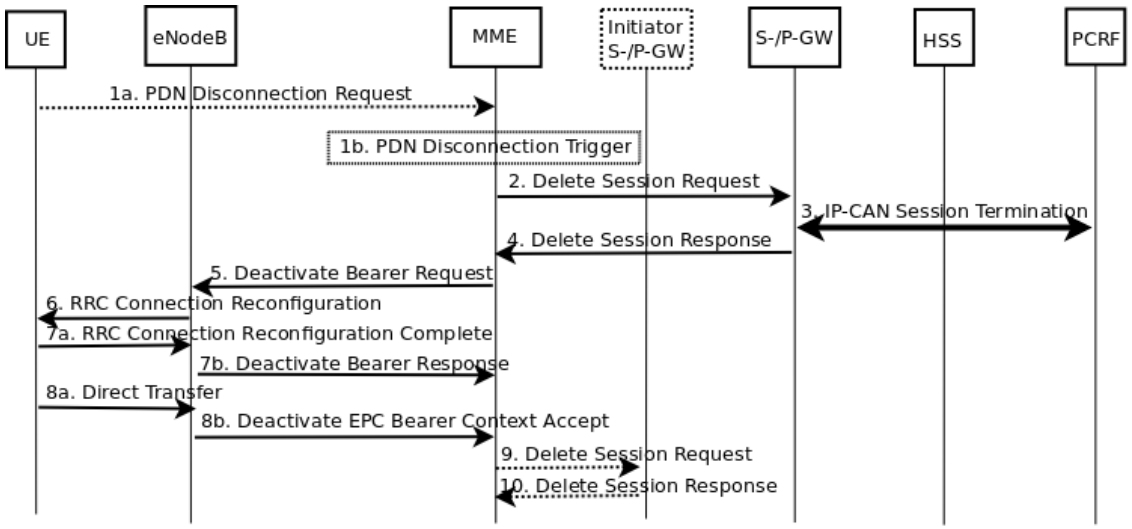


Figure 2.7: UE or MME requested PDN disconnection

The procedure is triggered by either step 1a or step 1b.

1a. The UE initiates the UE requested PDN disconnection procedure by the transmission of a PDN Disconnection Request (LBI) message. The LBI indicates the default bearer associated with the PDN connection being disconnected. If the UE was in ECM-IDLE mode, this NAS message is preceded by the Service Request procedure. Step 1b) The MME decides to release the PDN connection. This may be e.g. due to change of subscription or lack of resources.

1b. The MME decides to release the PDN connection. This may be e.g. due to change of subscription or lack of resources.

2. The EPS bearers in the S-/P-GW for the particular PDN connection are deactivated by the MME by sending Delete Session Request (Cause, LBI) to the S-/P-GW. This message includes an indication that all bearers belonging to that PDN connection shall be released. If the S-/P-GW requested UE's location info (determined from the UE context), the MME also includes the User Location Information IE in this message. If the UE Time Zone has changed, the MME includes the UE Time Zone IE in this message.

3. The S-/P-GW employs the PCEF Initiated IP-CAN Session Termination procedure as defined in [44] to indicate to the PCRF that the IP-CAN session is released if PCRF is applied in the network.

4. The S-/P-GW acknowledges with Delete Session Response.

5. This step is the same as step 7 in Section 5.10.3 of [43].

6. This step is the same as step 8 in Section 5.10.3 of [43].

7a. This step is the same as step 9a in Section 5.10.3 of [43].

7b. This step is the same as step 9b in Section 5.10.3 of [43].

8a. This step is the same as step 10a in Section 5.10.3 of [43].

8b. This step is the same as step 10b in Section 5.10.3 of [43].

9. If an entry having EPS bearer identity as the one in the LBI is found in the Initiator S-/P-GW Table the correspondent S-/P-GW address and TEID are used by the MME to send a Delete Session Request message (Cause, LBI) to Initiator S-/P-GW. A special Cause value is used indicating that Delete Session Request message is referred to a previously Transferred EPS Bearer.

   If no entry is found in the Initiator S-/P-GW then no S-/P-GW relocation had previously occurred for this PDN connection and this message will not be sent.

10. The S-/P-GW drops the correspondent EPS bearer context from the Transferred EPS Bearers Table and the associated UE IP address(es) can then be allocated for new PDN connection.
    The S-/P-GW acknowledges with Delete Session Response. The correspondent EPS bearer context is removed from the MME UE Context by the MME and the entry stored with the specific EPS bearer identity is removed from the Initiator S-/P-GW Table.

2.2.4 Summary

The modifications needed to the EPC entities and procedures in order to support the establishment of EPS bearers which are kept active also when their anchor point has been relocated, are summarized in the following:

- UE IP address(es) needs to be stored inside the MME UE context of the MME serving the specific EPS bearer. A UE IP address is sent to the MME by the Initiator S-/P-GW during the E-UTRAN Initial Attach Procedure inside a Create Session Response message (see Section 5.3.2.1 of [43]) where it replaces the PDN Type and PDN Address fields which are not used due to the fact that S5/S8 bearers are removed from the Bearer Architecture. Each IP address needs to be topologically anchored to the S-/P-GW that allocated it. A pool of addresses topologically related to its position, will be therefore instantiated at every S-/P-GW.

- UE IP address(es) of the EPS bearers kept active by the UE after handover need to be forwarded from source MME to target MME (when MME relocation occurs during handover). This information will replaces the PDN Type and PDN Address of Forward Relocation Request message. UE IP address(es) replace PDN Type and PDN Address type into the EPS bearer context and they will be then forwarded to the target S-/P-GW in a Create Session Request message. A different filed for IPv4 addresses and IPv6 address might need to be used but this specification is out of the scope of this document.

- Initiator S-/P-GW Table needs to be deployed inside the MME and used to store IP address and TEID used in signaling by the S-/P-GW entity that has established the PDN connection. The EPS bearer identity is also stored and used to retrieve the correct entry when required (primary key).

- PCEF Initiated IP-CAN Session Modification procedure needs to be triggered by the target S-/P-GW upon reception of a Modify Bearer Request from the MME. The UE IP address(es) contained in the Create Session Request message previously received from the MME needs to be used by the PCEF in this procedure.

- Transferred EPS Bearers Table needs to be deployed in all S-/P-GWs. This table is used to store EPS bearer context(s) belonging to PDN connection(s) that has been transferred to another S-/P-GW. IP address(es) used in bearers contained in this table are not to be allocated to new PDN connection. A second table can be used to store the non-routable IP addresses.

- A Delete Session Request's Cause field needs to be defined and it will be use to signal a S-/P-GW that the EPS bearer(s) belonging to the specific session need to be moved into the Transferred EPS Bearer table.

The modifications needed to the EPC entities and procedures in order to support the removal of EPS bearers which were kept active after the UE has been handed off to a new S-/P-GW domain, are summarized in the following:

- A second Delete Session Request's Cause field needs to be defined and it will be use to signal a S-/P-GW that the EPS bearer(s) belonging to the specific session need to be removed from the Transferred EPS Bearer Table.

- Delete Session Request message using the Cause field specified above is to be sent by the MME to the S-/P-GW which established the PDN connection(s). This message is sent when, upon reception of a Deactivate EPS Bearer Context Accept message from the eNodeB, the MME finds the specific S-/P-GW IP address and TEID in the Initiator S-/P-GW Table with the EPS bearer identity of the PDN connection to be disconnected as entry's key.

The introduction of an entity which embodies functionalities of both Serving and PDN GWs generates the need to modify other 3GPP standard procedures such as PDN GW and Serving GW selection functions, Initial Attach, Tracking Area Update, UE and MME initiated Detach, bearer modification procedure, Intra RAT handover among others.

Since these procedures are out of the scope of this research their modifications will not be discussed in this report.

Chapter 3

Proposed solutions

Following the requirements, functional framework and architectural design of a DMM solution for cloud based LTE systems presented in Chapter 2, two main approaches to perform traffic redirection in the transport network above the EPS will be introduced and described in this Chapter. Three solutions derived from the aforementioned approaches will be selected and compared to give a preliminary indication of their possible impacts on current operators' mobile networks.

3.1 Double NAT DMM solution

The first proposed DMM solution adopts the concept of an identifier-locator split to solve the routing in the transport network above the EPS, and it is inspired by the work of M. Liebsch in [45]. Identifier refers to the UE's IP address of the flow(s) that has been kept active after performing handover. This IP address has been allocated by the S-/P-GW where the UE was attached when the flow has been initiated. Belonging to a pool of addresses different from the one of the UE's current S-/P-GW, the identifier address is topologically unrelated with the UE's current position. For this reason the allocation of a new IP address from the current S-/P-GW's address pool is required. This address is referred to as the locator address and it is used to forward downlink packets to the UE's current position. The locator address will not be advertised to the UE, since its usage is restricted to the operator's transport network only. If the UE can support multiple-addresses and he initiates a new flow while attached to the current S-/P-GW, a new address, from the same pool as the locator address, will be allocated and provided to him.

Forwarding downlink packets to the UE's current S-/P-GW can be achieved using tunnels as already done in both MIP [10] and PMIP [11] solutions. To avoid encapsulation overhead introduced by tunneling, Network Address Translation (NAT) is used at both ends of the operator's transport network. Two new entities, performing address translation from identifier address to locator address and vice-versa, need to be introduced in the network. These entities are referred to as Ingress NAT router and

Egress NAT router.

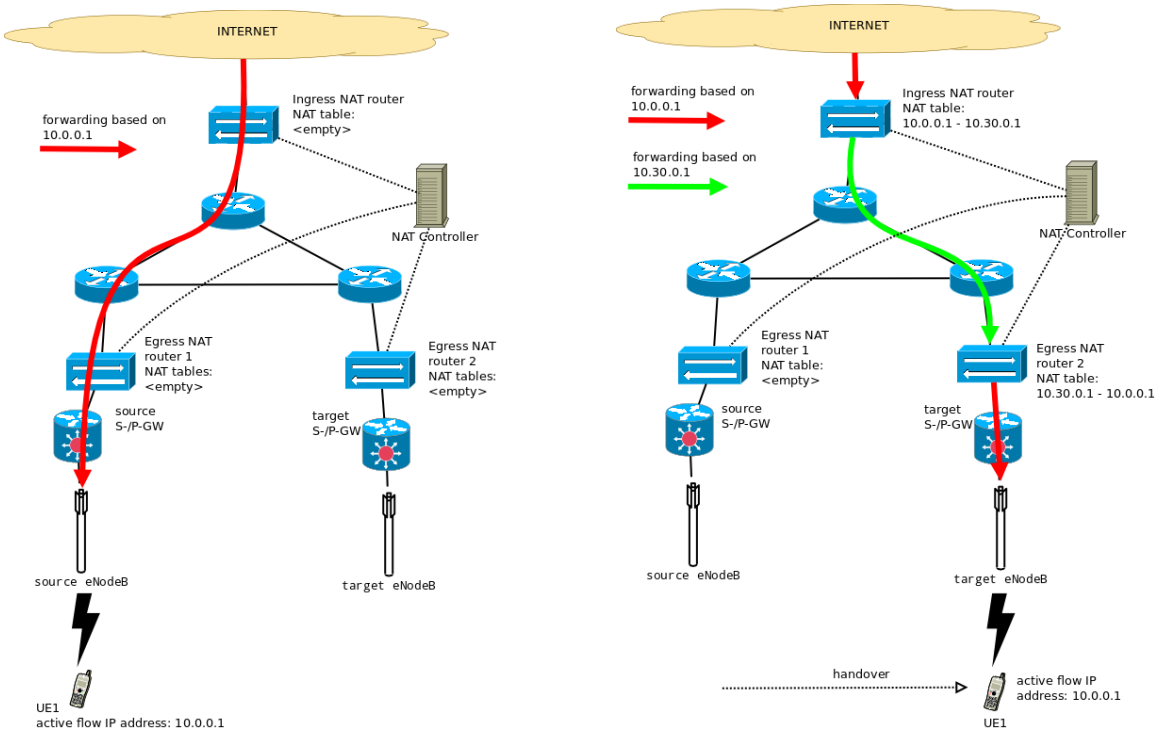
Being traffic redirection required only for downlink traffic, the Ingress NAT router performs translation of the identifier address into the locator address and it forwards the packets down into the operator's transport network. The Egress NAT router, on the other hand, translates the locator address back to the identifier address in order to forward the packet to the EPC. The Egress NAT routers will therefore always be placed closer to the southern edge of the operator's transport network than the Ingress NAT routers. More precisely the Egress NAT router needs to be placed not further away than the previous router in the downlink path towards the S-/P-GW since Layer3 routing cannot be used to forward downlink traffic after the second translation due to the fact that the identifier address is topologically unrelated with the S-/P-GW location. Alternatively the Egress NAT router can be co-located with the S-/P-GW.

Due to the fact that two address translations are required in the downlink path towards the UE to guarantee flow's IP address continuity, the described solution has been named Double NAT. When not specified differently, the procedures and signaling messages introduced in the remainder of this Section have been defined within this research.

Please note that in the remainder of this Section, since a virtualized LTE network deployment is used, an EPS entity (either S-/P-GW, MME or eNodeB) is intended as the corresponding virtualized EPC service instance component that is instantiated and running in one or more micro/macro data centers.

Figure 3.1a, gives a more detailed example of what has been described so far. UE1 first attached to the source S-/P-GW and get assigned a new IP address (10.0.0.1) from the pool of addresses belonging to this S-/P-GW. The operator's transport network is setup to route traffic directed to 10.0.0.1 to the source S-/P-GW.

When UE1 is handed over to the target eNodeB which is served by the target S-/P-GW, if one or more flows have been kept active by UE1, the DMM needs to provide IP address continuity for these flows. Inside the EPC, IP address continuity is provided by the mechanisms explained in Section 2.2. In order to deliver the flows previously initiated by UE1 to the target S-/P-GW, Double NAT can be used in the operator's transport network. IP address 10.0.0.1 is the flow's identifier address. After receiving the specific signaling from its serving MME (see Section 2.2.2), the target S-/P-GW allocates IP address 10.30.0.1 from its pool of addresses. This address will not be advertised to the UE1 and it will be used as the flow's locator address in the operator's transport network. Double NAT takes care of the correct splitting between identifier and locator addresses: downlink traffic directed to 10.0.0.1, will therefore be translated into 10.30.0.1 by one of the Ingress NAT routers and then forwarded into the operator's transport network. When packets directed to 10.30.0.1 will arrive at the SGi router of the target S-/P-GW, the Egress NAT function present on this router will translate the destination address back to 10.0.0.1 and forward it to the target S-/P-GW. After being processed by the UE's anchor point the packet will be encapsulated into a GTP tunnel and forward it to the target eNodeB which will take care of deliver it to UE1 via the LTE air interface. This procedure is shown in Figure 3.1b.



(a) before handover

(b) after handover

Figure 3.1: Double NAT data plane mechanism

3.1.1 NAT tables

In order to perform pre-routing destination NAT on the traffic downlink path, the Ingress NAT routers need to held a per-host state for the UE's IP address in their NAT table. If no per-host state is available in the table, the packet is forwarded according to a longest prefix match. If the UE has never changed his S-/P-GW, the packet will be delivered correctly, since the flow's identifier address is topologically anchored at the UE's current position.

But when the UE is handed over to an eNodeB served by a different S-/P-GW, a new NAT rule needs to be proactively entered in both Ingress and Egress NAT routers' tables in order to provide IP address continuity.

On the other hand when all the flows related to an IP address, which has required Double NAT management, are terminated, its correspondent NAT rules need to be removed since this address may now be allocated to a new UE by the S-/P-GW that allocated the IP address in the first place.

To accomplish the above an entity called NAT Controller is introduced in the network. The role of this entity is to signal all the interested NAT routers (either Ingress or Egress) in order to add or withdraw a rule from the NAT table. More than one NAT Controller can be present in a network, but a protocol is then needed to keep the NAT tables consistent throughout the network. In the following only the

case with one NAT Controller will be addressed, leaving the multi-controller case for further studies.

As already mentioned, due to the fact that addresses are topologically anchored to the entity that allocate them (S-/P-GW) there is no need to translate each UE's address into the address of the S-/P-GW where the correspondent flow has been initiated. Longest prefix match will be used in these cases while a proactive mechanism needs to be deployed to keep the Ingress and Egress NAT tables up-to-date in case that the UE changes his mobility anchor point.

As it can be deduced ports are not NATted and therefore the NAT tables of Ingress NAT routers appears as follow:

Table 3.1: Ingress router NAT table

local_ip [PRIMARY]	***global_ip***	***last_edit***
10.0.0.1/24	10.30.0.1/24	20130821T145430

- The ***local_ip*** (64 bits) is the address used to forward downlink UE's traffic in the external network (i.e. the Internet or IMS network). It is denoted as *identifier_address* by the NAT Controller. The *local_ip* is the primary key of the NAT table.
- The ***global_ip*** (64 bits) is the address used to forward downlink UE's traffic in the operator's transport network. It is denoted as *locator_address* by the NAT Controller.
- The ***last_edit*** (64 bits) is the date and time of the last modification to this specific entry in ISO8601 format.

The combination of *local_ip* and *global_ip* of a NAT table entry is defined as *NAT rule*.

An Egress NAT router can be connected to more than one S-/P-GW making the mapping of downlink traffic to the correspondent output interface more complex since after translating the locator address back to the identifier address, the longest prefix match cannot be used anymore. For this reason a slight modification is needed to be introduced in his NAT table, adding a column used to map each identifier address with the output interface as shown in Table 3.2. The output interface is determined using a longest prefix match on the entry locator address. In other words the output interface is the interface where the Egress NAT router is connected to the S-/P-GW that has allocated the entry's locator address.

Table 3.2: Egress router NAT table

local_ip [PRIMARY]	***global_ip***	***last_edit***	***output_iface***
10.0.30.1/24	10.0.0.1/24	20130821T145429	10.30.0.x/24
...	...	...	...

- The ***local_ip*** (64 bits) is the address used to forward downlink UE's traffic in the operator's transport network. It is denoted as *locator_address* by the NAT Controller. The *local_ip* is the primary key of the NAT table.
- The ***global_ip*** (64 bits) is the address used to forward downlink UE's traffic in the EPC. It is denoted as *identifier_address* by the NAT Controller.
- The ***last_edit*** (64 bits) is the date and time of the last modification to this specific entry in ISO8601 format.
- The ***output_iface*** (32 bits) is the interface used to output the incoming packets which destination IP address matches the *global_ip* entry of the table.

3.1.2 Controller-to-router signaling

A secure connection (e.g. TCP and TLS) is established between the NAT Controller and each Ingress and Egress NAT routers. Controller-to-router messages are always initiated by the NAT Controller and do not require any response from the NAT routers.

Two types of messages are used by the NAT Controller to signal the Ingress and Egress NAT routers:

- ***Rule Update*** is used to add/modify a rule into the NAT table;
- ***Rule Withdraw*** is used to remove a rule from the NAT table

The 130 bits payload of a *Rule Update* and *Rule Withdraw* messages is identical and it is as follow:

2 bits	32 bits	32 bits	64 bits
flag	*identifier_address*	*locator_address*	*timestamp*

- The ***flag*** field if set to 00 indicates that the packet carries a *Rule Withdraw* message. In all the other cases the packet carries a *Rule Update* message.
- The ***identifier_address*** field represents the identifier address of flow(s) kept active by the UE after handover.
- The ***locator_address*** field represents the new IP address allocated by the target S-/P-GW and used to route the UE's previously initiated traffic in the operator's transport network towards its current anchor point.
- The ***timestamp*** field indicates the date and time when this packet was sent. It is represented in ISO8601 format.

In order to send the signaling messages to the correct entities Ingress and Egress NAT routers information needs to be stored in the NAT Controller. Two tables are deployed for this purpose, namely *Ingress NAT Info* and *Egress NAT Info*.

Table 3.3: Ingress NAT Info

id [PRIMARY]	***ip_address***	***tcp_src_port***	***tcp_dest_port***	***state***
nat34562i_NL	192.168.90.x/24	2340	2710	Active
...	...	...	...	...

- The ***id*** (16 bits) field stores the ID used to identify the specific Ingress NAT router. This field is the primary key of the table.
- The ***ip_address*** (64 bits) field stores the destination IP address used in the secure connection between NAT Controller and Ingress NAT router.
- The ***tcp_src_port*** (16 bits) field stores the source TCP port used in the secure connection between NAT Controller and Ingress NAT router.
- The ***tcp_dest_port*** (16 bits) field stores the destination TCP port used in the secure connection between NAT Controller and Ingress NAT router.
- The ***state*** (4 bits) field stores the current state of life of the Ingress NAT router. Current valid values for this field are Active and Inactive. Inactive is used when the NAT router is currently unused due to failure or maintenance.

The *Egress NAT Info* table of the controller, also requires an extra field used to send *Rule Update* and *Rule Withdraw* messages to the correct Egress NAT router. This field is called *egress_subnet*. To discover which Egress NAT routers needs to be signaled by the NAT Controller a longest prefix match calculation is performed between the flow's locator address and all the *egress_subnet* entries in the table. Since the *egress_subnet* field contains the subnetwork used by the S-/P-GW in allocating the locator addresses, the match will always be in the entry containing information regarding the Egress NAT routers connected to that specific S-/P-GW. When more than one match is found in the *Egress NAT Info* table, the *Rule Update* or *Rule Withdraw* messages will be forwarded to all the Egress NAT routers marked as Active in the table.

The *Egress NAT Info* table is as follow:

Table 3.4: Egress NAT Info

id [PRIMARY]	***ip_address***	***tcp_src_port***	***tcp_dest_port***	***state***	***egress_subnet***
nat31082e_NL	192.168.90.y/24	2341	2711	Active	10.30.0.0/16
...	...	...	...	...	...

- The ***id*** (16 bits) field stores the ID used to identify the specific Egress NAT router. This field is the primary key of the table.
- The ***ip_address*** (64 bits) field stores the destination IP address used in the secure connection between NAT Controller and Egress NAT router.
- The ***tcp_src_port*** (16 bits) field stores the source TCP port used in the secure connection between NAT Controller and Egress NAT router.
- The ***tcp_dest_port*** (16 bits) field stores the destination TCP port used in the secure connection between NAT Controller and Egress NAT router.
- The ***state*** (4 bits) field stores the current state of life of the Egress NAT router. Current valid values for this field are Active and Inactive. Inactive is used when the NAT router is currently unused due to failure or maintenance.
- The ***egress_subnet*** (64 bits) field contains the subnetwork used by the S-/P-GW in allocating UE's IP addresses.

The protocols and procedures to add and remove entries from the NAT Controller's *Ingress* and *Egress NAT Info* tables are not in the scope of this document.

Whenever the NAT Controller sends a new *Rule Update* to one or more Ingress and Egress NAT routers, the IDs of all the signaled NAT routers are inserted in a separated table in a tuple together with the *identifier_address* transmitted in the *Rule Update* message. This table is called *Signaled Routers* table and it is shown below. This table is used by the NAT Controller whenever a *Rule Withdraw* needs to be sent in the network.

Table 3.5: Signaled Routers

identifier_address	***ip_address***
10.0.0.1	nat31082e_NL
10.0.0.1	nat34562i_NL
...	...

- The ***identifier_address*** field corresponds to the *identifier_address* used in the *Rule Update* message sent to the Ingress or Egress NAT routers with *id* specified in the next column.
- The ***id*** (16 bits) field stores the ID used to identify the specific Ingress or Egress NAT routers which have been signaled with a *Rule Update* message having as *identifier_address* the value specified in the previous column.

3.1.3 MME-to-controller signaling

In order to signal the entitled entities (Ingress NAT routers and Egress NAT router) three type of information are needed by the NAT Controller:

1. flow's identifier address(es)
2. flow's locator address
3. Egress NAT router IP address

Two EPC entities have knowledge of the information required by the NAT Controller after the completion of the handover procedure by the UE, target S-/P-GW and its serving MME.

Being a control plane entity the serving MME has been selected as the peer entitled to signal the NAT Controller when DMM traffic redirection is required. Furthermore one MME can serve multiple S-/P-GW reducing the amount of active connections needed between the EPC and the NAT Controller.

A secure connection (e.g. TCP and TLS) will be established between each MME and the NAT Controller and it is used to forward specific signaling packets that will be introduced in the following. MME-to-controller messages are always initiated by the MME and do not require any response from the NAT Controller.

Two types of messages are used by the MME to signal the NAT Controller:

- ***Make Path***: Establishes a downlink path in the operator's transport network using Ingress and Egress NAT functionalities;
- ***Tear Down Path***: Remove a previously established downlink path in the operator's transport network.

The payload of both *Make Path* and *Tear Down Path* messages is as follow:

32 bits	32 bits	2 bits
previous_address	*current_address*	*flag*

- The ***previous_address*** field represents the identifier address of flow(s) kept active by the UE after handover. This information is already contained in the UE Context stored in the MME (see Section 2.2.2)
- The ***current_address*** field represents the new IP address allocated by the target S-/P-GW. This information is given to the MME by the target S-/P-GW using a Modify Bearer Messages (as specified in Section 2.2.2).
- The ***flag*** field if set to 00 indicates that the packet carries a *Tear Down Path* message. In all the other cases the packet carries a *Make Path* message.

In order for the MME to send the signaling messages to the NAT Controller a table is deployed for this purpose, namely NAT Controller Info table. Having currently deployed only one NAT Controller in the system this table will have only one entry.

The *NAT Controller Info* table is as follow:

Table 3.6: NAT Controller Info

id [PRIMARY]	***ip_address***	***tcp_src_port***	***tcp_dest_port***	***state***
nat43891c_NL	192.168.90.z/24	2342	2712	Active
...	...	...	...	...

- The ***id*** (16 bits) field stores the ID used to identify the specific NAT Controller. This field is the primary key of the table.
- The ***ip_address*** (64 bits) field stores the destination IP address used in the secure connection between NAT Controller and MME.
- The ***tcp_src_port*** (16 bits) field stores the source TCP port used in the secure connection between NAT Controller and MME.
- The ***tcp_dest_port*** (16 bits) field stores the destination TCP port used in the secure connection between NAT Controller and MME.
- The ***state*** (4 bits) field stores the current state of life of the NAT Controller. Current valid values for this field are Active and Inactive. Inactive is used when the NAT Controller is current unused due to failure or maintenance.

The protocols and procedures to add and remove entries from the NAT Controller's *Ingress* and *Egress NAT Info* tables are not in the scope of this document.

3.1.4 Message Flow

Having already introduced the signaling messages required to deploy IP address continuity and traffic redirection in the operator's transport network, Figure 3.2 shows the message flows needed to setup a new downlink path towards the UE's current S-/P-GW (target S-/P-GW) in the operator's transport network.

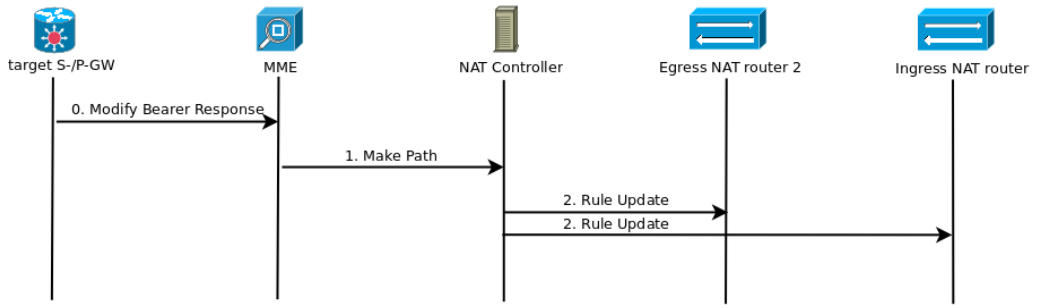


Figure 3.2: Message flow to establish a new downlink path upon reception of a Modify Bearer Response message during handover with S-/P-GW relocation procedure

0. This step is the same as step 5 in Section 2.2.1 and step 17 in Section 2.2.2.

1. The MME creates a *Make Path* message (*flag* field sets to either 01, 10 or 11 value) setting as the *current_address* field the IP address extracted from the Modify Bearer Response message received from the target S-/P-GW. By looking in the UE Context of the UE in object, the MME can retrieve the IP address belonging to flow(s) kept active by the UE after handover. This address is set as the *previous_address* field in the *Make Path* message. If more than one active IP addresses are found in the UE Context, the MME duplicates the *Make Path* message created so far and set the *previous_address* field accordingly. Each message is then encapsulated in a TCP/IP packet using the information retrieved from the *NAT Controller Info* table and then forwarded to the NAT Controller.

2. When a message is received from the MME the NAT Controller looks at the *flag* field to discover whether is a *Make Path* or a *Tear Down Path* message. In this case a *Rule Update* message is created by the NAT Controller setting the *flag* field to either 01, 10 or 11. The *previous_address* (10.0.0.1) and *current_address* (10.30.0.1) fields of the *Make Path* message are copied respectively into the *identifier_address* and *locator_address* field of the *Rule Update* message. The current date and wall-clock value are translated into ISO8061 format and set as the *timestamp* field of the *Rule Update* message.
   For every entry marked as Active in the *Ingress NAT Info* table, the NAT Controller duplicates the *Rule Update* message and uses the values of the entry in object to encapsulate the duplicated *Rule Update* message in a TCP/IP packet. The *identifier_address* of the message and the *id* of the active entry in the *Ingress NAT Info* table composes a tuple which is then inserted in the *Signaled Routers* table.
   If a longest prefix match is found between the *locator_address* and the *egress_subnet* entries of the *Egress NAT Info* table, the entry where the matching *egress_subnet* value belong is used to encapsulate the original *Rule Update* message in a TCP/IP packet. If no match has been found by the longest prefix match calculation: for every entry marked as Active in the *Egress NAT*

Info table, the NAT Controller duplicates the *Rule Update* message and uses the values of the entry in object to encapsulate the duplicated *Rule Update* message in a TCP/IP packet. The *identifier_address* of the message and the *id* of the matching entry (or active entries if no match is found) in the *Egress NAT Info* table composes a tuple which is then inserted in the *Signaled Routers* table. All the packets are then forwarded to the entitled entities.

The case when *Rule Update* messages are received with a substantial advance at the Ingress DMM routers than at the Egress DMM routers can cause the unfortunate situation where Egress NAT tables have not been updated while downlink data packets of IP flows belonging to the moving UE(s) are being already NATted at the Ingress NAT routers. These downlink data packets will be then delivered to the target S-/P-GW carrying the wrong IP address and they will therefore be dropped.
To avoid the above, *Rule Update* messages directed to the Ingress NAT routers can be transmitted with a safety delay than the one directed to Egress NAT routers. The value of this safety delay is dependent on the network topology deployment.

Upon reception of a *Rule Update* the Ingress NAT router will look up its NAT table for an entry whose *local_ip* matches the *identifier_address* contained in the packet payload. If an entry is found and the *timestamp* value of the *Rule Update* packet is higher than the *last_edit* value of the entry, the entry is withdrawn and the new rule is added to the table setting *timestamp* as its *last_edit* value. If an entry is not found than the new rule is added directly setting *timestamp* as its *last_edit* value. If an entry is found but its *last_edit* is higher than *timestamp*, then no changes will occur to the NAT table.

When an Egress NAT router receives a *Rule Update* message from the NAT Controller, it will look up its table for an entry whose *local_ip* matches the *locator_address*. If an entry is found and the *timestamp* value of the received packet is higher than the entry's *last_edit* value, that entry is withdrawn and the new rule is added to the table setting *timestamp* as its *last_edit* value. If an entry is not found then the new rule is added directly setting *timestamp* as its *last_edit* value. If an entry is found but its *last_edit* is higher than *timestamp*, the new rule is then dropped. When a new rule is to be added to the NAT table, a longest prefix match calculation is run on the *locator_address*. The resulting interface will be set as the entry's output_iface. This can be achieved due to the fact that locator addresses are topologically anchored to the S-/P-GW that allocate them.

Using the example described previously and depicted in Figure 3.1 the *Make Path* and *Rule Update* messages respectively are as follows:

Make Path:

10.0.0.1	10.30.0.1	10

Rule Update:

11	10.0.0.1	10.30.0.1	20130821T145429

After receiving and processing the *Rule Update* messages the Ingress and Egress NAT routers table respectively looks as Table 3.1 and Table 3.2 (defined above).

Figure 3.3 shows the message flows needed to remove the previously established downlink path.

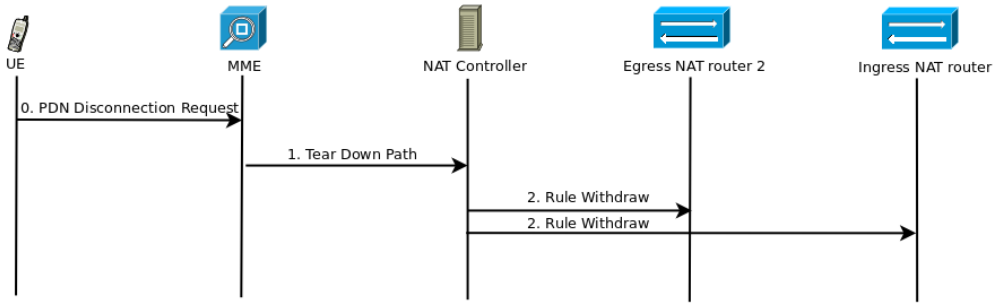


Figure 3.3: Message flow to remove a downlink path upon reception of PDN Disconnection Request from the UE

0. This step is the same as step 1a in Section 2.2.3.

1. The MME creates a *Tear Down Path* message (*flag* field sets to 00). Current_address and Previous_address messages are extracted from the default bearer indicated by the PDN Disconnection Request (LBI) message received from the UE and from its UE Context. The *Tear Down Path* message is then encapsulated in a TCP/IP packet using the information retrieved from the *NAT Controller Info* table and then forwarded to the NAT Controller.

2. When a message is received from the MME the NAT Controller looks at the *flag* field to discover whether is a *Make Path* or a *Tear Down Path* message. In this case a *Rule Withdraw* message is created by the NAT Controller setting the *flag* field to 00. The *previous_address* and *current_address* fields of the *Make Path* message are copied respectively into the *identifier_address* and *locator_address* field of the *Rule Update* message. The current date and wall-clock value are translated into ISO8061 format and set as the *timestamp* field of the *Rule Withdraw* message. A lookup is performed in the *Signaled Routers* table. For every entry whose *identifier_address* is equal to the *identifier_address* set in the *Rule Withdraw* message, its peer ID value is used to retrieve TCP/IP info from the *Ingress NAT Info* table. If a match is found and the entry is marked as Active, the NAT Controller duplicates the *Rule Withdraw* message and uses the values of the entry in object to encapsulate the duplicated *Rule Withdraw* message in a TCP/IP packet.
   If no match has been found in the *Ingress NAT Info* table the same procedure is repeated for the *Egress NAT Info* table. If a match is found and the entry is marked as Active, the NAT Controller duplicates the *Rule Withdraw* message and uses the values of the entry in object to encapsulate the duplicated *Rule Withdraw* message in a TCP/IP packet.
   Whenever a match is found in the *Ingress NAT Info* or in the *Egress NAT Info* table, then the

tuple is removed from the *Signaled Routers* table. All the packets are then forwarded to the entitled entities.

When receiving a *Rule Withdraw* the Ingress NAT routers will look up their table for an entry whose *local_ip* matches the *identifier_address*. If an entry is found and *timestamp* is higher than the entry's *last_edit*, that entry is withdrawn from the table. If an entry is not found or *timestamp* contained in the receive packet is lower than the entry's *last_edit*, then no changes will occur to the NAT table.

Upon reception of a *Rule Withdraw* the Egress NAT router will look up his table for an entry whose *local_ip* matches the *locator_address*. If an entry is found and the *timestamp* contained in the packet's payload is higher that the entry's *last_edit* value, the rule is withdrawn from the table. If an entry is not found or *timestamp* is lower than the entry's *last_edit* value, then no changes will occur to the NAT table.

Using the example described previously and depicted in Figure 3.1, the *Tear Down Path* and *Rule Withdraw* messages respectively are as follows:

Tear Down Path:

10.0.0.1	10.30.0.1	00

Rule Withdraw:

00	10.0.0.1	10.30.0.1	20130822T160100

After receiving and processing the *Rule Withdraw* messages the entry previously added in both Ingress and Egress NAT router's table will be removed.

3.1.5 Challenges

NAT is a well-known and used procedure in the modern Internet network. Integrating NAT functionalities into several operator's transport network routers is therefore a trivial operation. Handling the tables to maintain IP address continuity and traffic redirection above the EPS layer can be performed using a basic signaling protocol as the one described above. The main challenge that can be identified for the Double NAT solution are the placement of the NAT entities (controller(s), Ingress and Egress routers) in the operator's network. If a private separated network is used between the signaling entities, the problem is reduced to a rather simple private network topology design issue. Being the signaling load very small compared to the traffic, in the remainder of this report the signaling messages are assumed to be transported using the same operator's data traffic transport network.

As discussed previously, for the Egress NAT routers the problem regarding their positioning does not present itself, since they must be placed as the previous hop in the downlink path towards each S-/P-GW entity. The specific network topology might require that one Egress NAT router is connected to more than one S-/P-GW. Routing traffic towards the correct mobility anchor point cannot be based on

longest prefix match calculation when the flow's IP address has not be allocated by the current UE's S-/P-GW. This issue has been solved by introducing the *output_iface* field in the Egress NAT table.

Although positioning the Ingress NAT routers as close as possible to the operator's Internet or IMS Point of Presences (PoPs) seems the most logical solution, there might be some network topologies that take advantages of having a lower hop count between Ingress and Egress NAT routers.

The placement of the NAT Controller in the operator's network can be influenced by both EPC deployment and Ingress NAT routers positioning. A trade-off between placing the controller too close to the MME or too close to the Ingress NAT routers needs to be determined.

3.2 OpenFlow-based DMM solution

OpenFlow [46] is a communication protocol that has attracted the interest of both industry and academia. With OpenFlow the forwarding plane of a network switch or router can be accessed over the network and modified according to the needs. The vast majority of Ethernet switches and routers used nowadays contains flow-tables to implement firewall, NAT, QoS and other functionalities. A flow-table of an OpenFlow-enabled switch or router can be remotely programmed partitioning the network's traffic into separated flows.

Although mainly developed to run experimental protocols in everyday networks using so-called *research flows* (counterpart of the common *production flows*), the features offered by OpenFlow can be used to deploy a DMM solution offering IP address continuity and traffic redirection in the operator's transport network. This can be achieved by treating each traffic path from the PoPs to the S-/P-GWs as a separated flow. In this way traffic can be re-routed to the new mobility anchor point without involving any IP address translation/modification.

Alternatively OpenFlow switches contains a list of actions that can be applied on every transiting packet belonging to a specific flow. Example of these actions are: *Drop, Push-Tag, Pop-Tag, Group* and *Set-Field*. The optional *Set-Field* action is the most interesting for the purposes of this research giving to the OpenFlow switches the possibility to modify packet headers such as Ethernet, VLAN, MPLS and IP among the others.

Both flow tables and action list are added and removed by the OpenFlow Controller which has a dedicated secure connection with each OpenFlow routers and switches. The procedures and messages on how to perform such modifications are specified in the OpenFlow specification document [47].

Two different DMM solutions can be deployed using the set of features provided by OpenFlow. They will be both described and analyzed in detail in the following subsections. When not specified differently, the procedures and signaling messages introduced in the remainder of this Section have been defined within this research.

Please note that in the following parts of this Section, since a virtualized LTE network deployment is

applied, an EPS entity (either S-/P-GW, MME or eNodeB) refers to the corresponding virtualized EPC service instance component that is instantiated and running in one or more micro/macro data centers.

3.2.1 Full OpenFlow transport network

The first solution proposes the use of flow tables updates to re-route the downlink traffic towards the UE's current S-/P-GW. Due to the fact that no modification will be performed on the data packets, traffic redirection can be performed only if all routers and switches in the operator's transport network are OpenFlow enabled.

When a UE changes his EPC mobility anchor point after being handed over to a target eNodeB, flow tables of all routers and switches will be updated to route the downlink traffic to the current S-/P-GW. For uplink traffic, a static forwarding path will be setup when a flow is created. If a path to the specific CN already exists, for instance in case of widely visited end-hosts, no changes will be needed to the setup paths. This static path will be used throughout the whole life of a flow in the operator's transport network.

In Figure 3.4 an example of a simple full OpenFlow network, used as transport network above the EPS, is shown. All the routers of the operator's transport network above the EPS are OpenFlow enabled and their flow tables are managed by the same entity (OpenFlow Controller). Input and output interfaces of every switches are marked in blue and they will be used in the following examples.

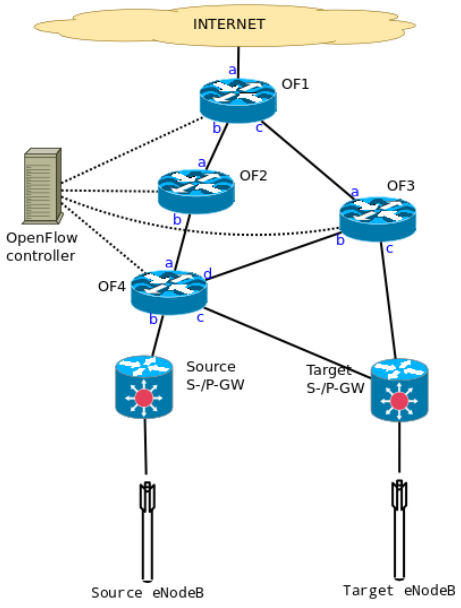


Figure 3.4: Full OpenFlow transport network

When a UE (UE1) attaches to the access network, it gets a new IP address allocated from the source S-/P-GW (e.g. 10.0.0.1). Only if all the OpenFlow switches (OF1, OF2, OF3 and OF4) are *OpenFlow-hybrid* (i.e. they support both OpenFlow operation and normal router operation) then the traffic can be

routed based on L3 routing. If this is not the case switches are called *OpenFlow-only* and the incoming traffic always needs to go through the switches flow-tables pipeline.

Due to the lack of L3 routing functionalities, the downlink paths towards the source S-/P-GW for the traffic directed to IP address 10.0.0.1 needs to be setup at runtime. New OpenFlow tables entries will be added to each switch by the OpenFlow Controller. Two different approaches can be used in this case: *per-flow forwarding* or *per-anchor forwarding*.

3.2.1.1 Per-user flow forwarding

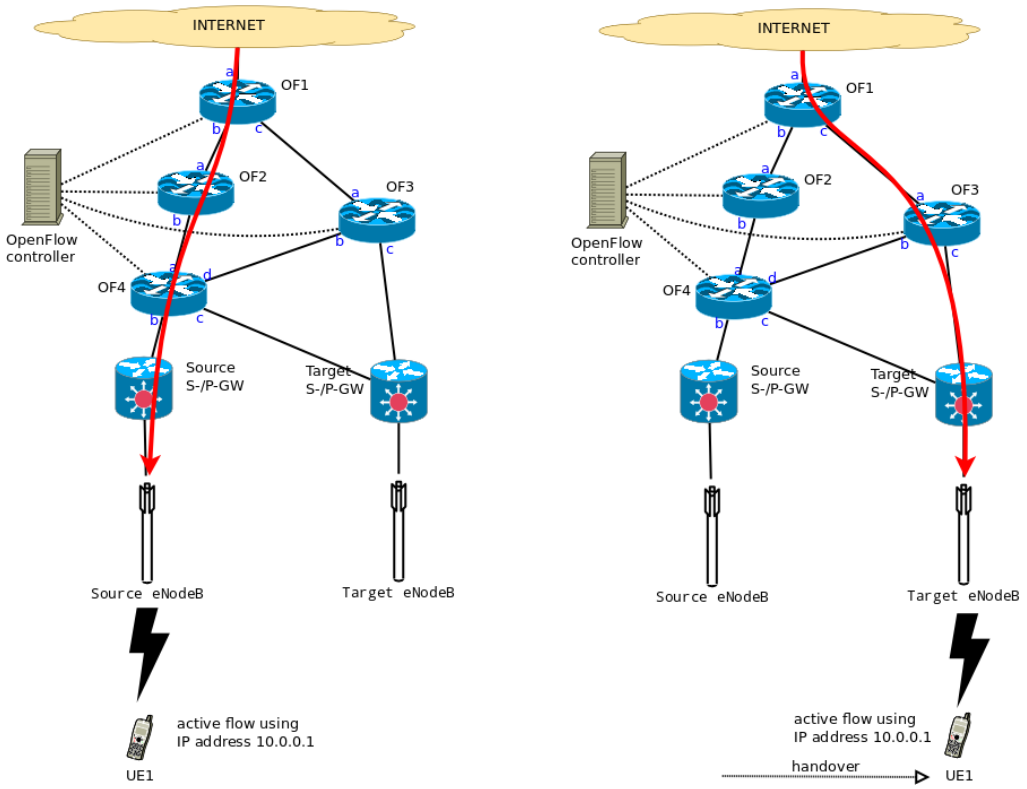
The OpenFlow Controller setups per-user's flow route towards their current S-/P-GW. In the example network depicted in Figure 3.5a, a specific route will be established by adding to the OpenFlow switches an action which will specify to which output port packets belonging to this flow have to be routed. This *Output* action will be added to the action set of the flow(s) belonging to address 10.0.0.1. Due to the different interfaces that a switch can have, this instruction needs to be created on switch granularity by the OpenFlow Controller. For instance, for flows belonging to UE1, the OpenFlow Controller can decide to route traffic via interfaces *b* of OF1, OF2 and OF4. The OpenFlow Controller will send a *Modify-State* [47] message to OF1, OF2 and OF4, adding the correct Output action to the action set of the flow having 10.0.0.1 as IP address destination.

Therefore, if UE1 has initiated a flow towards IP address 10.74.191.85, when OF1 receives downlink packets directed to UE1, it will peek the IP header of the packet. If a match is found in his first flow table using the triple <ingress port: a; IP source address: 10.74.191.85; IP destination address: 10.0.0.1> then the switch will execute the actions present in the action set of this flow. If no match is found in the first table or the action related to the flow is a *go to* instruction, then the packet will be processed by the following table in the switch pipeline. For the sake of this example, OF1 found a match in his first flow table. The instruction related to the flow is an *Output* action, setting interface *b* as the output port where to send the received packet. The same procedure is repeated at OF2 and OF4 and the traffic is routed to the source S-/P-GW via the red path showed in Figure 3.5a.

When a UE is handed over from source to target eNodeB, his EPC anchor point is also changed.

If UEs wish to keep part or all of their traffic active, IP address continuity needs to be provided in both EPS and transport network above it.

In the operator's transport network, OpenFlow switches' *Output* actions need to be modified in order to route traffic to the correct target S-/P-GW. The OpenFlow Controller will send to each interested switch a *Modify-State* message to modify the action set of the interested flow accordingly. The results of such operations are shown in Figure 3.5b where the flow kept active by UE1 after handover completion is routed via interfaces *c* of both OF1 and OF3. In this case *Modify-State* messages are also sent by the controller to both OF2 and OF4 to remove the previously added flow from their flow-tables.



(a) before handover

(b) after handover

Figure 3.5: Full OpenFlow transport network: downlink per-user flow forwarding

3.2.1.2 Per-anchor point flow forwarding

The OpenFlow Controller setups per-anchor point route towards the EPC. The approach used to route the packets is identical to the per-flow forwarding case, using *Output* instructions in all the switches. But to avoid the need to setup a number of paths equal or higher (in case UE's has multiple IP addresses and/or multiple active flows per address) than the number of UEs attached to a S-/P-GW, traffic can be forwarded based on the S-/P-GW IP or MAC address instead.

Similarly to what has been explained in the DoubleNAT solution, three group of switches can be defined: Ingress, Egress and transport. Ingress switches are the downlink first hop switches in the operator's transport network. They are directly connected to the external networks (Internet, IMS etc). Egress switches are the downlink last hop switches in the operator's transport network. They are connected with the S-/P-GW entity via the SGi interface. All the other switches belonging to the operator's transport network are referred to as transport switches. For going back to the example depicted in Figure 3.5a, OF1 is an Ingress switch, OF3 and OF4 are Egress switches and OF2 is a transport switch.

For traffic forwarding purposes, transport switches will always have *Output* instructions in their flows'

action sets. *Output* instructions will also be present in the flows' action sets of both Ingress and Egress switches but they will always be preceded (in the case of Ingress switches) or succeeded (in the case of Egress switches) in the set by a *Set-Field* action. As mentioned previously the *Set-Field* instruction is used to replace fields from different type of packet headers.

Exploiting this functionality, Ingress switches can replace the IP source and destination addresses of a packet with a unique IP destination address used to indicate traffic which has to be routed towards the target S-/P-GW. This address can be, for instance, the IP address of the SGi interface of the correspondent S-/P-GW or a special address used only inside the operator's transport network (this is possible because L3 routing is not available). Alternatively the IP header can be left untouched and the Ethernet header can be changed instead. In this case the MAC address of one of the S-/P-GW's NIC can replace the destination MAC address or once again a special MAC address can be used instead.

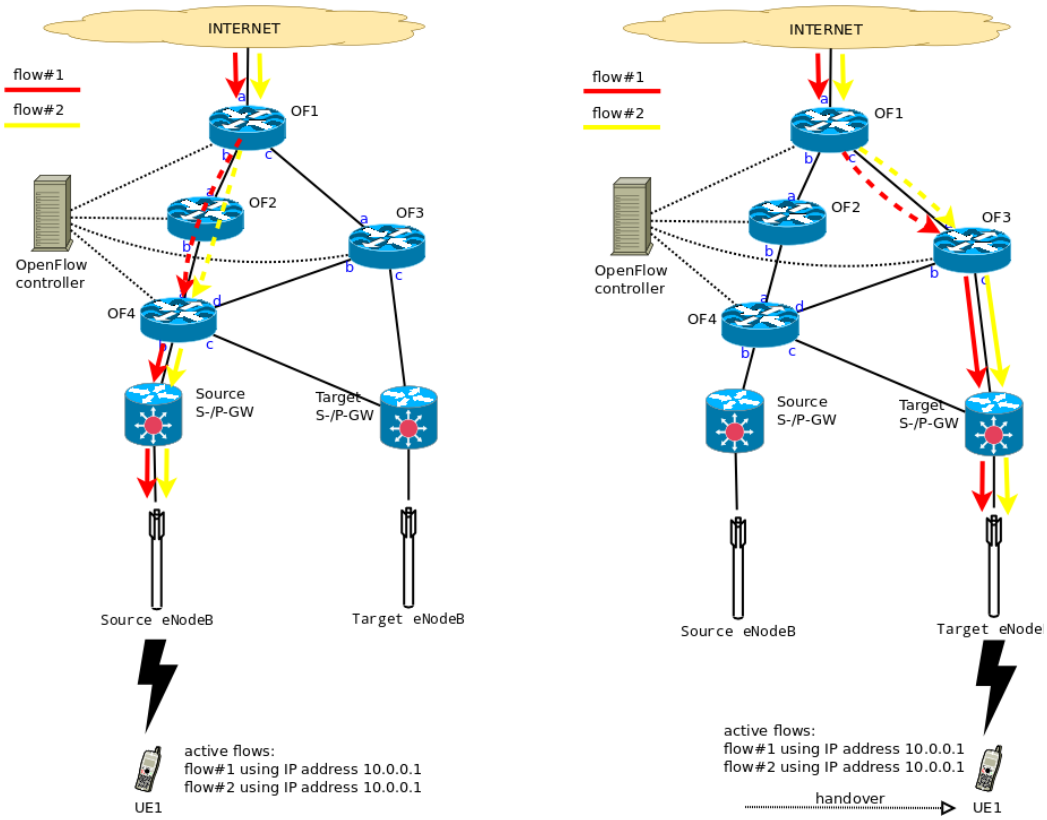
The *Set-Field* action used in the Ingress switches has a peer action which is used in the Egress switches. This peer action is also a *Set-Field* instruction used to reverse the modified packet header to its original. In the Egress switches this instruction follows the *Output* action in the flow's action set because one Egress switch can be connected to more than one S-/P-GW.

By using the example network depicted in Figure 3.4, when UE1 attaches to the access network via source eNodeB, he gets allocated IP address 10.0.0.1 from the S-/P-GW. This IP address is used in more than one flow by UE1, for instance flow#1 with CN having IP address 10.74.191.85 and flow#2 with a CN having IP address 172.31.45.68. The OpenFlow Controller will instruct the *Set-Field* action in OF1 to translate destination IP address 10.0.0.1 into IP address 192.170.0.91 which is the SGi address of the source S-/P-GW. This *Set-Field* action is added to the action set of both flow#1 and flow#2 in OF1.

After being passed via OF1 and undergo the *Set-Field* action, packets belonging to flow#1 and packets belonging to flow#2 can both be seen as packets belonging to a special flow indicated as flow#sourceSPGW. Being this flow routed via static path, predefined by the OpenFlow Controller, there is no need to add or modify *Output* actions into the Ingress, Egress and transport switches flows' action sets.

The *Set-Field* action of the Egress switch OF4 is instruct to translate SGi address 192.170.0.91 back to the original destination address 10.0.0.1. This instruction is added to the action set of both flow#1 and flow#2 entry in OF4's flow-tables.

In order to distinguish packets belonging to flow#1 from packets belonging to flow#2, flow matching in the Egress switches is based on the combination of source IP address with the used transport layer ports whether in all the other switches flow matching will be based on the destination IP address of the incoming packet. Detailed instructions on flow matching can be found at Section 5.3 of [47].



(a) before handover

(b) after handover

Figure 3.6: Full OpenFlow transport network: downlink per-anchor point flow forwarding

The downlink paths for both flow#1 and flow#2 are shown in Figure 3.6a, where the solid lines represent the paths where the flow packets' IP header are not changed, while in the dashed-line paths packets travel carrying IP address 192.170.0.91 as IP destination address. *Set-Field* actions are added to a switch's flow action set by the controller using a *Modify-State* message as defined in [47].

When a UE is handed over from source to target eNodeB, his EPC anchor point is also changed. If UEs wish to keep part or all of their traffic active, IP address continuity needs to be provided in the EPS and the transport network above it. In the operator's transport network, only Ingress and Egress switches *Set-Field* actions will need to be added or modified since the path towards the target S-/P-GW will have been already setup in the transport network.

Using the same example as above, if UE1 moves to target eNodeB served by target S-/P-GW and wishes to keep both flow#1 and flow#2 active, the *Set-Field* action of Ingress switch OF1 needs to be modified to replace IP destination address 10.0.0.1 into target S-/P-GW's SGi address 173.45.89.12. This action applies to packets belonging to both flow#1 and flow#2. The destination IP address of incoming packets belonging to both flow#1 and flow#2 (which after passed via OF1 can both be seen as packets belonging to a special flow indicated as flow#targetSPGW) will then be replaced back with

the original 10.0.0.1 address in the Egress switch OF3. A *Set-Field* action for flow#1 and one for flow#2 have been setup in OF3 by the OpenFlow Controller.

The downlink paths for both flow#1 and flow#2 after that UE1 has been handed over to target eNodeB are shown in Figure 3.6b.

3.2.2 Partial OpenFlow transport network

This solution exploits the *Set-Field* action present in OpenFlow switches. For this reason it is very similar to the *Per-anchor point flow forwarding* solution introduced previously. The main difference from that solution is that L3 routing is used in the operator's transport network instead of flow forwarding. This is possible because the operator's transport network will not be fully composed by *OpenFlow-only* switches as it was in the network topologies used so far in this Section. An hybrid network model is used, where OpenFlow-aware switches (either *OpenFlow-full* or *OpenFlow-hybrid*) are placed at the downlink ingress and egress point of the network (the same position as in Full OpenFlow transport network), while normal IP routers (or *OpenFlow-hybrid* switches) are used to transport traffic from the ingress to the egress point of the network.

An example of this partial OpenFlow transport network is shown in Figure 3.7. In this network OpenFlow-enabled Ingress and Egress switches are managed by the same controller.

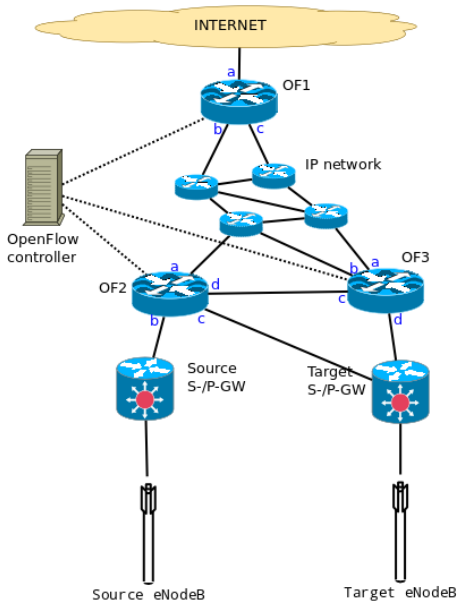


Figure 3.7: Partial OpenFlow transport network

When a UE attaches to the access network for the first time, he gets an IP address allocated from the source S-/P-GW. This IP address is used in flow(s) initiated when the UE was attached to the mobility anchor point. To provide IP address continuity when a UE changes his EPC anchor point while keeping some flows active, inside the operator's transport network a different IP addresses will be used to

route traffic to the target S-/P-GW. This address can be, for instance, the SGi address of the current UE's mobility anchor point. This case will be referred to as *per-anchor point routing* scheme since the same address will be used in all the flows anchored at a S-/P-GW. Ingress and Egress switches will take care of replacing the original destination IP address with the SGi address using the *Set-Field* action as it has been explained previously in Section 3.2.1.2. *Output* actions are needed in both Ingress and Egress routers to forward incoming downlink traffic to the transport network and EPC respectively. Flow matching in the Egress switches is based on the combination of source IP address with the used transport protocol ports.

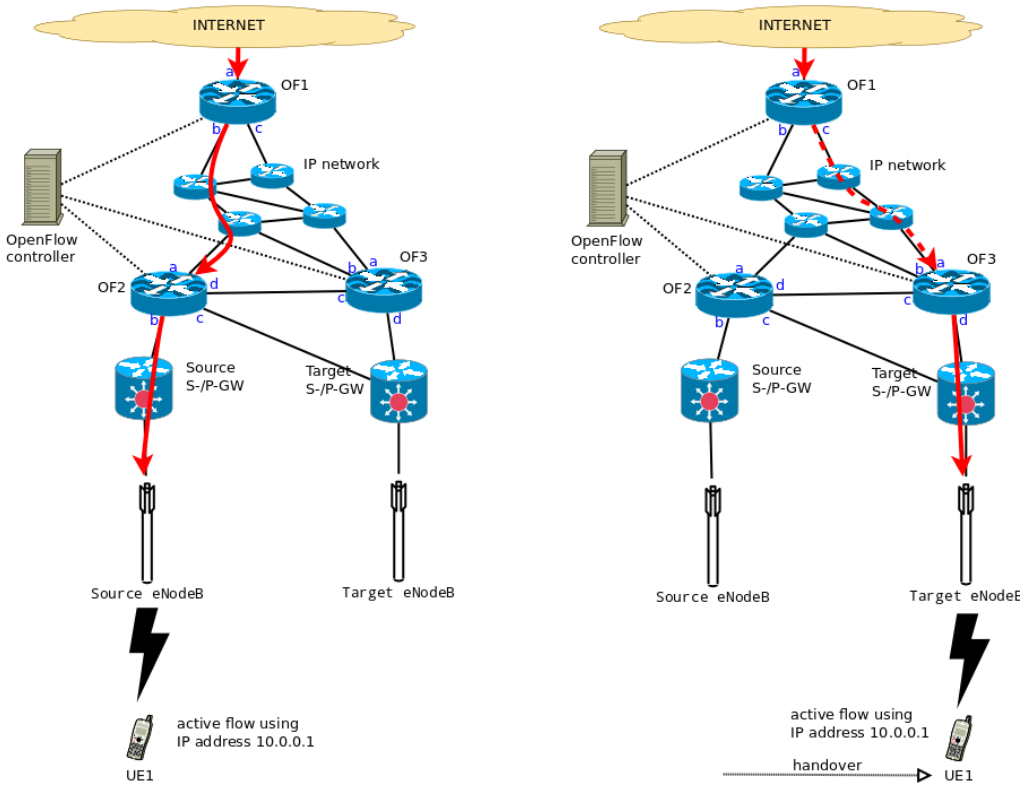
If operators wish to spread the traffic inside their network, specially allocated addresses can be used to route either uplink or downlink traffic. Being IP address continuity a problem concerning downlink traffic, the following will focus on this particular case. These specially allocated addresses need to be topologically anchored to the current UE's S-/P-GW and therefore they can be allocated from the anchor point's pool of addresses. Multiple flows of different UEs anchored at the same S-/P-GW can be routed using the same IP address with the advantage that multiple paths towards a single S-/P-GW may exist in the operator's transport network.

Following this way of reasoning, if flows initiated with different IP addresses are routed using different IP addresses in the operator's transport network, a *per-user routing* scheme will be created.

The choice on whether scheme is better is left to operators since the usage of one scheme over the other is just a matter on how the OpenFlow Controller has been setup. Since all schemes require the same set of operations to establish the correct downlink path, in the following example a controller implementing a *per-anchor point routing* scheme will be used. The same network as the one depicted in Figure 3.7 is used in the example.

When UE1, which was previously attached to the source S-/P-GW and had initiated flow#1 using IP address 10.0.0.1 (see Figure 3.8a), is handed over to the target S-/P-GW and wishes to keep flow#1 active, *Set-Field* instructions need to be added to the action set of flow#1 in the flow tables of Ingress switch OF1 and Egress switch OF3. The *Set-Field* instruction in the OF1 replace the IP destination address 10.0.0.1 with the destination S-/P-GW's SGi address (for instance 172.90.102.2). Consequently the *Set-Field* instruction in OF3 replace the SGi address with the original IP address 10.0.0.1. *Output* actions are also needed in both Ingress and Egress router. In OF1 *Output* action will forward incoming downlink traffic belonging to flow#1 via interface *c* and it will be placed after the *Set-Field* operation in the switch's flow's action set. In OF3 incoming downlink traffic will be forwarded via interface *d* but the specific *Output* action will be placed before the *Set-Field* operation in the switch's flow's action set.

Figure 3.8b shows the downlink path for flow#1 after that UE1 has been handed over to target eNodeB. The solid line indicates paths where IP 10.0.0.1 is used for routing, while in the dashed-line paths the SGi address is used.



(a) before handover

(b) after handover

Figure 3.8: Partial OpenFlow transport network downlink flow forwarding

It is important to note that a S-/P-GW's SGi address can be used to route traffic belonging to multiple flows anchored at target S-/P-GW leading to a less complex and more scalable flows management.

3.2.3 Signaling

Signaling messages and procedures between OpenFlow Controller and Ingress, Egress and transport OpenFlow-enabled switches are specified in [47] and they will not be covered by this document.

To setup *Output* and *Set-Field* actions in the operator's transport network, the OpenFlow Controller necessitate some information from the EPC. As for the Double NAT solution, the MME has been selected as the entity communicating with the control plane part of the operator's transport network (i.e. OpenFlow Controller).

Secure connections (e.g. TCP and TLS) are established between the controller and the MMEs. If more OpenFlow Controllers are present in the network then a connection to each MME will need to be setup for all of them. Support to multiple controllers is specified in Section 6.3.4 of [47].

MME-to-controller messages are initiated by the MME only and they do not require any answer from the OpenFlow Controller. Either if the network is fully OpenFlow enabled or partially OpenFlow

enabled, the message provided by the MME to the controller is identical to what have been defined for the Double NAT case (see Section 3.1.3).

Two types of messages are used by the MME to signal the OpenFlow Controller(s):

- ***Make Path***: establishes a downlink path in the operator's transport network using Ingress and Egress switches functionalities;
- ***Tear Down Path***: removes a previously established downlink path in the operator's transport network.

The payload of both Make Path and Tear Down Path messages is as follow:

32 bits	32 bits	2 bits
previous_address	*current_address*	*flag*

- The ***previous_address*** field represents the identifier address of flow(s) kept active by the UE after handover. This information is already contained in the UE Context stored in the MME (see Section 2.2.2).
- The ***current_address*** field represents the new IP address allocated by the target S-/P-GW. This information is given to the MME by the target S-/P-GW using a Modify Bearer Messages (as specified in Section 2.2.2).
- The ***flag*** field if set to 00 indicates that the packet carries a *Tear Down Path* message. In all the other cases the packet carries a *Make Path* message.

In order for the MME to send the signaling messages to the OpenFlow Controllers a table is deployed for this purpose, namely *OpenFlow Controllers Info* table.

The *OpenFlow Controllers Info* table is as follow:

Table 3.7: OpenFlow Controller Info

id [PRIMARY]	***ip_address***	***tcp_src_port***	***tcp_dest_port***	***state***
of431289_NL	192.168.90.x/24	3434	3551	Active
...	...	...	...	...

- The ***id*** (16 bits) field stores the ID used to identify the specific OpenFlow Controller. This field is the primary key of the table.
- The ***ip_address*** (64 bits) field stores the destination IP address used in the secure connection between OpenFlow Controller and the MME.

- The ***tcp_src_port*** (16 bits) field stores the source TCP port used in the secure connection between OpenFlow Controller and the MME.
- The ***tcp_dest_port*** (16 bits) field stores the destination TCP port used in the secure connection between OpenFlow Controller and the MME.
- The ***state*** (4 bits) field stores the current state of life of the OpenFlow Controller. Current valid values for this field are Active and Inactive. Inactive is used when the controller is currently unused due to failure or maintenance.

3.2.4 Message Flow

Having already introduced the signaling messages required to deploy IP address continuity and traffic redirection in the operator's transport network, Figure 3.9 shows the message flows needed to setup a new downlink path towards the UE's current S-/P-GW (target S-/P-GW) in the operator's transport network.
The following procedures refer to the Partial OpenFlow transport network solution presented in Section 3.2.2.

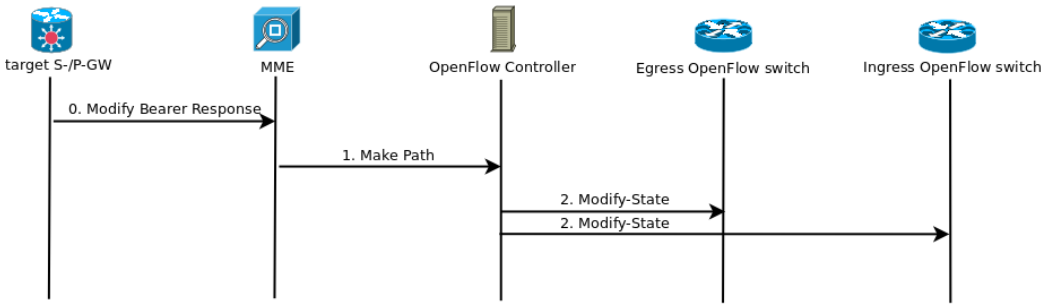


Figure 3.9: Message flow to establish a new downlink path upon reception of a Modify Bearer Response message during handover with S-/P-GW relocation procedure

0. This step is the same as step 5 in Section 2.2.1 and step 17 in Section 2.2.2.

1. The MME creates a *Make Path* message (*flag* field sets to either 01, 10 or 11 value) setting as the *current_address* field the IP address extracted from the Modify Bearer Response message received from the target S-/P-GW. By looking in the UE Context of the UE in object, the MME can retrieve the IP address belonging to flow(s) kept active by the UE after handover. This address is set as the *previous_address* field in the *Make Path* message. If more than one active IP addresses are found in the UE Context, the MME duplicates the *Make Path* message created so far and set the *previous_address* field accordingly. Each message is then encapsulated in a TCP/IP packet using the information retrieved from the *OpenFlow Controller* Info table and then forwarded to the OpenFlow Controller(s).

2. Each OpenFlow Controller, when receiving a message from the MME, looks at the *flag* field to discover whether is a *Make Path* or a *Tear Down Path* message. In this case two sets of *Modify-State* messages needs to be setup.
   The first set is intended to be sent to the Ingress switches and therefore in the flow's action set two operations are specified in the following order: *Set-Field* and *Output*. Both actions are defined as specified before in this Section, depending on which network (full or partial OpenFlow-enabled) and solution have been implemented in the operator's transport network.
   A different *Modify-State* message is created for the Egress switch connected to the target S-/P-GW. Differently from the messages directed to the Ingress switches, the *Output* action precedes the *Set-Field* action in the action set list. The "egress" *Set-Field* action is defined in a way that reverses the modification performed on the data packet's headers by the Ingress switch. All the packets are then forwarded to the entitled entities.

If *Modify-State* messages are received with a substantial advance at the Ingress OpenFlow switches than at the Egress OpenFlow switches, this can cause the unfortunate situation where Egress flow tables have not been updated while IP headers of downlink data packets directed to the moving UE(s) are being already modified at the Ingress OpenFlow switches. These downlink data packets will be delivered to the target S-/P-GW carrying the wrong IP address and they will then be dropped.

To avoid the above, *Modify-State* messages directed to the Ingress OpenFlow switches can be transmitted with a safety delay than the one directed to Egress OpenFlow switches. The value of this safety delay is dependent on the network topology deployment.

Figure 3.10 shows the message flows needed to remove the previously established downlink path.

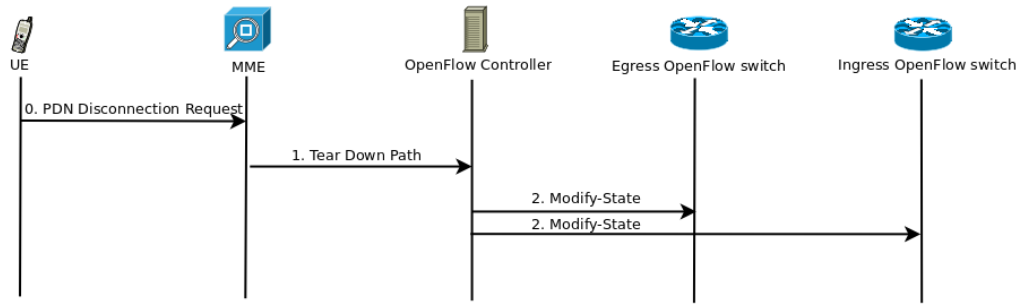


Figure 3.10: Message flow to remove a downlink path upon reception of PDN Disconnection Request from the UE

0. This step is the same as step 1a in Section 2.2.3.

1. The MME creates a *Tear Down Path* message (*flag* field set to 00). *current_address* and *previous_address* messages are extracted from the default bearer indicated by the PDN Disconnection Request (LBI) message received from the UE and from its UE Context. The *Tear Down*

Path message is then encapsulated in a TCP/IP packet using the information retrieved from the *OpenFlow Controller Info* table and then forwarded to the OpenFlow Controller(s).

2. When a message is received from the MME the OpenFlow Controller looks at the *flag* field to discover whether it is a *Make Path* or a *Tear Down Path* message. In this case two sets of *Modify-State* messages will be created and forwarded to the Ingress switches and the Egress switch respectively. The *Modify-State* messages to remove the terminated flow from the Ingress or Egress flow table are defined as specified in [47].

3.2.5 Challenges

OpenFlow is a protocol still under development but it has already attracted enough attention. In this Section different network design and solutions that can shape the operator's transport network of the future have been shown. Choosing which design is more suitable is dependent on operator's will, given the fact that switching to either full or partial OpenFlow-enabled network can be a big economic investment. Differently from what has been mentioned in Section 3.1.5, the position of the controller in the network has a lower impact on a full OpenFlow DMM solution performance, since [47] specifies that controller-to-switch signaling is not to be run through the OpenFlow pipeline and therefore a separate dedicated network can be used for signaling. Still the use of multiple controllers apart from increasing the reliability of the system, it also may have an impact on the time needed to setup the Ingress and Egress switches flow tables. This impact is yet to be researched.

3.3 Preliminary comparison

In this Section the Double NAT based DMM solution and the two OpenFlow based DMM solutions (Full OpenFlow data transport network and Partial OpenFlow data transport network) will be compared. This is a preliminary comparison since the performances of the solutions are not taken yet into consideration. The goal of this comparison is to clarify how performances of solutions developed using different protocols and mechanisms can be compared and to better understand which impacts the deployment of either one DMM solution or the other can have on operators.

The metrics used in the following comparison are specified as following:

- ***Total cost of ownership (TCO):*** defined as the financial estimation of the direct and indirect costs of a technological product (in this case solution or system) over its life cycle [48], the TCO is composed by three main categories of costs: *hardware and software*, *operation expenses* and *long term expenses*.
  For the purpose of this comparison, considered part of the *hardware and software* cost are: network hardware and software, installation and integration of hardware and software, warranties

and licenses, compliance, migration expenses and risks (susceptibility to vulnerabilities, availability of upgrades, patches and future licensing policies, etc.).
For the purpose of this comparison the following element are considered part of *operation expenses*: infrastructure, downtime, outage and failure expenses, security (prevention), backup and recovery process, audit and technology training.
Replacement and future upgrade costs are considered part of the *long term expense*.

- ***Complexity***: it is a common thought that an operator's ability to manage a network decreases as the network become more complex [49]. The complexity is not to be intended only as how complex the network design is but also how network manager activities are affected by an increased complexity (in this case the addition of the DMM solution in the operator's transport network).

- ***Scalability***: is the ability of a network to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth [50].

- ***Potential***: how powerful is the proposed solution in terms of configurability, features provided to the network, flexibility and extensibility.

The following three different DMM solutions have been compared:

1. *Double NAT* based with single NAT Controller.

2. *Full OpenFlow* data transport network solution with single OpenFlow Controller and per-anchor point data forwarding scheme.

3. *Partial OpenFlow* data transport network solution with single OpenFlow Controller.

For every metrics 5 different values can be given: very low, low, medium, high and very high. The results of the comparison are summarized in Table 3.8 and discussed in the following.

Table 3.8: Preliminary comparison

	Double NAT	*Partial OpenFlow*	*Full OpenFlow*
TCO	very low	high	very high
Complexity	low	low	high
Scalability	high	high	low
Potential	low	high	very high

At first glance the Double NAT based DMM solution and the Partial OpenFlow data transport network solution seem to have outperformed the solution implemented in a full OpenFlow-enabled data transport network. Although having better results in three out of four metrics, *Potential* can be considered

a crucial criteria. It is not a secret that OpenFlow is a very powerful tool, and having a full OpenFlow-aware network brings to operators huge advantages. Specific flows can be used to test new protocols, routes, recovery mechanism and so on. But OpenFlow is not only this. Next to the normal switch and router features many other actions (some of them required by standard while other optionals) are available. As explained extensively in Section 3.2, an instruction like the *Set-Field* option can be seen as an enhanced Network Address Translation. In fact this action modifies not only IP and transport protocol headers fields but also Ethernet headers, VLAN headers and MPLS labels, among the others. Furthermore different queue schemes can be setup and packets can be specifically assigned to a queue using the *Set-queue* action. Tags such as VLAN tags can be popped or pushed from packets. IPv4 TTL, IPv6 Hop Limit or MPLS TTL values of a packet can be modified at run-time giving to operators flexibility to cope, for instance, with failures in the network. Last but not the least the OpenFlow protocol is still under development and it had attracted attention also from academia. Greater future potentials are therefore to be considered when evaluating OpenFlow.

Everything has a price. Implementing a full OpenFlow data transport network will require an important increase in the network management *complexity*. Flows require to be setup and controlled per-switch. Actions are assigned on a per-flow basis. And all these operations are handled by the OpenFlow Controller. Fortunately from release 1.3.0 [47] a protocol to support multiple controllers has been designed. Furthermore if the network is composed by *OpenFlow-only* switches, then a separated network is required to transport the signaling from the OpenFlow Controller to the switches since this traffic is not to be run through the OpenFlow pipeline. Since signaling traffic can be transported over the same network as the data traffic, the adoption of a partial OpenFlow data transport network solution has a lower impact on the network complexity. This is possible because OpenFlow switches are used only at the edges of the network to redirect traffic (by implementing *Set-Field* actions to replace packet header fields) while from the ingress to the egress point of the network a normal IP routing network is deployed. Network management complexity is also severely reduced in the partial solution (at the cost of having a lower flexibility and less features in the transport data network) since only Ingress and Egress switches have to be managed by the OpenFlow Controller. If a per-anchor point forwarding scheme is used the amount of flow table updates needed due to UEs moving and anchoring their traffic to a new anchor point is almost halved.

The partial OpenFlow data transport network is very similar to the Double NAT solution and this can be seen on the impact that they both have on the actual network deployment, not only in terms of complexity but also *scalability*. Both solutions (Double NAT and Partial OpenFlow) add entities only at the edges of the network and therefore the scalability of the inner transport network is similar. It is although important to notice that whereas the OpenFlow Controller entity and the protocol used to interact with the OpenFlow switches is already available, a controller and a protocol to correctly setup the NAT routers remotely still need to be developed.

The Full OpenFlow solution, if deployed using *OpenFlow-hybrid* switches, will have a decreased level

of network design complexity since the data transport network can be reused by exploiting the fact that the OpenFlow switches also implement L3 routing-based forwarding. But the scalability of the system will be heavily affected because the addition or removal of an entity in the operator's transport network will require a number of flow updates that can be equal to the number of switches present in the network. Besides this a sudden growth in the traffic can be easily handled in a full OpenFlow network since traffic can be spread fairly in the transport network, giving a higher traffic management capability.

Both *OpenFlow-hybrid* switches and *OpenFlow-full* switches are much more expensive than normal routers with NAT capability. If the entire operator network has to be updated to provide full OpenFlow awareness, the network installation costs can be simply not affordable to operators. The hardware and software slice of the *TCO* are on the other hand low for both Double NAT and Partial OpenFlow solutions but if a big number of Ingress and Egress switches are needed, the hardware, installation and integration costs of the partial OpenFlow solution can become quite high. Furthermore the OpenFlow protocol is still under development and extra cost for future upgrading might need to be taken into consideration.

While OpenFlow might need a longer technological training compared to the old and known NAT, there is not a big difference in the operation expenses of the three solutions but OpenFlow, as a brand new solution, might bring lower long term expenses compared to Double NAT which will just add new functionalities to entities already present in the operator's network.

In conclusion Double NAT and partial OpenFlow solutions provide a very similar DMM level of support with a low impact on the current network deployment. Although integrating the OpenFlow solution in their network will be definitely more expensive for operators, the current extra set of features and extensibility of the OpenFlow protocol can be very appealing. Decision will probably be related to the size of the network, and therefore of investments that operators are willing to make, since whereas Double NAT mainly requires to add functionalities to existing network entities, OpenFlow requires the addition and integration of completely new entities and technology to the operator's network.

Regarding the full OpenFlow data transport network solution, due to the huge upfront investment needed to migrate the current transport network to a fully OpenFlow-enabled network, it can be seen as a more feasible solution for new and small operators that will approach the market in the near future with the introduction of fresh sources of revenue in the form of new telecommunication technologies.

Table 3.9 shows how the three compared DMM solutions fulfill the requirements to be supported by a DMM solution when applied in cloud based LTE systems and previously introduced in Section 2.1.1.

The two OpenFlow solutions do not fulfill the *Co-existence* requirement since they both require the introduction of new entities and technology in the operator's transport network.

Being data forwarding based on L3 routing schemes, both Double NAT and Partial OpenFlow DMM

Table 3.9: Requirements fulfillment
(*Y: Yes, N: No, P: Partial, N.c: Not considered, C: Considered, O.l.: Only local, D: Depend*)

	Double NAT	*Full OpenFlow*	*Partial OpenFlow*
Distributed deployment	Y	Y	Y
Transparency	Y	Y	Y
IPv6 deployment	Y	Y	Y
Co-existence	Y	N	N
Security	N.c.	N.c.	N.c.
Flexible multicast distribution	Y	Y	Y
Dynamicity	N	Y	N
Separating control and data planes	Y	Y	Y
Network-based	Y	Y	Y

solution cannot provide dynamic split of data flows belonging to the same UE along different paths. A combination of *Set-Field* and *Output* actions can be used to provide dynamic per-flow forwarding in the case that the operator's transport network is fully OpenFlow-aware.

Security has not been considered yet for all three solutions.

Given the above, in the following Chapters the performances of these three DMM solutions will be evaluated using a simulation approach. The goal of this evaluation is to demonstrate that each solution can provide seamless DMM when UEs change their mobility anchor point in a cloud based LTE system.

Chapter 4

Simulation Experiments Approach

In this Chapter, the choice of NS3 LENA simulation environment is motivated, and then the implementation of the solution is discussed. The implementation of the models is based on the requirements, specifications and designs in Chapters 2 and 3.
The implementation source codes for all the modules described in this Section can be found in [51], and the guideline to use the modules is included in the Appendix.

4.1 Simulation environment and assumptions

NS3 LENA has been chosen as the simulation tool to implement and evaluate the DMM solutions introduced in Chapter 3. NS3 [52] is an open-source discrete-event network simulator, targeted primarily for research and educational use. NS3 is gaining more and more population compared to its long established predecessor NS2. Differently from NS2, network simulation in NS3 can be implemented in pure C++, while the building environment is managed using Python.

Both IP and non-IP based networks are supported within the NS3 core. However, the large majority of its users focus on wireless/IP simulations which involve models for WiFi, WiMAX, or LTE layers 1 and 2 and a variety of static or dynamic routing protocols such as OLSR and AODV for IP-based applications.

Being an open source driven tool most of the models and features of NS3, although very powerful, are not fully complete and working. Therefore users are requested to adapt their needs to the actually implemented features, or to extend NS3 on their own.

LENA [53] is an LTE-focused branch of NS3 developed at CTTC (Centre Tecnològic Telecomunicacions Catalunya). This experimental branch is based on the developing branch on NS3, but uses a completely rewritten and enhanced model for LTE from the original one of NS3. Periodically the major release version of LENA is merged with the developing branch of NS3.

4.1.1 EPC model in LENA

An overview of the LTE-EPC simulation model is depicted in Figure 4.1. The overall architecture of the LENA simulation model is comprised of two main components:

- LTE Model. This model includes the LTE Radio Protocol stack (RRC, PDCP, RLC, MAC, PHY). These entities reside entirely within the UE and the eNodeB nodes.
- EPC Model. This models includes core network interfaces, protocols and entities. These entities and protocols reside within the S-GW, P-GW and MME nodes, and partially within the eNodeB nodes.

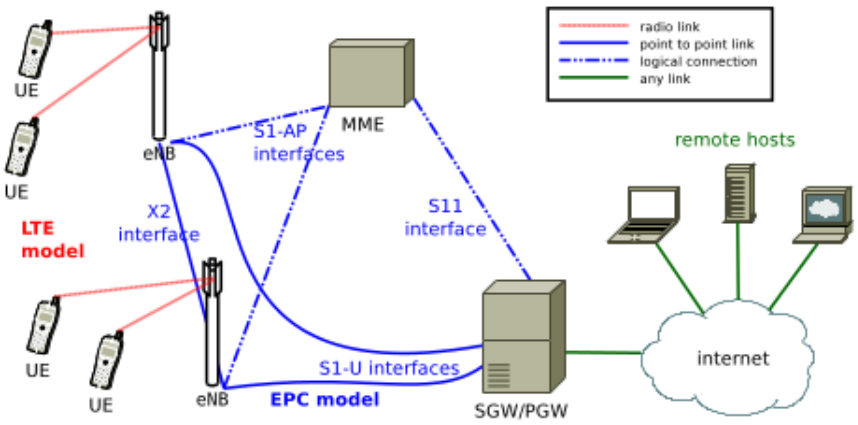


Figure 4.1: Overview of NS3 LENA's LTE-EPC simulation model, copied from [6]

Being this research focused on the EPC and the transport network above it, only the EPC model will be extensively discussed in the following. Furthermore the architecture of both control plane and data plane implementation will be useful to better understand the enhancements that had to be applied to LENA.

The main objective of the EPC model is to provide means for the simulation of end-to-end IP connectivity over the LTE model. To this aim, it supports for the interconnection of multiple UEs to the Internet, via a radio access network of multiple eNodeBs connected to a single S-GW/P-GW node, as shown in Figure 4.1.
S-GW and P-GW functionality are contained in a single S-GW/P-GW node, which removes the need for the S5 or S8 interfaces specified by 3GPP. On the other hand, for both the S1-U protocol stack and the LTE radio protocol stack all the protocol layers specified by 3GPP are present although some of them only partially implemented.

The EPC model has several design criteria [6]:

- the only PDN supported type is IPv4;
- S-GW and P-GW functionalities are encapsulated within a single node, referred as S-GW/P-GW node;
- inter-cell mobility is not implemented, hence just a single S-GW/P-GW node is defined;
- any standard NS3 application working over TCP or UDP must work with EPC, so to be able to use EPC to simulate end-to-end performance of realistic applications;
- it is possible to define more than just one eNodeB, every one of which with its own backhaul connection, with different capabilities; hence data plane protocols between eNodeBs and S-GW/P-GW had to be modeled very accurately;
- it is possible for a single UE to use different applications with different QoS requirements, so multiple EPS bearer should be supported (and this includes the necessary TCP/UDP over IP classification made on the UE for uplink traffic and on eNodeB for downlink traffic);
- accurate EPC data plane modeling is the main goal, while EPC control plane was to be developed in a simplified way;
- main objective for EPC simulations is the management of active users in ECM connected mode, so all the functionalities that are relevant only for ECM idle mode (i.e. tracking area update and paging) are not modeled at all;
- the model should allow the possibility to perform an X2-based handover between two eNodeBs.

From these design criteria it appears clear that modifications are required to the current NS3 LENA EPC model in order to support multiple mobility anchor points for UEs' traffic (S-GW/P-GW) within the same simulation. These modification will be discussed in Section 4.2.1.1.

Control plane

The EPC control plane interfaces that are modeled explicitly are the S1-AP, the X2-AP and the S11 interfaces. The S1-AP and the S11 interfaces are modeled in a simplified fashion, by using just one pair of interface classes to model the interaction between entities that reside on different nodes (the eNodeB and the MME for the S1-AP interface, and the MME and the S-GW for the S11 interface). In practice, this means that the primitives of these interfaces are mapped to a direct function call between the two objects. On the other hand, the X2-AP interface is being modeled using protocol data units sent over an X2 link (modeled as a point-to-point link); for this reason, the X2-AP interface model is more realistic.

Data plane

The S1-U interface is modeled in a realistic way by encapsulating data packets over GTP/UDP/IP, as done in real LTE systems. The corresponding protocol stack has been shown in Figure 4.2. As shown in the figure, there are two different layers of IP networking. The first one is the end-to-end layer, which provides end-to-end connectivity to the users; this layers involves the UEs, the P-GW and the remote host (including eventual Internet routers and hosts in between), but does not involve the eNodeB. By default, UEs are assigned a public IPv4 address in the same subnet network as the P-GW address. The P-GW address is used by all UEs as the gateway to reach the Internet.

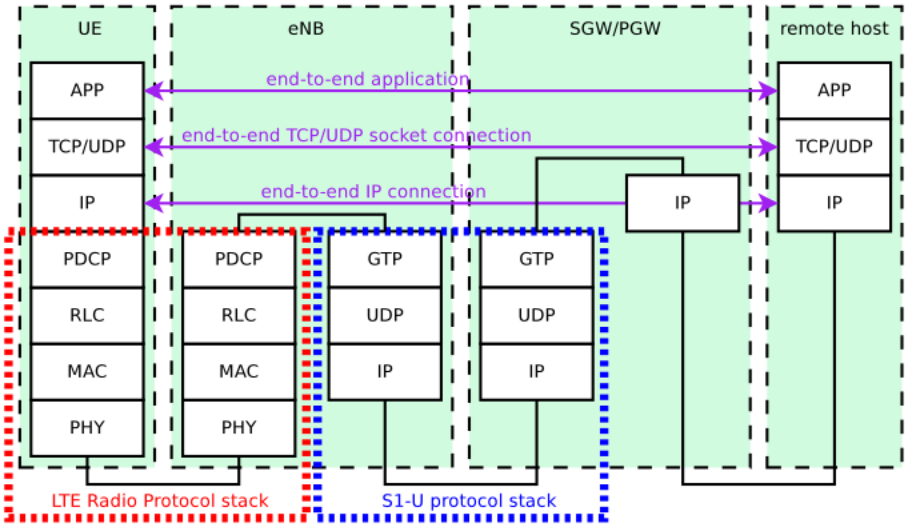


Figure 4.2: NS3 LENA's LTE-EPC data plane protocol stack, copied from [6]

The second layer of IP networking is the EPC local area network. This involves all eNodeB nodes and the S-GW/P-GW node. This network is implemented as a set of point-to-point links which connect each eNodeB with the S-GW/P-GW node; thus, the S-GW/P-GW has a set of point-to-point devices, each providing connectivity to a different eNodeB.

The readers are advised to refer to the Design Documentation of LENA [6] for more details about the design criteria and implementation of the LTE and EPC models in LENA.

4.1.2 NAT model in LENA

To implement Network Address Translation an external model [54] needs to be added to the current NS3 stable release. The design of NAT for NS3 is basically divided into two main categories:

- Static NAT which allows IP flows to be established in either direction. It is designed to perform host to host NAT and it also has a variant to specify the NAT for specific protocol and port.

- Dynamic NAT which allows IP flows to be established only in one direction, from private address realm to public address realm. Often, multiple hosts may be multiplexed onto a single public IP address via port translation. The NAT state is dynamic and times out after a period of inactivity.

Using an `Ipv4NatHelper` entity, `Ipv4Nat` capabilities can be added to an NS3 `Node` entity. From that moment the `Node` will perform NAT table lookup on each incoming packet. Both pre-routing NAT or post-routing NAT features have been implemented. `StaticNatRule` and `DynamicNatRule` can be added and withdrawn from the NAT table of a NAT-able node using a simple function call.
No additional delay is added to simulate the NAT table lookup process.

Currently this model works only with IPv4 addresses. The source code of this model can be both found at [55].

4.1.3 OpenFlow model in LENA

A module implementing OpenFlow switches is already present in the latest NS3 stable releases. The model relies on building an external OpenFlow switch library (OFSID), and then using some wrappers to call out to the library.

The OpenFlow module presents a `OpenFlowSwitchNetDevice` and a `OpenFlowSwitchHelper` for installing it on nodes. The NS3 OpenFlow switch device models an OpenFlow-enabled switch. It is designed to express basic use of the OpenFlow protocol, with the maintaining of a virtual Flow Table and TCAM to provide OpenFlow-like results.
An additional delay of 30 nanoseconds (it can be varied setting `m_lookupDelay`, an attribute of the `OpenFlowSwitchNetDevice` module) is added as an overhead to perform a lookup in the OpenFlow table (30 ns correspond to the delay of a standard TCAM on an FPGA).

The functionality comes down to the Controllers, which send messages to the switch that configure its flows, producing different effects. Controllers can be added by the user, under the ofi namespace extending ofi::Controller.

The description of the NS3 OpenFlow module implementation and the instruction on how to integrate the OFSID library can be found at [56] while the used source code is available at [57].

4.1.4 Assumptions

The following assumptions have been made in order to implement a cloud based LTE system using NS3 LENA:

- The MME, source and target eNodeBs and source and target S-/P-GWs belong to the same EPS system.
- Each EPC entity (MME, source S-/P-GW and target S-/P-GW) is assumed to be virtualized and running inside a micro or macro data center.

- Source and target S-/P-GW virtualized entities are running into data centers placed in difference locations.
- Each E-UTRAN entity (source eNodeB and target eNodeB) is assumed to be virtualized and running inside a micro data center placed closer to the location of the physical serving base station.
- Source and target eNodeB virtualized entities are running into micro data centers placed in difference locations.
- All the data centers where the virtualized EPS entity are running are connected to the same transport network. Furthermore no additional delays have been implemented to emulate the local delivery of packets within a data center and the processing time required by the virtualized applications to fulfill their tasks.

4.2 Simulation topology and parameters

This Section describes the simulation topology and the modifications required to support it in NS3 LENA, explains the simulation parameters used for the simulation experiments and finally shows the metrics that will be used to evaluate the performance of the solutions.

4.2.1 Simulation topology

The goal of the experiments is to evaluate the seamlessness of the DMM solutions described in Chapter 3 when UEs are changing their mobility anchor point which in cloud based LTE networks is identified with the S-/P-GW entity (an entity embodying functionalities of both S-GW and P-GW). A DMM solution in a LTE system is considered to be seamless if, when UEs are handed over to a target eNodeB served by a different S-/P-GW, no extra delay nor packet loss is introduced in the traffic flows kept active upon movement.

Figure 4.3 shows the logical topology used in the experiments.

UEs are handed over from source eNodeB to target eNodeB being served by two different S-/P-GW namely source S-/P-GW and target S-/P-GW. IP flows are initiated by UEs when attached to the source eNodeB using an IP address allocated from the source S-/P-GW address pool and topologically anchored to the source S-/P-GW. Due to the fact that the X2-handover procedure has been already implemented, although still under development, in NS3 LENA, the procedure described in Section 2.2.1 is used to deploy IP address continuity in the EPS. Thus when UEs attach to the target eNodeB, the previously initiated IP flows are kept active. If present the X2 tunnel deployed between source and target eNodeB is used to forward packets received by the source eNodeB during the handover execution. Being the IP addresses used by the active flows topologically unrelated with the position

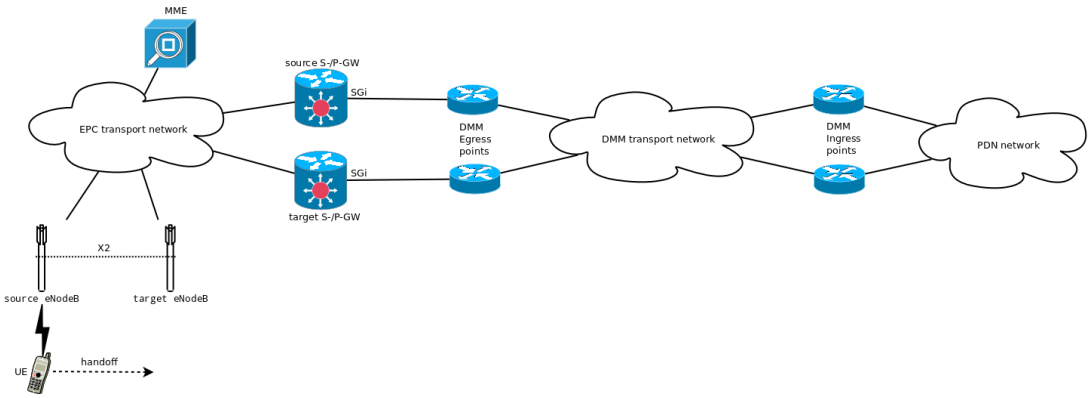


Figure 4.3: Logical simulation topology

of the target S-/P-GW, the solutions proposed in Chapter 3 are used to re-direct the traffic towards the currently used S-/P-GW. Traffic redirection happens therefore in the operator's transport network above the EPS, while the usual bearer encapsulation is used within the EPS.

In a cloud based LTE system EPC and (optionally) E-UTRAN components are running on virtual machines located inside macro or micro data centers. These data centers are spread within the tracking area covered by the LTE network which they belong to. Virtualized E-UTRAN components such as eNodeBs and HeNBs are located as close as possible to their peer physical entity, while the location of EPC components such as S-/P-GW and MME can vary depending on the dimension of the LTE network, dimension of the operator's backbone network, location of the data centers, efficiency reasons and so forth.

All these data centers are connected with each other via what has been referred to as the operator's transport network. The term might not be fully correct if one thinks, for instance, at the case when the cloud infrastructure is provided to LTE operators by an external provider. In the following the data center traffic and EPS traffic are assumed to be forwarded via the same network.

The same operator's IP transport network topology is used in all the experiments. This topology is a small part of the real network topology of one of the biggest European ISP: Ebone. The topology has been implemented in LENA using a map provided by the Rocketfuel project [58]. ISP topology are inferred by the Rocketfuel engine using a traceroute technique[59].

A part of the Ebone topology covering The Netherlands, north-east Belgium and north-west Germany has been extracted and it is depicted in Figure 4.4.

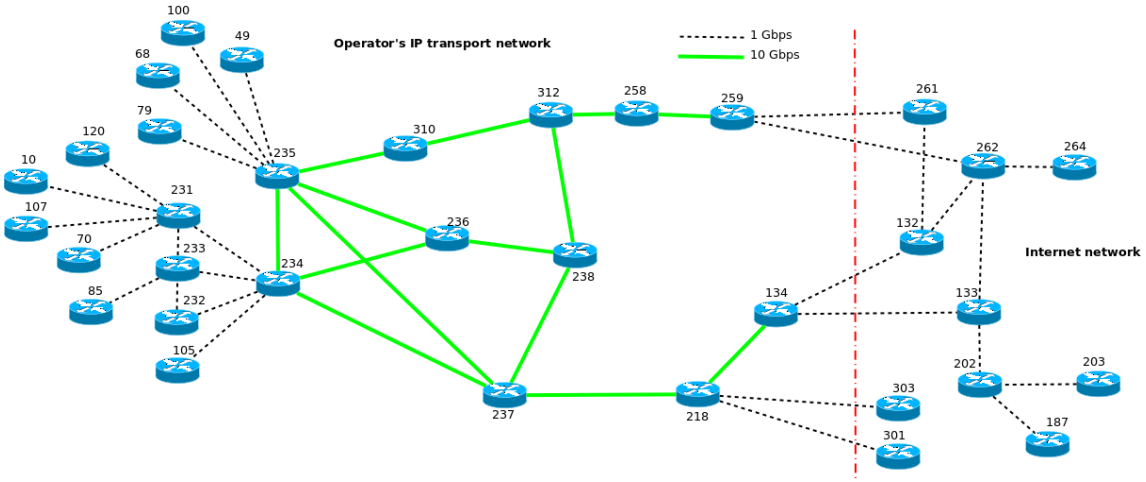


Figure 4.4: Simulation IP transport network topology

The backbone part of the network is depicted with solid lines and it uses 10-gigabit Ethernet links while the rest of the operator's network and the Internet network use gigabit Ethernet links (dotted lines).

Node 134, 259 and 218 are the operator's point of presence to the Internet.

Due to the fact that, in the simulation experiments, the impact of having a higher distance (hops count) between the Ingress and Egress points of the DMM transport network will be evaluated, eight additional routers (number 35, 36, 37, 38, 39, 40, 41 and 42) have been added to the topology between two of the operator's Internet PoPs and the backbone network. The modified simulation topology is shown in Figure 4.5.

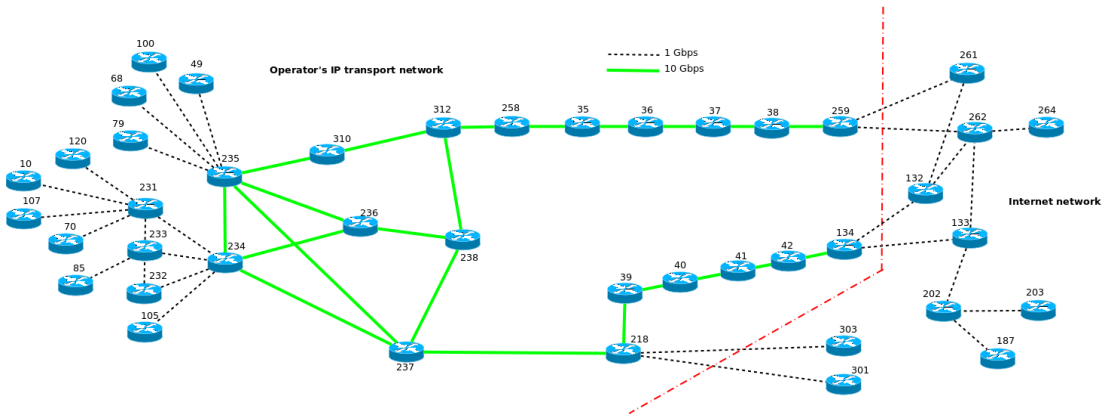


Figure 4.5: Modified simulation IP transport network topology

Independently on which DMM solution is deployed, during the simulation experiments the best positioning for the DMM Ingress and Egress routers will be evaluated. For this reason three different topologies have been defined and are shown in Figure 4.6, Figure 4.7 and Figure 4.8 respectively.

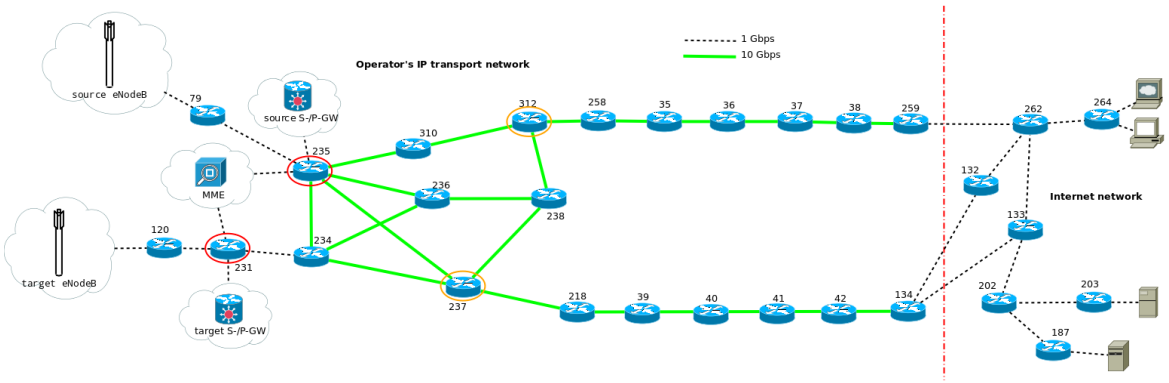


Figure 4.6: First simulation network topology with virtualized LTE-EPC entities: DMM Ingress and Egress points close to the EPC part of the network

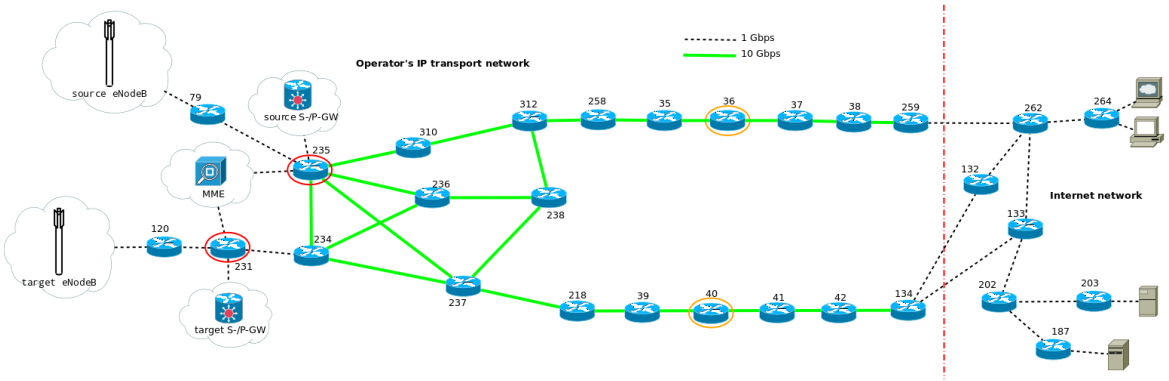


Figure 4.7: Second simulation network topology with virtualized LTE-EPC entities: DMM Ingress placed deeper in the core network

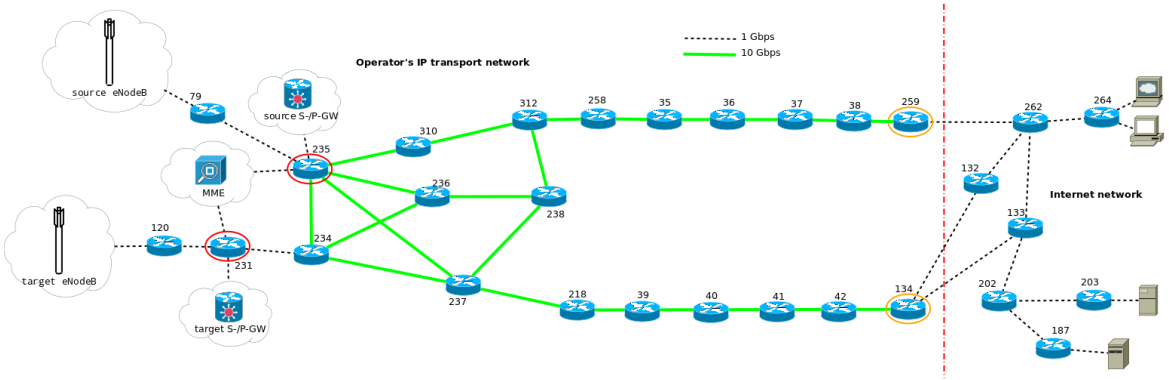


Figure 4.8: Third simulation network topology with virtualized LTE-EPC entities: DMM Ingress and Egress on opposite edges of the operator's network

The virtualized EPC and E-UTRAN entities are shown in all figures together with their point of attachment to the operator's IP transport network. Both source and target S-/P-GW virtualized entities

are placed with a one hop distance to the data centers where source and target eNodeB virtualized entities are running. The virtual MME entity is running in a data center placed at the same distance to all other four EPC and E-UTRAN entities.
Circled in red are the two routers which will implement Egress DMM functionalities while routers implementing Ingress functionalities have been circled in gold. In all three topologies the Egress routers are the same, being the next hop routers from the data centers where source and target S-/P-GW entities are running.
The position of the routers implementing Ingress DMM functionalities determines the difference between the three topologies. In the first topology (Figure 4.6), Ingress DMM routers have been placed closer to the part of the network where the data centers running the virtualized EPC and E-UTRAN entities are located. Therefore the hop count between Ingress DMM routers and Egress DMM routers is on average close to 1.
In the second topology (Figure 4.7) the Ingress DMM routers have been located halfway between the EPC and the Internet network. The average hop count between Ingress DMM routers and Egress DMM routers has increased to 4.
In the third and last topology (Figure 4.8) the Ingress DMM routers have been located as close as possible to the Internet network, on two of the three operator's Internet PoPs. The average hop count between Ingress DMM routers and Egress DMM routers has increased to 7.

Remote hosts which have active connection with the UEs attached to both source and target eNodeB are in all topologies accessing the Internet network via nodes 264, 203 and 187.

Whereas the defined positions of Ingress and Egress DMM points can be considered trivial, the same cannot be said for the position of the DMM Ingress and Egress control function in the network. As explained in details in Section 3.1 and 3.2, the DMM Controller (either NAT Controller or OpenFlow Controller) receives UE mobility context information from the MME. These information are needed to correctly setup the DMM Ingress and Egress function in order to implement the correct traffic redirection towards the UE's currently used S-/P-GW. Due to the limited size of the simulation network only one controller has been implemented. When required the controller is attached to the core network using a 1 Gbps link. Three different positions have been defined for the DMM Controller and are show in Figure 4.9.

- position #1 (*MME co-located*): the DMM Controller is co-located with the EPC MME entity making the signaling between the two entities an intra-data center communication. The fact that signaling from the MME is delivered locally to the DMM Controller, can decrease the latency in setting up DMM traffic redirection. This position can be also referred to as *MME co-located*.
- position #2 (*middle of CN*): the DMM Ingress and Egress control function is placed closer to the Egress part of the DMM transport network, precisely at an average hop distance of 1.5 from

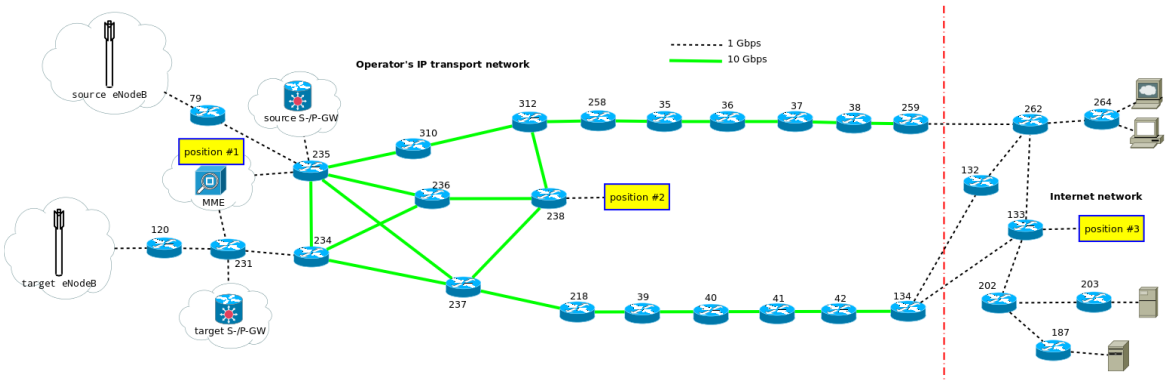


Figure 4.9: DMM Ingress and Egress control function positioning

both Egress points. The data center where the virtualized MME entity is running is located two hops further. This position can be also referred to as *middle of CN*.

- position #3 (*outside CN*): the DMM Controller is placed further away from the access network, closer to the operator's Internet PoPs. Precisely the DMM Controller is located in a position external to the operator's IP transport network. This can happen for instance if the DMM functionalities are provided to the operator by an external provider or if the simulation topology is only a slice of the entire operator's network. The distance of the DMM Controller from the MME and to the DMM Egress functions reaches the 9 hops with a mixed link speed (1 Gbps or 10 Gbps). This position can be also referred to as *outside CN*.

4.2.1.1 Modifications required to support simulation topology in LENA

In order to support the presented simulation topologies several modifications were required to NS3 LENA. Being the researchers that implement LENA mainly focused on the radio access part of the LTE protocol, only one S-GW/P-GW entity can be instantiated in the current implementation. Furthermore X2-based handover has been lately added to LENA stable versions but, for clear reasons, S-GW and MME relocations are not supported. Lastly only CSMA transmission technology is present as an alternative to Point-to-Point connections.

The modifications applied to overcome the briefly introduced limitations will be discussed in the remainder of this subsection. The implementation source codes for all the modules described in this subsection can be found in [51], and the guideline to use the modules is included in the Appendix.

Support to multiple S-GW/P-GWs

In order to support the instantiation of multiple S-GW/P-GW entities, the `EpcHelper` module has been enhanced. S-GW/P-GW entities are independent from each other meaning that one eNodeB can be served by one and only one S-GW/P-GW. In other words multiple standalone EPS subsystems

have been created. Since S11-based signaling with the MME has been implemented only in a virtual fashion, the MME entity can be shared between multiple S-GW/P-GWs.

Being IP address continuity needed in the EPS to support DMM, each S-GW/P-GW entity gets assigned a pool of IP addresses in the same subnet as their IP address advertised outside the EPS system (i.e. the IP address assigned to the SGi interface). Furthermore, in NS3 LENA a virtual network device, namely `TunDevice`, is used on the SGi interface (together with an actual `NetDevice`) to filter downlink packets belonging to UEs which traffic is anchored at the S-GW/P-GW. The `TunDevice` has assigned an IP address from the S-GW/P-GW's pool and it will push up the stack only packets which IP destination address belong to the same subnet. In order to accept IP addresses topologically anchored to a different S-GW/P-GW, multiple `TunDevice` are allocated within each S-GW/P-GW entity. The enhanced source code can be found in [60].

Support to X2-based handover procedure with S-GW/P-GW relocation

Having introduced multiple S-GW/P-GW entities in the simulation topology, modifications are required in order to support the relocation of the UE's mobility anchor point upon handover. Following the modifications described above, handover's source and target eNodeB will be part of two separated EPS subsystems. Despite that, an X2 connection can be established between the two eNodeBs, being both cells in principle just NS3 `Node` within the same network. For this reason the same X2-based handover procedures as the one currently deployed in NS3 LENA can be used to perform radio handover between two cells, although they are served by two different S-GW/P-GW entities. For this reason upon handover UEs can keep their previous initiated IP flows active using the original flows' IP address(es).

S1-U tunnels need to be established between target S-GW/P-GW and target eNodeB to forward both uplink and downlink UEs' traffic. For doing so a `ModifyBearer` function has been implemented within the `EpcHelper` module. This function will be called as soon as the radio handover has been completed. This function emulate what done between steps 2 and 5 of the procedure described in Section 2.2.1 by establishing an EPS bearer in the the target S-GW/P-GW's `EpcSgwPgwApplication` and allocating a new IP address for the moving UE. This IP address has a double functionality: it can be used by the moving UE to establish new bearer within the new EPS subsystem and it is used to setup DMM traffic redirection within the operator's transport network.

Since no S-GW/P-GW relocation mechanisms was present, the deletion of EPS bearers from the source S-GW/P-GW has not been implemented in current NS3 LENA release. This has been exploited as an advantage to implement realistic X2 downlink data forwarding during *handover completion* phase. In fact, although a new EPS bearer has been established in the target S-GW/P-GW, downlink traffic might still be received at the source S-GW/P-GW in the meantime that DMM traffic redirection has not been setup yet. Being the moving UE's session still active in the source S-GW/P-GW, downlink traffic will be forwarded to the source eNodeB. Surprisingly X2 bearers are also kept

infinitely active between source and target eNodeB since a finite state system is used to prevent the misusage of the X2 path. In order to emulate the pushing of downlink data via the X2 path until the expiration of the Delete Session Timer (see Section 2.2.1 for more details), the prevention system used by the developers of LENA had to be bypassed. This is done in the `LteEnbRrc` module by scheduling every 1 ms a function which checks if new packets have been received and buffered in the application layer (`EpcEnbApplication`) of the source eNodeB, and if this is the case, buffered packets will be forwarded via the X2 path to the target eNodeB which will then deliver them to the moving UE. The function is schedule for the first time in the source eNodeB's `LteEnbRrc` module when a Handover Request Ack message is received from the target eNodeB (*RecvHandoverRequestAck* function) and it is rescheduled until the expiration of the Delete Session Timer.
The buffer used in the eNodeB `EpcEnbApplication` is an enhancement to the current LENA's X2-based handover procedure. At the moment in LENA, uplink and downlink traffic are forwarded via the X2 path only if received after the Handover Request Ack message from the target eNobeB. Any packets received at the source eNodeB before the reception of the Handover Request Ack from the target eNodeB and not yet delivered to the UE, or to the S-GW-P-GW, are dropped. To avoid this unpleasant situation all downlink packets received via the S1-U tunnel are buffered at the cell application layer (`EpcEnbApplication`).
No mechanisms has been yet implemented to prevent uplink data to be dropped.

The enhanced source code can be found in [60].

Support for DMM signaling and operations

The signaling procedures described in Section 3.1.4 for Double NAT-based and Section 3.2.4 for OpenFlow-based DMM solutions have been implemented in LENA.
At the moment, signaling traffic is based on packets which carry only IP address information (*previous_address/identifier_address* and *current_address/locator_address*) and are then padded to reach the same size of the messages described in Sections 3.1.2, 3.1.3 and 3.2.3.
`UdpSockets` are setup between the MME node and DMM Controller node, DMM Controller node and Egress routers and DMM Controller node and Ingress routers. Signaling traffic is forwarded through these sockets and upon reception of a new control plane message, a callback is made to a specific function used to process the incoming packet.

When the `ModifyBearer` function has been executed and the new IP address allocated by the target S-GW/P-GW, the MME generate a *Make Path* packet (the *previous_address/identifier_address* is already known to the MME entity) and forward it to the DMM Controller.

In case of Double NAT, the DMM Controller, upon reception of the *Make Path* packet from the MME, generates new *Rule Update* packets and then forwards them to the Ingress and Egress *Node*s. The IP addresses carried by the received *Rule Update* message are used by the Ingress and Egress NAT routers to setup a `StaticNatRule` and add it to their NAT tables.

In the case of OpenFlow, the implementation is different for the Partial OpenFlow network case and the Full OpenFlow network case. In the former, the DMM Controller upon reception of the *Make Path* packet from the MME, creates a *SetField* and *Output* actions and store them for future use. In the case of Full OpenFlow network, only an *Output* action is created. The *SetField* action is created using the IP addresses carried by the *Make Path* message while for the *Output* action it can be assumed that the OpenFlow Controller has a full knowledge of the DMM transport network layout and it is therefore able to map each OpenFlow switch with the correct downlink output port. In both cases the DMM Controller then creates two *ModifyState* packets using the IP addresses carried by the received *Rule Update* message and forwarded them to the Ingress and Egress `Nodes`.
Ingress and Egress OpenFlow switches (both Full and Partial) upon reception of a padded *ModifyState* message extract the *previous_address/identifier_address* and insert it into a buffer situated in the `Ipv4` module. This buffer is used to perform packet filtering at the OpenFlow node. Upon reception of an incoming packet OpenFlow switches will peek the IP header and search for the destination IP address in the buffer. If present, the incoming packet will be run through the OpenFlow table(s). If no actions are present in the flow table(s) yet, the OpenFlow switch will contact the OpenFlow Controller which will then provide them with the previously stored *SetField* and/or *Output* actions. Since the provisioning of the aforementioned actions is done via function call, the flow of messages described in Section 3.2.4 is indeed respected.
If the incoming packet's destination IP address is not present in the buffer, the packet will be transmitted using standard layer-3 routing in case of Partial OpenFlow while in Full OpenFlow network the packet is forwarded out of a predefined standard downlink output port.

The modifications described above have been performed on the source codes of the `Ipv4Nat`, `OpenFlowInterface` and `OpenFlowSwitchNetDevice` modules which be found at [55] and [57].

Support for gigabit Ethernet and 10-gigabit Ethernet

Currently only Point-to-Point links and CSMA bus-style connections are available in NS3 LENA. Being Ethernet also based on the CSMA technology, the `Csma` model has been modified to simulate gigabit and 10-gigabit Ethernet links. Precisely a gigabit Ethernet link can be seen as a pair of parallel CSMA links established between two nodes. Each node acts as the source for one link and the recipient for the other. Each CSMA link has therefore only one source which will then always see the medium as free and transmit as soon as a packet is ready. The `CsmaChannel` module has been modified to implement the above: the channel always remains in the IDLE state in order to avoid the execution of the backoff mechanism while, to deploy a full-duplex medium, current transmitted packets are stored into a vector mapped with the `NodeId` of the transmitter. In this way packets are always delivered to the correct end of the medium even if multiple transmissions have been initiated in parallel.

The enhanced source code can be found in [61].

4.2.2 Traffic generators

In this subsection the applications used to generate traffic in both LTE and IP transport network are introduced.

4.2.2.1 LTE traffic

The traffic mix specified in [62] and [63] and shown in Table 4.1 is used to model the entire LTE traffic (active and background).

Table 4.1: LTE traffic models mix

Application	**Traffic category**	**Percentage of users**
VoIP	Real-time	30%
FTP	Best effort	10%
Web browsing / HTTP	Interactive	20%
Video streaming	Streaming	20%
Gaming	Interactive real-time	20%

VoIP has been selected as the type of traffic used by moving UE which will trigger DMM traffic redirection. The reason for this is given by the fact that a change in the LTE mobility anchor point, although distributed, will mainly be caused by a movement in the order of kilometers (i.e. auto, train or other means of transport). Since vehicles speeds have been used to determine the residence time of a UE in a tracking area, VoIP traffic is more likely to be used by a person when driving or riding along on a car.

The design and implementation in NS3 LENA of the above traffic models mix has been performed by two fellow students. More details on the used design approaches and implemented applications can be found in Section 5.4 and Section 5.5 of their M.Sc. Thesis reports [64, 65].

Being VoIP commonly a two party communication, a server (to receive) and a client (to transmit) applications are installed on both UE and the corresponding remote host. Trace files are generating by both client and server applications on the transmission and reception of packets.

The source code used to implement the above can be found in [66].

4.2.2.2 Operator's IP transport background traffic

Background traffic has been implemented in the operator's IP transport network using an application for generating realistic Internet traffic in NS3 [67]. This tool uses a Poisson Pareto Burst Process (hence the name PPBP-application) model to generate accurate network traffic that matches statistical properties of real-life IP network.

In the PPBP model, bursts arrive according to a Poisson process with rate λ_p, and their length follows a Pareto distribution characterized by Hurst parameter H, typically between 0.5 and 0.9, and a mean T_{on}.
Since each burst gives birth to a flow with a constant bit- rate r, the overall rate of the PPBP, λ, is computed using the following formula:

$$\lambda = T_{on} \times \lambda_p \times r$$

Therefore to generate traffic with rate λ, the bursts arrival rate can be computed as follow:

$$\lambda_p = \frac{\lambda}{T_{on} \times r}$$

At last the average number of active bursts is given by:

$$E[n] = T_{on} \times \lambda_p$$

The source code used to implement the above can be found in the [66].

4.2.3 Handover simulation

UEs are handed over from a source eNodeB to a target eNodeB when the MME is unchanged and the MME decides that the S-/P-GW is to be relocated. IP flows initiated by UEs when attached to the source eNodeB will be kept active upon handover demanding IP address continuity. The procedure introduced in Section 2.2.1 is used for this purpose.

UEs attached to the cell can be divided into two groups: static and moving. Static UEs are meant to remain attached to the same cell for the entire simulation time. They are uniformly distributed in a disc of radius r around the cell where they are attached to. Their task is merely to generate the cell background traffic by sending dummy uplink packets which will be dropped immediately by the serving eNodeB. Their traffic is generated following the traffic mix models introduced in Section 4.2.2.1.
Moving UEs are placed in positions at the same distance from both source and target eNodeB. The adjective "moving" is actually incorrect since no movement is required to be performed by UEs in order to trigger the handover. This is due to the fact that in NS3 LENA the times when handovers occur are per-scheduled by the user before that the simulation is started. A distribution of time is therefore used to schedule the handovers' triggers.
Although the mobility anchor points are placed closer to the E-UTRAN as required by the DMM feature, the frequency with which the UE's S-/P-GW is relocated is not equal to its cell residence time. Z. Haas et al. [68] has estimated that the sojourn time of a GSM mobile station within a Location Area (LA) is exponentially distributed with mean equal to

$$t_{sojourn} = \frac{9R}{(3+2\sqrt{3})V}$$

where R is the "radius" of the LA and V is the average mobile velocity.

In the LTE standard the term Location Area has been replaced by the term Tracking Area (TA) and in the experiments presented in this report, each S-/P-GW entities has been assumed to serve a single Tracking Area with a radius of 50 km. Multiple cells are presented in a tracking area but for the purpose of the experiments the source eNodeB and the target eNodeB cells are assumed to be neighboring cells belonging to different TAs. For this reason when UEs are handed over to the target eNodeB, their TA changes and so does the serving S-/P-GW.

Since moving from one TA to another is most likely to be a movement which requires a transport vehicle (e.g. driving on a highway), the average velocities (V) of moving UEs are estimated using the free speed distributions model introduced in [69]. The data to built this model have been collected in different day-time periods on the A9 dutch motorway.
A normal distribution is built using mean and standard deviation extracted from Table 3 of [69]. These values refer to the speed of any vehicle (cars and trucks) riding on the highway between 11 AM and 3 PM and they have been estimated using a modified Kaplan-Meier approach [70]. The values are reported in Table 4.2.

Table 4.2: Mean and standard deviation of normal distributed UEs moving velocities

mean	standard deviation
32.1 (m/s)	4.33

The defined TA's radius and distributed velocity are then used to calculate the TA residence time of each UE. Due to the fact that the resulting times are in the order of hours, the 99.999% of the time is assumed to be passed when the simulation experiments are started (source code available at [71]).

Two handover scenarios are deployed in the simulation experiments. The difference between the two scenarios will be the usage or not of the X2 path.

If present the X2 path is assumed to be established between the two eNodeBs when handovers are triggered and it is used to forward both control and (if present) data plane messages.
X2 data forwarding can occur in both *handover execution* (in both uplink and downlink direction) and *handover completion* (only downlink) phases. When in the latter the source eNodeB checks every 1 millisecond if any packets have been added to the buffer kept for the moving UE. If present, this packet(s) will be forwarded to the target eNodeB via the X2 path. Since the required S11 signaling used to delete the UE's sessions (EPS bearers) at the source S-/P-GW has not been implemented yet in NS3 LENA, a timer is used to avoid that the X2 path remains open forever. This timer can be setup manually by the user and it reflects the functionality of the timer used in the 3GPP's standard by the

MME to release the resources in the source S-/P-GW (see step 4 of the procedure described in Section 2.2.1).

In order to evaluate the impact of not having the X2 data path between source and target eNodeB, this scenario will also be simulated. Signaling traffic will still be sent through the X2 path because no alternatives are present in current NS3 LENA version.

The X2-based handover source code can be found in [60].

4.2.4 Simulation parameters

This subsection presents all the parameters used to conduct the simulation experiments.

4.2.4.1 LTE network

The configuration of the simulation parameters for the LTE network are summarized in Table 4.3. All the parameters are typical for the LTE Release 8 [19] which is implemented in the NS3 LENA simulation environment.

Table 4.3: LTE network simulation parameters

Parameters	**Values**
Uplink bandwidth	5 MHz (25 Resource Blocks)
Downlink bandwidth	5 MHz (25 Resource Blocks)
Source eNodeB Uplink EARFCN	21100 band7 (2535 MHz)
Source eNodeB Downlink EARFCN	3100 band7 (2655 MHz)
Target eNodeB Uplink EARFCN	21150 band7 (2540 MHz)
Target eNodeB Downlink EARFCN	3150 band7 (2650 MHz)
eNodeB MAC scheduler	Proportional Fair (PF)
CQI generation period	10 ms
Transmission mode	MIMO 2x2
UE transmission power	26 dBm
UE noise figure	5 dB
eNodeB transmission power	49 dBm
eNodeB noise figure	5 dB
Cell radius	~3000 m
Cell distance	4000 m

The used channel bandwidth is 5 MHz for both uplink and downlink traffic. This value has been chosen because it corresponds to one of the most common bandwidths implemented currently by operators.

Since source and target eNodeB are neighboring cell two different carrier frequencies belonging to the same band (band7) need to be used to avoid inter-cell interferences in both uplink and downlink

direction. The carrier frequencies are numerically represented by the EARFCN number as per table 5.7.3-1 of [72].

MIMO 2x2 is also applied in the network since it is the advanced technology designed to upgrade the data rate as well as the quality of LTE traffic.

The cell radius of 2 km is chosen since the UEs are moving within a sub-urban area.

Background traffic 80% has been chosen as the level of cell utilization in downlink for both source and target eNodeB. This value respects what is considered the maximum level of resources utilization for LTE operators since 20% of the capacity is kept as safety for unexpected situations.

The background traffic is generated in uplink direction. The UEs attached to a cell are generating traffic following the traffic mix models specified in 4.2.2.1. Several experiments, using the parameters specified in Table 4.3, have been performed. The number of UEs attached to a single eNodeB has been increased by ten each experiment. The result can be seen in Figure 4.10.

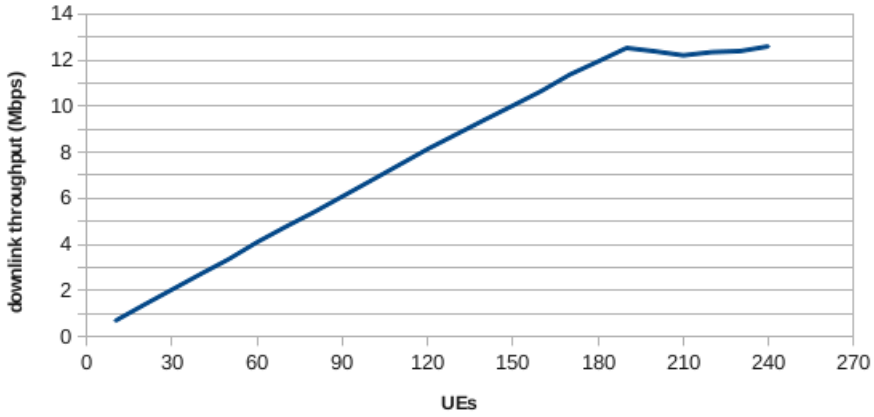


Figure 4.10: eNodeB's downlink throughput growth curve

From the figure it can be inferred that the saturation point of the cell is reached with a number of UEs greater than 180. When the cell is saturated the throughput level varies between 12.35 Mbps and 12.60 Mbps. Therefore the cell is estimated to be utilized at the 80% level of its total downlink capacity when the total traffic load is close to 10 Mbps. 150 UEs with an initialization time of around 4 seconds are needed to generate this amount of LTE traffic.

The amount of moving UEs of a simulation will be a part of the 150 active UEs initially attached to the source eNodeB, with the remaining UEs representing the LTE background traffic. The same amount of background UEs will also be initially attached to the target eNodeB in order to have enough space to host all the moving UEs during the remaining duration of the simulation.

The traffic generated by the moving UEs will be forwarded via the operator's transport network to the Internet, while the background traffic will be dropped at the eNodeB.

The background traffic implementation source code can be found in [71], while to drop this type of traffic at the eNodeB a filter has been implemented at the cell's application layer (*EpcEnbApplication* module). The implementation of such filter can be found in [60].

4.2.4.2 EPC/operator's IP transport and Internet network

The parameters setup for the wired IP network are summarized in Table 4.4.

Table 4.4: Wired network simulation parameters

Parameters	**Values**
Transmission technology	Ethernet
MTU	1.500 bytes
Backbone link data rate	10 Gbps
Internet link data rate	1 Gbps
S1-U link data rate	1 Gbps
S11/S1-AP link data rate	1 Gbps
SGi link data rate	1 Gbps
Queue scheme	Drop-tail
Backbone nodes buffer size	31.250 MB
EPC transport nodes buffer size	3.125 MB
Internet nodes buffer size	3.125 MB

The deployed network is a modern fast network with IP over 10-gigabit Ethernet implemented in the backbone (please refer to Figure 4.5, Section 4.2 for more details) and IP over gigabit Ethernet in both EPC transport and Internet networks.

NS3 offers two queue schemes: Drop-tail and RED. Due to its simplicity the standard Drop-tail scheme has been preferred.

For calculating the buffer size values, the same approach as the one defined by researchers of Stanford University [73], has been used. In 2005 D. Wischik and N. McKeown [74] have first defined a rule of thumb which is practically used by operators where the router buffer size in bit is equal to

$$C \times RTT$$

where C is the link speed and RTT is the round trip time of the flow. In their research they demonstrated that the values returned by the aforementioned rule of thumb are always too big compared to the effective needed buffer size.
Furthermore in 2006 Y. Ganjali and N. McKeown [75] defined a more precise formula to estimate the router's needed buffer size:

$$\frac{C \times RTT}{\sqrt{N}}$$

where N is the number of long-lived flows sharing the link.
Due to the fact that estimating the number of long lived flows on a Internet core network link is not trivial, they stated that the above formula can be generalized as

$$\frac{C \times RTT}{10}$$

This formula has been used to calculate the buffer size of all the routers within the wired IP network. For the purpose of this research a *RTT* value of 250 ms (as the one used in [74] and [75]) has been used in the calculations.

Link utilization Also for the wired network, 80% has been chosen as the maximum level of utilization for both EPC/operator's IP transport and Internet networks links (therefore for both links with 1 Gbps and 10 Gbps capacity). Having only an average 20% of the capacity available in the entire network means that the network is fully utilized because in reality it is difficult that network operators decide to exceed this limit due to safety reasons.

The 80% traffic is mainly generated using the PPBP application introduced in Section 4.2.2.2 plus a small load belonging to the end-to-end VoIP traffic generated by the moving UEs and their peer remote hosts.

For the 1 Gbps links, in order to generate ~800 Mbps of background traffic 40 bursts have been used. Each burst represent an aggregator generating 20 Mbps traffic for a mean period of 0.002 seconds. As suggested by [67], the Hurst parameter has been setup to 0.7.

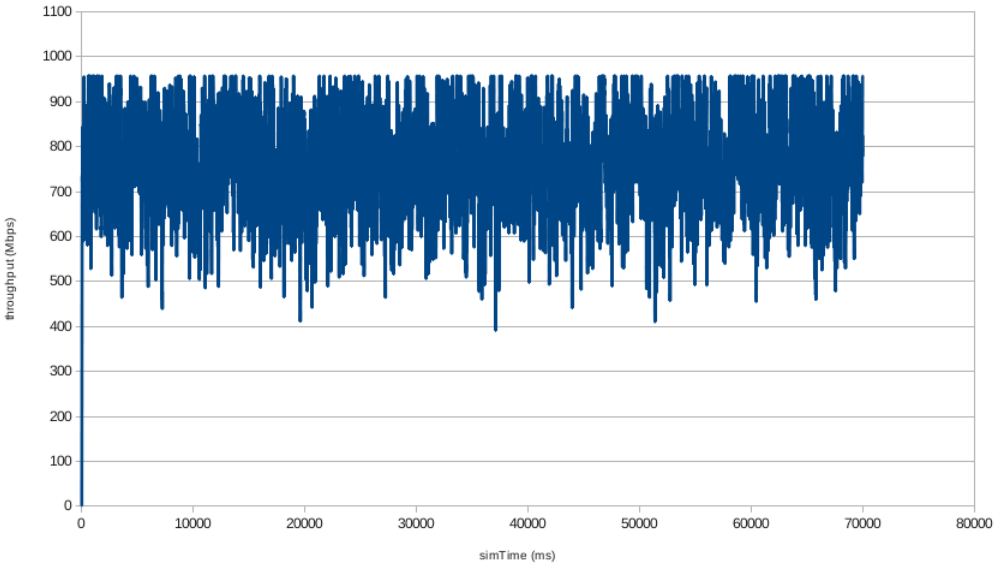


Figure 4.11: Throughput fluctuation on a 1 Gbps using NS3 PPBP application

The fluctuation of the throughput generated by the PPBP application on a 1 Gbps link using the parameters specified above is illustrated in Figure 4.11. The results shows that, although aggregator of

traffic are used to facilitate the execution of the simulation, a high throughput fluctuation is achieved during the entire 70 seconds simulation time. This is mainly due to the high bit-rate of each burst which together with the current number of active bursts (Pareto distributed) affect the rate at which packets are generated by the application.
The throughput samples have been measured every 0.01 seconds.

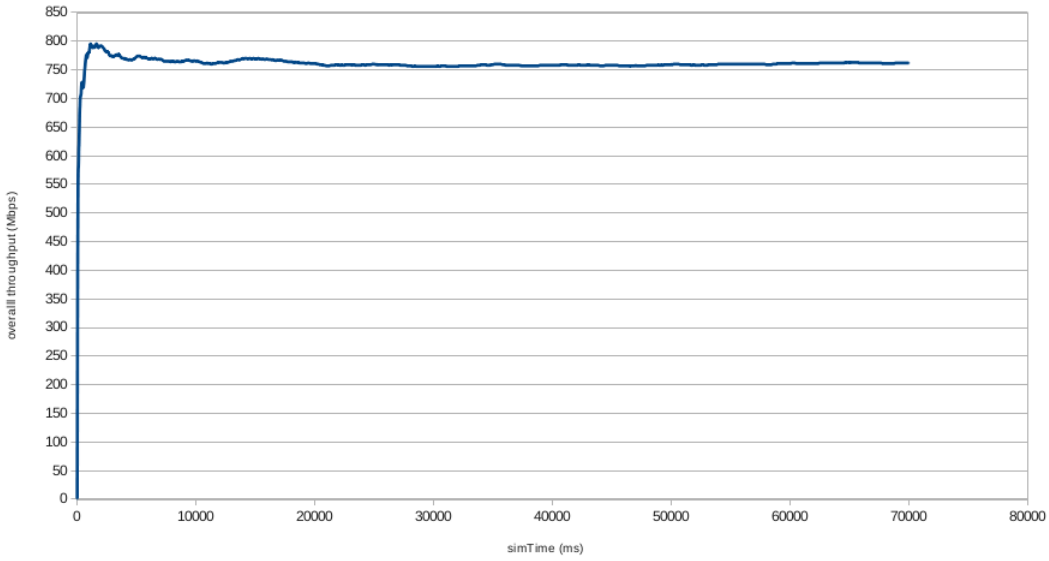


Figure 4.12: Overall throughput growth on a 1 Gbps using NS3 PPBP application

In order to determine the initialization time needed by the PPBP application (setup as above) to reach the desired background traffic data rate on a 1 Gbps link, the growth of the measured overall throughput has been plot in Figure 4.12.
The figure shows that after 10 seconds the overall throughput stabilizes at ~760 Mbps. The high bursts' bit-rate causes the application to reach the desired data rate quickly. Therefore the results of the simulation experiments are collected after an initialization time of 10 seconds.

The same procedure has been performed also for the 10 Gbps link. The same number of aggregators have been used but this time each one of them generates 200 Mbps traffic. The length of a burst follows again a Pareto distribution with mean 0.002 seconds and Hurst parameter set to 0.7.

The higher bursts' bit-rate results in a higher throughput fluctuation as depicted in Figure 4.13; while Figure 4.14 confirms that 10 seconds are enough to reach the stabilization point of the overall throughput.

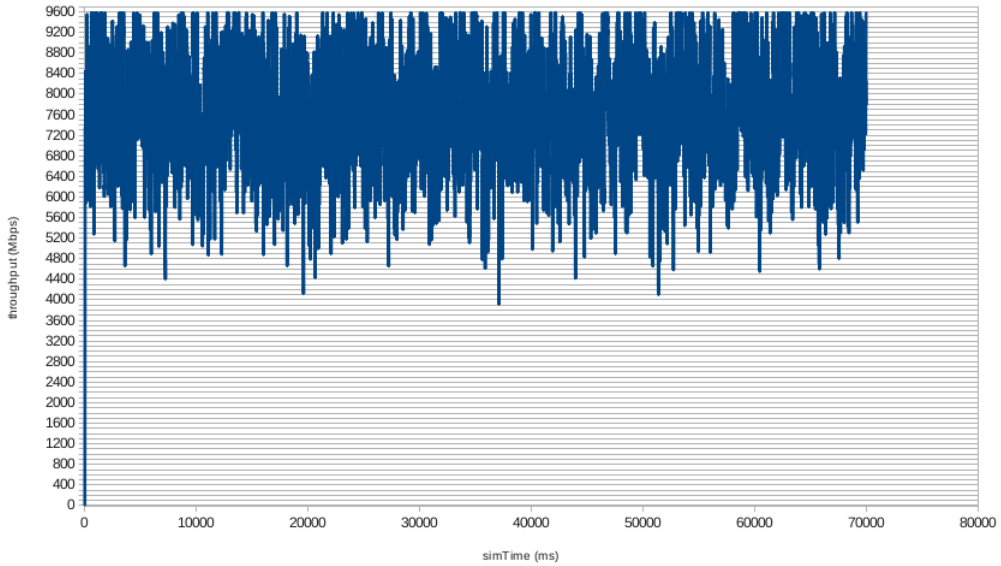


Figure 4.13: Throughput fluctuation on a 10 Gbps using NS3 PPBP application

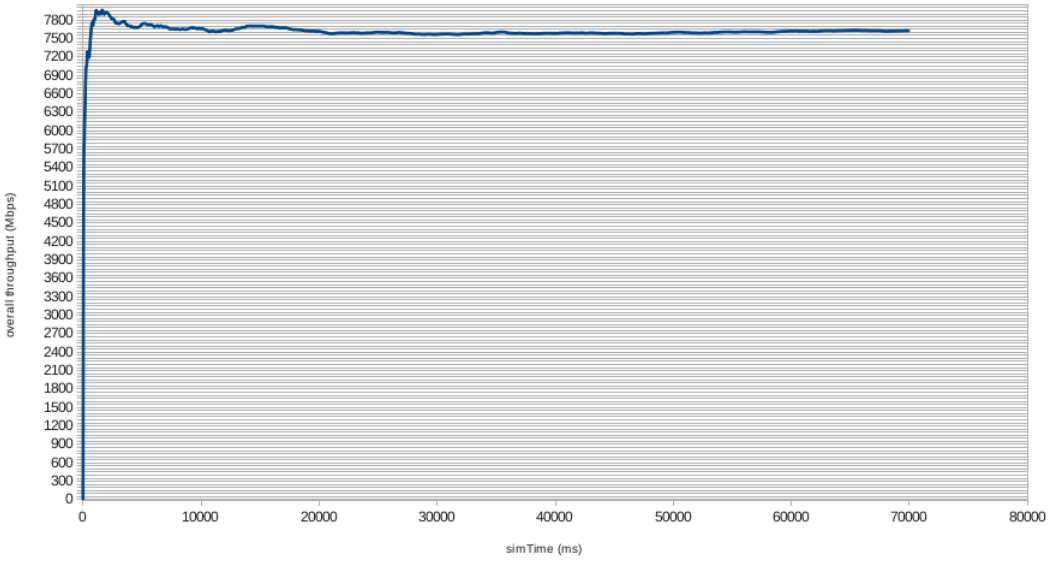


Figure 4.14: Overall throughput growth on a 10 Gbps using NS3 PPBP application

The parameters used to setup the PPBP application for generating background traffic corresponding to 80% capacity of both 1 Gbps and 10 Gbps links are summarized in Table 4.5.

Table 4.5: PPBP application parameters to generate 80% background traffic on 1 Gbps and 10 Gbps links

	H	T_{on}	λ_p	r	$E[n]$
1 Gbps link	0.7	0.002 s	2000	20 Mbps	40
10 Gbps link	0.7	0.002 s	2000	200 Mbps	40

4.2.5 Virtualization platform delay

One of the features of a virtualized network is the possibility to migrate content, functions and entities from one Virtual Machine (VM) to another within the same data center or from one data center to another, giving to operators the possibility to manage their network much more efficiently. Popular contents can be placed closer to the edge of the network in order to relieve the core network and offer faster access to end-users; functions can be remotely moved onto EPC entities in case of software or hardware upgrades or upon request from end-users; and mobile core, soft-EPC based, network components can be re-located to cope with dynamic network deployment (subscriber mobility, geographical distributions and traffic load) and failures.

In the simulation experiments a case of handover which requires the migration of the target S-/P-GW entity will be evaluated. This scenario can occur following a failure of the previous virtualized entity or a high fluctuation of the traffic demand in a particular geographical area.

Since no information is yet available on the procedures, resources and performances of a live VM migration whereas the VM implement the functionality of a S-/P-GW, data retrieved from literature will be used in the experiments.
M. Zhao et al. [7] collected the times needed to migrate VM entities from one host to another within the same data center. Their experiment considers VMs with different memory sizes to investigate the impact of size on migration times. The VM migration process considered in [7] entails of three phases, “suspend”, “copy” and “resume”. In the suspend phase, the VM is suspended on the origin host. In the copy phase, the VM’s configuration, memory state and disk files are transferred to the destination host through FTP. In the resume phase, the VM restores its memory state and then resumes its execution. Results on 8 different VM memory sizes are shown in Figure 4.15.

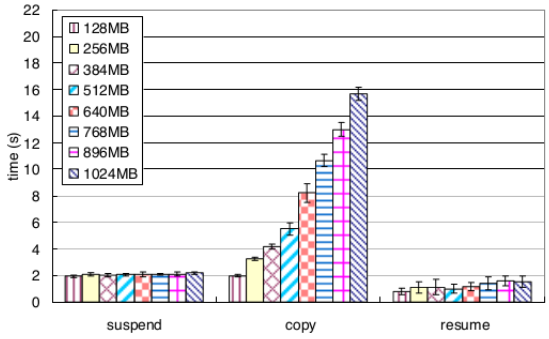


Figure 4.15: The time needed for the three phases of migrating a single VM with different memory sizes, copied from [7]

Data for different VM sizes are extrapolated from all three phases depicted in the graph above and used to emulate the migration of a S-/P-GW entity. The extrapolated values will be an additional delay of the X2 handover procedures, so that the impact of VM migration on UE’s mobility can be investigated. The "copy" phase depicted in Figure 4.15 refers to a local FTP transfer, therefore

the migration procedure is assumed to be performed on a platform separated from the operator's network. For this reason the delay introduced by the migration of an EPC entity/function is refer to as Virtualization Platform delay (or VP delay).
The VP delays used in the simulation experiments are summarized in Table 4.6.

Table 4.6: Virtualization Platform delays

VM size (MB)	VP delay (s)
128	4.46
256	5.65
512	9.03
1024	19.71

4.2.6 Confidence interval

In order to guarantee the reliability of the collected performance results, the performed experiments are repeated several times using different random seeds. The measured sample values from the experiment runs are used to compute the mean value of a metric with a confidence interval for this mean. The confidence interval is computed using a Student test or t-test [76].
Since the number of samples in the experiments is less than 30 samples, the following formula is used to calculate confidence interval:

$$x \pm t_{\frac{\alpha}{2},N-1}\left(\frac{\sigma}{\sqrt{N}}\right)$$

where, x is the sample mean, N is the number of sample, σ is the sample standard deviation, and $t_{\frac{\alpha}{2},N-1}$ is the upper critical value of the (Student) t distribution with $N-1$ degrees of freedom.

In the experiments, a confidence interval of 95% will be used, and the confidence interval should be less than 5% of the sample mean value.

4.3 Performance metrics

In the experiments several metrics will be used for performance evaluation. These metrics are presented as follows.

4.3.1 Average latencies of downlink data packets

Latency is an important metric to consider in the experiments since it is fundamental to define the seamlessness of a DMM solution. Four different average latencies of downlink data packet have been defined for the performed experiments. All average latencies are measured in milliseconds.

4.3.1.1 Average latency of data packet delivery prior to handover

This measurement represents the downlink path average delay of the data packets received by the moving UEs before the triggering of the inter S-/P-GW handover procedure. This metrics will be compared to latencies of packets forwarded via alternative paths which will be presented next.

It is measured using the following formula:

$$\frac{\sum(Rx_{prior} - Tx_{prior})}{N_{prior}}$$

where N_{prior} is the total number of downlink data packets received by the moving UEs before the triggering of the inter S-/P-GW handover procedure. Tx_{prior} is the time when the data packet has been transmitted by the remote host and its value is extracted from the timestamp information carried by the received VoIP packets, while Rx_{prior} is the time instant when the same packet is processed by the VoIP server application installed on the UE. Both Rx_{prior} and Tx_{prior} are read from a trace file generated at run-time by the server VoIP applications (see Section 4.2.2.1 for more details) and their difference represents the downlink delay of a data packet to reach the UE. Tx_{prior} of all packets directed to a specific UE must be smaller than the time when the handover of that particular UE is triggered.

4.3.1.2 Average latency of data packet delivery via X2 tunnel

This measurement represents the downlink path average delay of data packet(s) which are pushed from the source to the target eNodeB via the X2 path during inter S-/P-GW handover procedure. When the handover procedure is triggered in the E-UTRAN a link is established on the X2 interface between the source and target eNodeB. This link is used to forward both signaling and data traffic but only the delay of data packets will be used in the calculation of this metrics. Data packets are forwarded via the X2 path during the handover completion phase of the procedure described in Section 2.2.1. During this phase the UE has been already handed over to the target eNodeB (at the end of the handover execution phase) but its downlink packets are still being received by the source S-/P-GW and forwarded to the source eNodeB. This happens only if DMM traffic redirection has not been setup yet. Dependent on this, some handover procedures can not experience the forwarding of data via the X2 path.

This metric is measured using the following formula:

$$\frac{\sum(Rx_{x2} - Tx_{x2})}{N_{x2}}$$

where N_{x2} is the number downlink data packet(s) which are forwarded via the X2 path during inter S-/P-GW handover procedure. Tx_{x2} is the time when the data packet has been transmitted by the remote host and its value is extracted from the timestamp information carried by the received VoIP packets, while Rx_{x2} is the time instant when the packet is processed by the VoIP server application installed on the UE. The Rx_{x2} of packets directed to a specific UE must be greater than the time when the handover procedure for that particular UE is triggered. Both Rx_{x2} and Tx_{x2} are read from a trace file generated

at run-time by the server VoIP applications (see Section 4.2.2.1 for more details) and their difference represents the downlink delay of a data packet to reach the UE.
In order to select the Rx_{x2} and Tx_{x2} of data packets which have been forwarded via X2 path, information is extracted from such packets and written in a trace file upon their reception by the target eNodeB's *LteEnbRrc* application.

4.3.1.3 Average latency of first received DMM-redirected data packets

This measurement represents the downlink path average delay of the first data packet received by the moving UEs after the completion of the inter S-/P-GW handover procedure with X2 tunnel traffic forwarding. This metric will be compared with the average delay of packets forwarded via X2 paths, if present. The goal of the comparison will be to evaluate the performances of the X2 forwarding and DMM traffic redirection in similar LTE and wired traffic conditions.

It is measured using the following formula:

$$\frac{\sum(Rx_{first}-Tx_{first})}{N_{first}}$$

where N_{first} is the number of the first downlink data packets received by the moving UEs after the completion of the inter S-/P-GW handover procedure with X2 tunnel traffic forwarding. Tx_{first} is the time when the first data packet received after completion of the handover procedure has been transmitted by the remote host and its value is extracted from the timestamp information carried by the received VoIP packets, while Rx_{first} is the time instant when the same packet is processed by the VoIP server application installed on the UE. The Rx_{first} of the packet directed to a specific UE must be greater than the time when the handover procedure for that particular UE is triggered. Both Rx_{first} and Tx_{first} are read from a trace file generated at run-time by the server VoIP applications (see Section 4.2.2.1 for more details) and their difference represents the downlink delay of a data packet to reach the UE.

4.3.1.4 Average latency of data packet delivery after handover

This measurement represents the downlink path average delay of the data packets received by the moving UEs after the completion of the inter S-/P-GW handover procedure. This metrics will be compared to the average latency of packets received by the moving UEs before the trigger of the handover.

It is measured using the following formula:

$$\frac{\sum(Rx_{after}-Tx_{after})}{N}$$

where N is the number of downlink data packets received by the moving UEs after the completion of the inter S-/P-GW handover procedure. Tx_{after} is the time when the data packet has been transmitted

by the remote host and its value is extracted from the timestamp information carried by the received VoIP packets, while Rx_{after} is the time instant when the same packet is processed by the VoIP server application installed on the UE. Both Rx_{after} and Tx_{after} are read from a trace file generated at run-time by the server VoIP applications (see Section 4.2.2.1 for more details) and their difference represents the downlink delay of a data packet to reach the UE. Rx_{after} of all packets directed to a specific UE must be greater than the time when the handover of that particular UE is triggered and this packet must not be forwarded via the X2 path.

4.3.2 CDF of latency of first DMM-redirected data packets

The Cumulative Distribution Function (CDF) of latency of the first downlink data packets received by the moving UEs after the completion of the inter S-/P-GW handover procedure with X2 tunnel traffic forwarding is also calculated to identify what is the (approximately) measured maximum for this latency.

Being VoIP the studied UEs' traffic, a maximum one-way latency of 150 ms (as specified by ITU-T G.114 [77, 78]) is used as a threshold. Therefore the CDF will be used to demonstrate how the measured DMM-redirected data packets latencies fit with respect to the 150 ms threshold.

4.3.3 Throughputs

The throughput shows how much data is successfully transmitted over the LTE network. Throughputs of two types of traffic have been defined for the performed experiments.

4.3.3.1 X2 path throughput

It is calculated as the total number of the downlink data packets received by the UEs after have been forwarded via the X2 path over the simulation time.

The X2 path throughput is measured in packets/s (pps).

4.3.3.2 Total throughput

It is calculated as the total number of the downlink data packets received by the UEs over the simulation time.

The total throughput is measured in bits/s (bps).

4.3.4 Downlink Packet Loss Ratio

The packet loss ratio shows the reliability of the communication system. The downlink packet loss ratio is calculated using the following equation:

$$\frac{packets\_sent\_by\_remote\_hosts - packet\_received\_by\_UEs}{packets\_sent\_by\_remote\_hosts}$$

The number of packets is calculated at the application layer.
This metric will be used in the comparison between the three proposed DMM solutions.

4.3.5 Load of DMM signaling

Establishing traffic redirection in the DMM transport network requires a rather high number of message exchange between DMM Controller, Ingress and Egress functions and the EPC MME entity (see Sections 3.1.4 and 3.2.4). The signaling messages exchanged between EPS entities and specified in Section 2.2.1 are not taken into account for the calculation of the signaling load.
Although the size of a signaling message is usually very small (for both Double NAT and OpenFlow cases, only with IP and TCP headers it exceeds 100 Bytes), the total load of the signaling traffic can be used for a comparison with the amount of downlink data which is redirected via the path established by the implemented DMM solution.

It is calculated as the total transmitted signaling traffic (*Make Path* and *Rule Update* messages in the case Double NAT DMM or *Make Path* and *Modify-State* messages in the case of OpenFlow-based DMM solutions) over the simulation time.

In NS3 whenever a *Make Path*, a *Rule Update* or a *Modify-State* message is transmitted via the IP transport network, its size (with IP and TCP headers) is summed up to a global variable used at the end of the simulation to calculate this metric.

The total load of DMM-related signaling is measured in bit/s (bps).

4.4 Experiment scenarios

The three DMM solutions which have been evaluated in the experiments are:

- **Double NAT** based with single NAT Controller (see Section 3.1 for a detail description of this solution).
- **Partial OpenFlow** data transport network solution with single OpenFlow Controller (see Section 3.2.2 for a complete description of this solution).
- **Full OpenFlow** data transport network with single OpenFlow Controller and per-anchor point data forwarding scheme (see Section 3.2.1 for an extensive description of this solution and Section 3.2.1.2 for the description of the chosen data forwarding scheme).

The Double NAT and Partial OpenFlow solutions have been selected because of their similarities in both control plane and data plane deployment. In fact, where the same network topology is used,

the same number of Ingress and Egress functions need to be added to the network for both solutions. Address translation is used at the edges of the DMM transport network in both solutions to redirect downlink packets belonging to moving UEs. The only differences between them are the signaling traffic load (*Modify-State* messages are slightly bigger than *Rule Update* messages) and Ingress and Egress routers processing time.
Furthermore as already pointed out in a preliminary comparison carried out in Section 3.3, both solutions have a low impact on the current operator's transport network but OpenFlow-switches can offer a wider set of features while being more expensive to implement. It is therefore interesting to evaluate the performances of these two solutions taking into consideration the many similarities and few influential differences between them.

On the other hand the Full OpenFlow solution is a different type of investment for operators. The entire DMM transport network needs to be composed by *OpenFlow-full* switches. Whenever DMM traffic redirection has to be implemented to downlink flows, the OpenFlow switches' flow tables will need to be updated by the DMM Controller. Layer-3 routing is not present in the transport network and the forwarding path of each flow can therefore be controlled by the DMM Controller.
Although the OpenFlow specification ([47]) states that in a case of a network composed by *OpenFlow-full* switches the signaling is not to be run through the OpenFlow pipeline, in the simulation experiments the same transport network is used also to transport signaling traffic. This has been done to facilitate the comparison between the three DMM solutions.
Indeed a fully-based OpenFlow transport network provide operators with an extremely vast set of features but whereas performances are up to the level of the other two easy-to-deploy solutions is yet to be proven.

Two sets of simulation experiments have been defined:

1. X2 handover with S-/P-GW relocation: evaluation of DMM performances in mobility scenarios which require S-/P-GW relocation.

2. X2 handover with virtual S-/P-GW migration: evaluation of the impacts of migrating the virtualized target S-/P-GW entity or part of its functionalities during handover procedure.

The same EPC/operator's IP transport and Internet networks topology depicted in Figure 4.5 of Section 4.2 is used in both sets of experiments.

Furthermore the same LTE network's setup is used in all experiments. 150 UEs are initially attached to the source eNodeB and are generating traffic following the traffic models mix described in Section 4.2.2.1. The reason for choosing this amount of UEs is given by the fact that an 80% level of cell utilization is used in the simulations (see Section 4.2.4.1). Four fifth of the source eNodeB's active UEs (120) are considered static, i.e. they will remain attached to the source eNodeB for the entire simulation time. Their positions are uniformly distributed around the eNodeB in a disc with radius

equal to 2000 meters. The remaining 30 UEs (all of them running a VoIP application with a peer remote host) will be handed over to the target eNodeB during the execution of the simulation. For this reason these users are placed in positions at the same distance (2 km) from both source and target eNodeB.
120 UEs are initially attached to the target eNodeB and they are generating traffic using the same mix as the 120 static UEs attached to the source eNodeB. This is done in order to reach also a maximum level of cell utilization when all the UEs have been handed over from the source eNodeB. No UEs will be handed over from the target to the source eNodeB during the simulation.

Due to the fact that uplink traffic generated by static UEs is dropped at the eNodeB and that all the moving UEs are running a VoIP application upon handover, a number of remote hosts equal to the number of moving UEs are attached to the Internet network. Their traffic is assumed to enter the Internet network from the same edge points as the ones depicted in Figure 4.5 of Section 4.2.

Both set of experiments will be explained in details in the remainder of this Section after having defined which parameters will be varied in the simulation experiments.

4.4.1 Definition of parameters to be varied

4.4.1.1 Average distance (hops) between DMM Ingress and Egress points

DMM Ingress and Egress functionalities are implemented on core network routers. Being IP routing used in the entire network, each Egress router is attached to a S-/P-GW entity via the SGi interface. In this way downlink packets can be forwarded to the current UE's mobility anchor point even though the destination IP address is topologically unrelated with the S-/P-GW position.

The positioning of the routers implementing Ingress DMM functionalities do not have any particular constrains and it is mainly based on efficient network design. For this reason three different positions have been defined and are shown in Figure 4.6, 4.7 and 4.8 of Section 4.2.

In the first case (Fig. 4.6), DMM Ingress routers have been placed closer to the part of the network where the data centers running the virtualized EPC and E-UTRAN entities are located. The average hop count between DMM Ingress and Egress routers is 1.
Due to the fact that traffic redirection occurs only in the final hop of the IP transport network, this solution can be subjected to sub-optimal routing issues.

In the second case (Fig. 4.7) the DMM Ingress routers have been located at the same distance from the S-/P-GW entities and from the operator's Internet PoP. The average hop count between DMM Ingress and Egress routers is equal to 4 hops.

In the third case (Fig. 4.7) the DMM Ingress routers have been located as close as possible to the Internet network, on two of the three operator's Internet PoPs. The average hop count between DMM Ingress and Egress routers has increased to 7 hops.
This solution is expected to provide a more optimal routing to the DMM redirected traffic.

4.4.1.2 DMM Controller position

The placement of the DMM Controller in the operator's network can be critical. The controller receives UEs' mobility context information (UE's IP address, DMM forwarding IP address) from the EPC MME entity and used them in the signaling with the DMM Ingress and Egress functions to implement DMM traffic redirection. Three different positions have been defined for the DMM Controller and are show in Figure 4.9 of Section 4.2.1.

- position #1 (*MME co-located*): the DMM Controller is co-located with the EPC MME entity making the signaling between the two entities an intra-data center communication. The fact that signaling from the MME is delivered locally to the DMM Controller, can decrease the latency in setting up DMM traffic redirection.
- position #2 (*middle of CN*): the DMM Ingress and Egress control function is placed closer to the Egress part of the DMM transport network, precisely at an average hop distance of 1.5 from both Egress points. The data center where the virtualized MME entity is running is located two hops further.
- position #3 (*outside CN*): the DMM Controller is placed further away from the access network, closer to the operator's Internet PoPs. Precisely the DMM Controller is located in a position external to the operator's IP transport network. This can happen for instance if the DMM functionalities are provided to the operator by an external provider or if the simulation topology is only a slice of the entire operator's network. The distance of the DMM Controller from the MME and to the DMM Egress functions reaches the 9 hops with a mixed link speed (1 Gbps or 10 Gbps).

4.4.1.3 X2 forwarding

When the X2-based handover procedure is triggered, a path established on the X2 interface between the source and target eNodeB is used to forward both control and (if present) data plane messages as specified by the procedure introduced in Section 2.2.1.

Data packets are transmitted through the X2 path if they have been forwarded by the source S-/P-GW via the S1-U tunnel during the *handover execution* phase or if, after having established both uplink and downlink S1-U bearers between target eNodeB and target S-/P-GW (*handover completion* phase), the setup of DMM traffic redirection has not been completed yet. In the latter case downlink packets are still being received at the source S-/P-GW and, if the UE's session has not been deleted yet, they are delivered to the serving cell via the X2 path. Uplink packets are already forwarded via the newly formed S1-U tunnel.

The aforementioned procedure is already present in the latest NS3 LENA version (source code available at [60]) and it will be used in the simulation experiments. Besides this, the scenario when X2 data

forwarding is not available will also be evaluated in the experiments. This can be due to temporary lack of X2 connectivity between source and target eNodeB. Being the two eNodeB entities virtualized and running into separated data centers, the unavailability of X2 data forwarding can also be seen as a network outage issue. Evaluating the performances of a DMM solution in such scenario will give a clear indication of what is the impact of not having the X2 forwarding feature in the network.
Although the current 3GPP standard provides a solution to the lack of X2 capabilities in the form of the S1-based handover (a modification to support IP address continuity also in this type of handover has been described in Section 2.2.2), in the simulation experiments this feature has not been implemented and downlink data packets received in the source eNodeB during *handover execution* and *handover completion* phases will be dropped. Due to the fact that the current NS3 LENA version does not offer any alternatives to the control plane procedure of an X2-based handover, signaling traffic between source and target eNodeB will always be sent through the X2 path (it can be assumed that a third eNodeB is used as a relay for signaling traffic).

In the experiments where X2 forwarding is used, the X2 path is assumed to be already established between the two eNodeBs when handovers are triggered.

4.4.1.4 Delete Session Timer

As described in the previous subsection, X2 data forwarding can occur in both *handover execution* (in uplink and downlink direction) and *handover completion* (only downlink) phases (see Section 2.2.1 for detailed information).

When in the latter the source eNodeB checks every 1 millisecond if any packets have been added to the buffer kept for the moving UE. If present, this packet(s) will be forwarded to the target eNodeB via the X2 path. Since the required S11 signaling used to delete the UE's session(s) at the source S-/P-GW has not been implemented yet in NS3 LENA, a timer is used to avoid that the X2 path remains open forever. This timer (named for convenience as Delete Session Timer) can be setup manually by the user and it reflects the functionality of the timer used in the 3GPP's standard by the MME to delete the moving UE's session(s) in the source S-/P-GW (see step 4 of the procedure described in Section 2.2.1). The importance of the Delete Session Timer is given by the fact that if DMM traffic redirection setup is not completed upon its expiration, downlink data packets might be lost when received at the source S-/P-GW where no UE session is present anymore. The choice on how long keep the UE's session(s) active at the source S-/P-GW is therefore left to operators. In the simulation experiments a value of 10 ms has been setup.

If operators do not wish to use X2 downlink data forwarding in the *handover completion* phase (e.g. to save resources in both eNodeBs) the Delete Session Timer has to be setup to 0 ms. In this way the moving UE's session(s) will be removed in the source S-/P-GW as soon as the path switch procedure has been completed.
This scenario is also simulated in the experiments in order to evaluate the level of X2 data forwarding

usage in the *handover completion* phase and the impacts of not having this extra functionality.

4.4.1.5 Virtualization Platform delay

Migrating an EPC entity (target S-/P-GW) upon UEs mobility is emulated in the simulation experiments by delaying the completion of the handover procedure for a fixed amount of time. Since the migration procedure is assumed to be performed on a Cloud Computing platform (e.g. OpenStack [79]) and between two data centers, which is not using the virtualized EPC network, the introduced latency is referred to as Virtualization Platform delay (or VP delay). The migrated VM entity is not transmitted via the core network since the impact of this transfer is out of the scope of this research.

The goal of investigating the impact of the addition of a VP delay to the handover procedure will be to estimate the anticipation time needed to initiate a VM migration before the triggering of the handover by the access network in order to offer seamless mobility to UEs. At the end of the migration procedure, on-going user sessions belonging to moving UE will need to be present in the migrated VM entity.

The VP delays used in the simulation experiments are summarized in Table 4.6 of Section 4.2.5.

4.4.2 First set of experiments: X2 handover with S-/P-GW relocation

In this set of experiments, simulations are run for 16 seconds where 10 seconds are used to initialize both access and wired networks' background traffic using the methods introduced in Sections 4.2.2.1 and 4.2.2.2. During the remaining 6 seconds, 30 UEs are handed off from source to target eNodeB. Despite the name, all moving UEs are statically located at the same distance (2 km) from both cells for the entire simulation time. The time when each of the 30 handovers is triggered is given by a distribution of time as described in Section 4.2.3.

Each of the 30 moving UEs has initiated, prior to handover, a VoIP session with one of the 30 remote hosts attached to the Internet network. These VoIP sessions remain active until the end of the simulation.

Each handover will require the relocation of the serving S-/P-GW. UEs sessions will need to remain active after the completion of the handover procedure. Therefore UEs context informations need to be transferred to the target eNodeB and target S-/P-GW. While for the former X2 signaling will always be used, the MME will manage the establishment of the EPS bearers in the target S-/P-GW. The procedure described in Section 2.2.1 is used for this purpose.

When X2 handover is completed DMM traffic redirection will be implemented in the network to forward the moving UEs' downlink data packets towards the target S-/P-GW using the proposed DMM solutions (see the upcoming Sections 4.4.2.1 and 4.4.2.2).

During and after handover, the X2 path between source and target eNodeBs can be used to forward downlink data packets still being received at the source S-/P-GW.

To verify the seamlessness of the implemented DMM solution, the metrics introduced in Section 4.3 will be evaluated. UEs' downlink flows have not to experience an high growth in latency prior, during and after handover for the DMM solutions to be considered seamless. Being VoIP the studied UEs' traffic, a maximum one-way latency of 150 ms (as specified by ITU-T G.114 [77, 78]) is used as a threshold.

For the Double NAT and Partial OpenFlow solutions, the experiments are performed using all three network topologies introduced in Section 4.2.1; for the Full OpenFlow case only the topology depicted in Figure4.8 of Section 4.2.1 is used in order to have a higher density of *OpenFlow-full* switches. The position of the DMM Controller is varied in each of the network topologies following the three placements specified in Section 4.4.1.2. A total of 9 experiments topologies have therefore been defined for the Double NAT and Partial OpenFlow experiments while 3 experiments topologies are used in the Full OpenFlow experiments.
In order to investigate the impact that the absence of X2 data forwarding can have on the moving UEs' flows, this scenario has also been simulated. Furthermore the case when the Delete Session Timer is setup to 0 ms (i.e. no X2 data forwarding during *handover completion* phase) is also simulated in all topologies. When not specified the Delete Session Timer has been setup to 10 ms.

In order to guarantee the reliability of the collected performance results, the performed experiments are repeated several times using different random seeds.

The disconnection of UEs' flows is not studied in this research.

4.4.2.1 Double NAT and Partial OpenFlow

When the X2-handover is completed, the MME will signal the DMM Controller (either NAT Controller or OpenFlow Controller) in order to implement the correct DMM traffic redirection. The DMM Controller will setup the NAT tables or flow tables of the DMM Ingress and Egress routers as described in the message flows introduced in Section 3.1.4 for Double NAT, and Section 3.2.4 for Partial OpenFlow. The DMM Controller has to maintain a number of states equal to the number of DMM Ingress and Egress points present in the network.

4.4.2.2 Full OpenFlow

When the X2-handover is completed, the MME will signal the DMM Controller (OpenFlow Controller) in order to implement the correct DMM traffic redirection. The DMM Controller will setup the flow tables of all the interested OpenFlow switches on the downlink path towards the target S-/P-GW. A *Modify-State* message, as specified at Appendix A.3.4 in [47], is used to modify the switches output ports for data packets of flow(s) belonging to moving UEs. The DMM Controller has to maintain a number of states equal to the number of *OpenFlow-full* switches present in the DMM transport network.

4.4.3 Second set of experiments: handover with virtual S-/P-GW migration

This set of experiments is used to demonstrate that mobility prediction is needed in the network to provide seamless mobility to UEs which are handed over to a target cell requiring the migration of the serving S-/P-GW virtualized entity.

The migration of the virtualized target S-/P-GW entity, following the trigger of an X2 handover procedure by the access network, can give birth to two different scenarios:

1. UEs are handed over to the target eNodeB when the VM migration has been completed and the virtualized target S-/P-GW entity is correctly setup and in working state;
2. UEs are handed over to the target eNodeB without waiting the completion of the VM migration.

The first scenario is identical to what has been described in the previous Section (4.4.2) with the only difference that the actual X2 handover's trigger is postponed by a delay equals to the time needed to migrate the VM implementing the target S-/P-GW functionalities. For this reason this scenario can be considered as already addressed by the set of experiments described in Section 4.4.2.

The second scenario can refer to the case when UEs are forced to be handed over to a target cell which momentarily does not have any functional serving S-/P-GW. A VM implementing S-/P-GW functionalities needs to be migrated to a location closer to the edge of the network, in a data center which can be reached by the micro data center implementing the target eNodeB virtual entity. The migrated VM is not transferred via the operator's transport network since the migration is assumed to happen on a separated platform. For this reason the signaling and operations needed to migrate a VM are not discussed in this report and have been implemented in the simulations only as an additional delay to the handover procedure, named as Virtualization Platform delay (or VP delay).
Since the target eNodeB does not have any serving S-/P-GW upon handover termination, UEs' uplink and downlink traffic will be forwarded to the data center where the target S-/P-GW entity is migrated to, which in the following will be referred to as target data center. Uplink and downlink packets are buffered in the target data center and locally delivered to the target S-/P-GW virtualized entity once that the migration procedure is completed.
The requirements, design and implementation of the buffer used to store UEs' uplink and downlink data packets within the target data center and the mechanisms to deliver the buffered data to the migrated target S-/P-GW virtual entity are not discussed in this report.

A set of experiments has been defined to simulate the solution described above. The duration of each simulation is dependent on the used VP delay. The VM's sizes and correspondent VP delays used in the simulation experiments are summarized in Table 4.6 of Section 4.2.5.
10 seconds are used to initialize both access and wired networks' background traffic using the methods introduced in Sections 4.2.2.1 and 4.2.2.2. When the network has been initialized a single UE is

handed off from source to target eNodeB. Despite the name, the moving UE is statically located at the same distance (2 km) from both cells for the entire simulation time.
The moving UE has initiated, prior to handover, a VoIP session with one of the 30 remote hosts attached to the Internet network. This VoIP session remains active until the end of the simulation.
UE's context information is transferred to the target eNodeB and target S-/P-GW to keep UE's session(s) active upon handover. The procedure described in Section 2.2.1 is used for this purpose. It is assumed that the target S-/P-GW's S11 reference point and the logical part that manages the S-/P-GW signaling are already present in the target data center when handover is triggered.
In the simulation experiments the VP delay is emulated by introducing a fix propagation delay in the X2 link between source and target eNodeB. In this way X2 signaling required to complete the UE's radio connection to the target eNodeB will be delayed and thus the handover.
DMM traffic redirection is setup in the operator's transport network as soon as the handover procedure is triggered following the same procedures described in Section 4.4.2.1 for Double NAT and Partial OpenFlow and Section 4.4.2.2 for Full OpenFlow. Downlink data packets are then forwarded to the S-/P-GW node which assumes the role of the target data center while the virtualized S-/P-GW entity migration is underway. Received downlink packets are then buffered at the target data center and delivered to the UE once that the delayed handover procedures have been completed.

To verify the impact of VM migration on the seamlessness of UE mobility which require anchor point relocation, the average latency of the first downlink data packet received from the target eNodeB will be evaluated. This packet is redirected to the target S-/P-GW using the proposed DMM solutions.

The experiments are performed using the network topology depicted in Figure 4.8 of Section 4.2.1. The DMM Controller is positioned in what has been referred to as *MME co-located* position (or position #1) in Section 4.4.1.2 since this position gives a lower latency in the signaling between MME and the DMM Controller. The Delete Session Timer has been setup to 10 ms.

In order to guarantee the reliability of the collected performance results, the performed experiments are repeated several times using different random seeds.

The disconnection of UE's flow(s) is not studied in this research.

Chapter 5

Simulation Results and Analysis

This Chapter presents the simulation results and analysis for the conducted experiments.

5.1 X2 handover with S-/P-GW relocation

This set of experiments aims to demonstrate the seamlessness of the proposed DMM solutions in case of UEs handed over from a source eNodeB to a target eNodeB using the X2 reference point. In this scenario the S-/P-GW entity is relocated while the serving MME remains unchanged. IP flows initiated by UEs when attached to the source eNodeB are kept active upon movement and IP address continuity is implemented in the EPS using the procedure introduced in Section 2.2.1.
Since the IP address of flows kept active upon handover is topologically unrelated with the current UEs' S-/P-GW, traffic redirection is implemented using the DMM solutions proposed in Chapter 3. The performance metrics introduced in Section 4.3 are used to analyze the performances of Double NAT, Partial OpenFlow and Full OpenFlow DMM solutions and the results are presented in this Section. A comparison between the three solutions is also carried out at the end of this Section.

The readers are advised to refer to Section 4.4.2 for a complete definition of this set of experiments.

5.1.1 Double NAT

5.1.1.1 Average latencies of downlink data packets

Figure 5.1 shows the average latency of downlink data packet delivery before and after handover. When handover is completed, Double NAT is used to redirect the traffic to the current UEs' mobility anchor point (target S-/P-GW). The results from different distances (number of hops) between Ingress and Egress NAT routers are shown in the graph. Since no impact is given to these metrics by the position of the NAT Controller in the network and the value of the Delete Session Timer, the shown

results refer to the case when the NAT Controller is placed in the middle of the operator's core network and the Delete Session Timer is set to 10 ms.

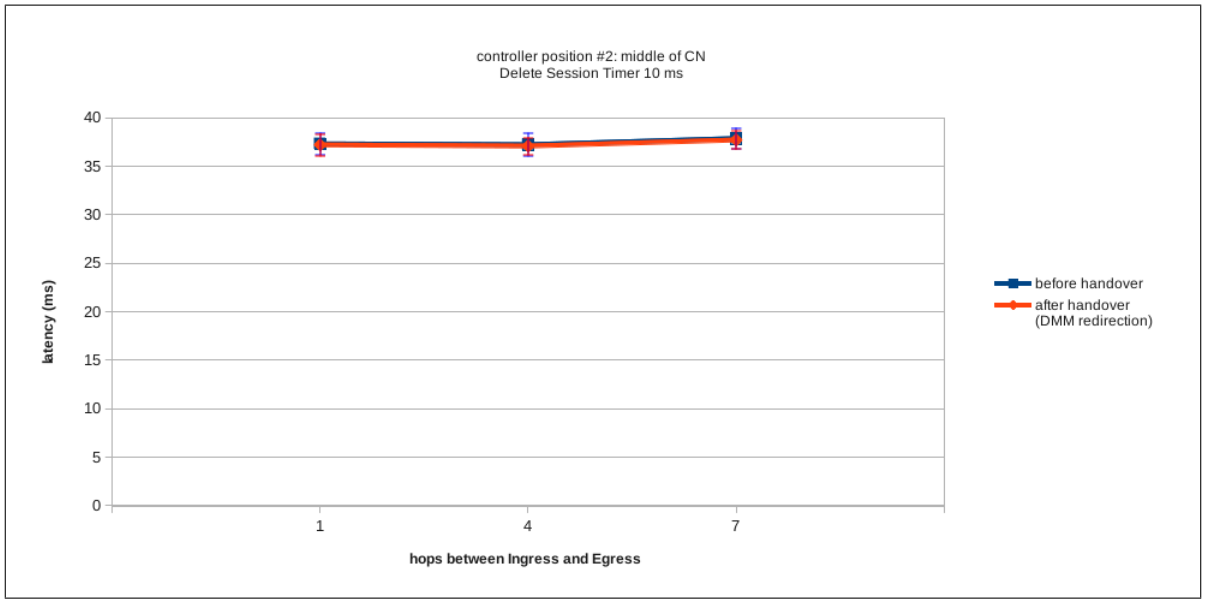


Figure 5.1: Average latency of downlink data packet delivery before and after handover. Double NAT used to redirect the traffic after handover.

The graph clearly demonstrates that DMM traffic redirection has no impact on the latency of downlink data packet delivery in the network topology used in the simulation experiments. Furthermore no variations were also observed in the collected results when the distance between Ingress and Egress NAT routers varies.

The observed results were expected since the used operator's transport network topology do not offer the possibility to exploit sub-optimal routes when the UEs' downlink traffic is redirected.

Figure 5.2 shows the average latencies of downlink data packets delivered via the X2 path and via DMM traffic redirection to moving UEs. The results from different distances (number of hops) between Ingress and Egress NAT routers are shown in the graph. The position of the NAT Controller is also varied, since it may impact the load of X2 traffic. Results are shown for the case when the Delete Session Timer has been setup to 10 ms, in order to have the maximum throughput in the X2 data path.

Although X2 data forwarding is a useful feature introduced in 3GPP standards to provide seamless mobility to UEs that move between neighboring cells (only if the PDN GW is not to be relocated, i.e. intra-operator handover), the graph shows that independently from the distance between Ingress and Egress points and the positioning of the NAT Controller, DMM traffic redirection outperforms X2 data forwarding. Therefore if a function to predict the mobility of UEs would be present in the network, DMM traffic redirection can be setup prior to the trigger of the handover procedure avoiding the usage

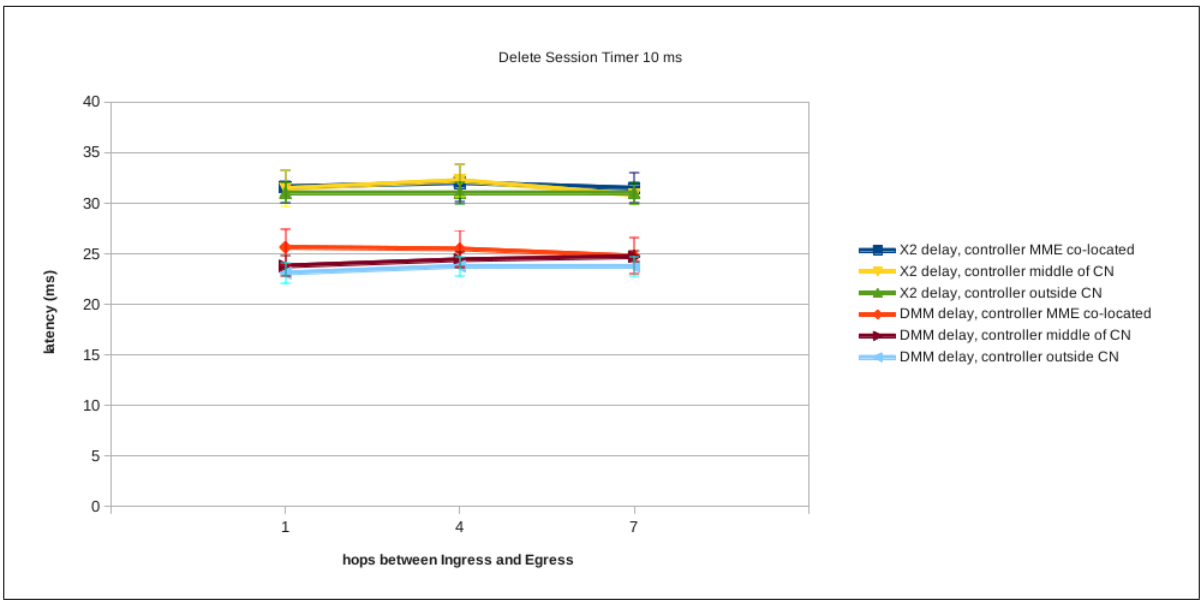


Figure 5.2: Average latency of downlink data packets delivered via X2 path and of the first packet redirected through Double NAT-based DMM transport network

of X2 data forwarding. In this way resources can be saved by operators in both RAN and CN.

5.1.1.2 CDF of latency of first DMM-redirected data packet

The Cumulative Distribution Function (CDF) is a useful tool to state with which probability an observed metric is above or below a certain threshold. To demonstrate the seamlessness of Double NAT as a DMM solution, the CDF of latency of the first DMM-redirected data packet after the completion of the handover procedure is shown in Figure 5.3.

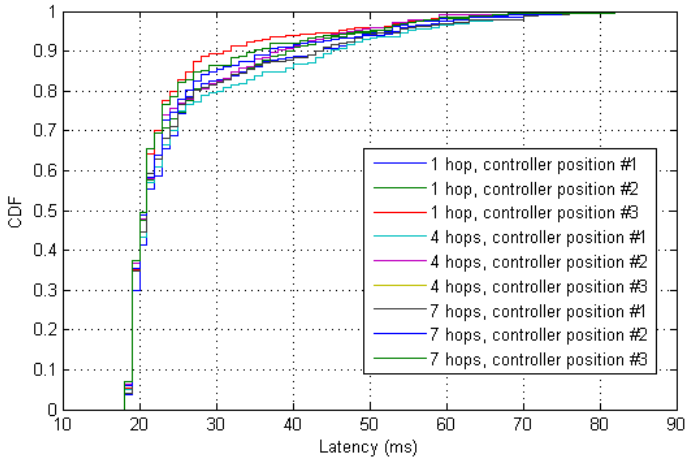


Figure 5.3: CDF of latency of first downlink data packets redirected through Double NAT-based DMM transport network

Being VoIP the studied UEs' traffic, 150 ms is accepted as a maximum one-way latency. The CDF in Figure 5.3 clearly shows that in all studied scenarios there is a 100% probability of having DMM-redirected data packets which do not exceed the maximum latency threshold of a VoIP session. Furthermore 90% of the observed packets has a latencies below the 50 ms which is commonly used as the backbone providers SLA maximum latency for VoIP traffic [80]. Being the measured metrics an end-to-end latency, it can be asserted that surely 90% of the observed latencies do not exceed also the VoIP providers SLA maximum latency.

5.1.1.3 Throughputs

In order to compare the impact of placing the Ingress NAT routers and Egress NAT routers at different distances (number of hops) with respect to the position of the NAT Controller, the throughput of data traffic forwarded via the X2 path is studied. In fact when DMM traffic redirection has not been setup yet upon completion of the UE radio handover, downlink data traffic will still be delivered to the source eNodeB which will then forward it to the target eNodeB via the X2 tunnel. In other words the load of the X2 forwarded traffic gives a good indication of the time required to setup DMM traffic redirection with respect to the positions of both Ingress NAT routers and NAT Controller.

Data forwarding via the X2 path can occur during the *handover execution* and the *handover completion* phase. Whether X2 data forwarding during the *handover execution* phase is not a choice for operators, setting the Delete Session Timer to 0 ms gives them the possibility to dismantle the X2 tunnel as soon as the path switch procedure has been completed. In standard condition the Delete Session Timer is setup to 10 ms. In order to observe the impact that this choice can have on the UE downlink traffic, results from both scenarios have been collected.

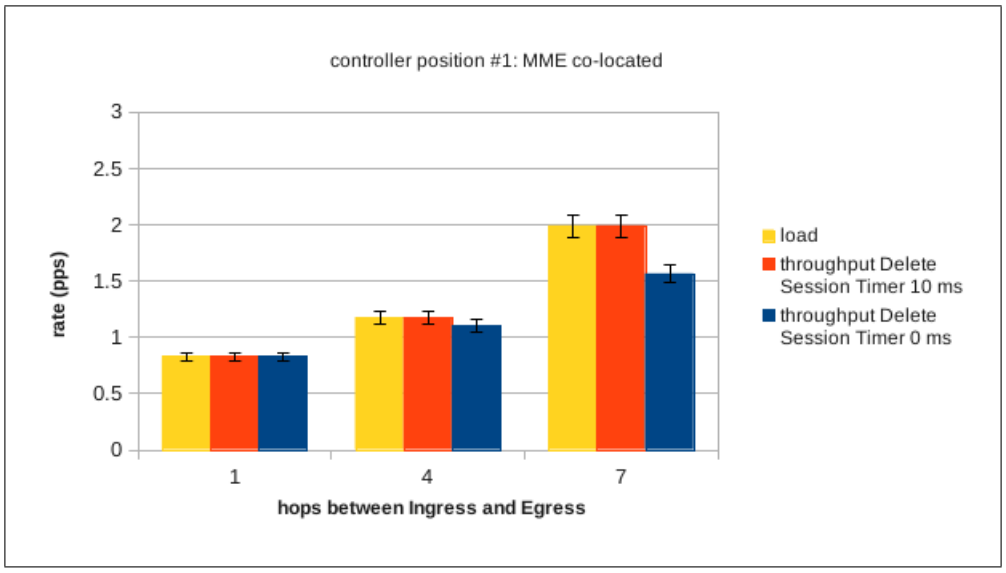


Figure 5.4: Load and throughput of X2 tunnel with NAT Controller co-located with MME

Figure 5.4 shows that the utilization of the X2 tunnel increases when the distance between the Ingress and Egress NAT routers increases. This is given by the fact that in this particular scenario the NAT Controller is co-located with the MME entity which is placed in the same part of the network where the data centers running the virtualized EPC entities are. As explained in Section 4.2.1, the Egress NAT routers have a fixed position as the next hop router on each S-/P-GW SGi interface in all simulated scenarios. Therefore when the number of hops between Ingress NAT routers and Egress NAT routers increases, so does the distance between the NAT Controller and the Ingress NAT routers and thus the time needed to setup DMM traffic redirection.

Furthermore, if the latency in setting up the NAT tables of both Ingress and Egress NAT routers increases, so does the probability of having X2 data forwarding after the completion of the path switch procedure. This situation can be observed in Figure 5.4 where the middle bar of each series represent the X2 tunnel throughput when the Delete Session Timer is setup to 10 ms while the last bar of each series when the Delete Session Timer is setup to 0. In all three topologies no packet loss is present if the Delete Session Timer is setup to 10 ms, meaning that those 10 extra ms are enough to cope with the latency needed to setup DMM traffic redirection. Although only few, some packets are lost when the Delete Session Timer is setup to 0 indicating that the decision on whether or not use X2 data forwarding after the completion of the path switch procedure is dependent on the size of the network and the approximate distance from the Ingress and Egress NAT routers to the NAT Controller.

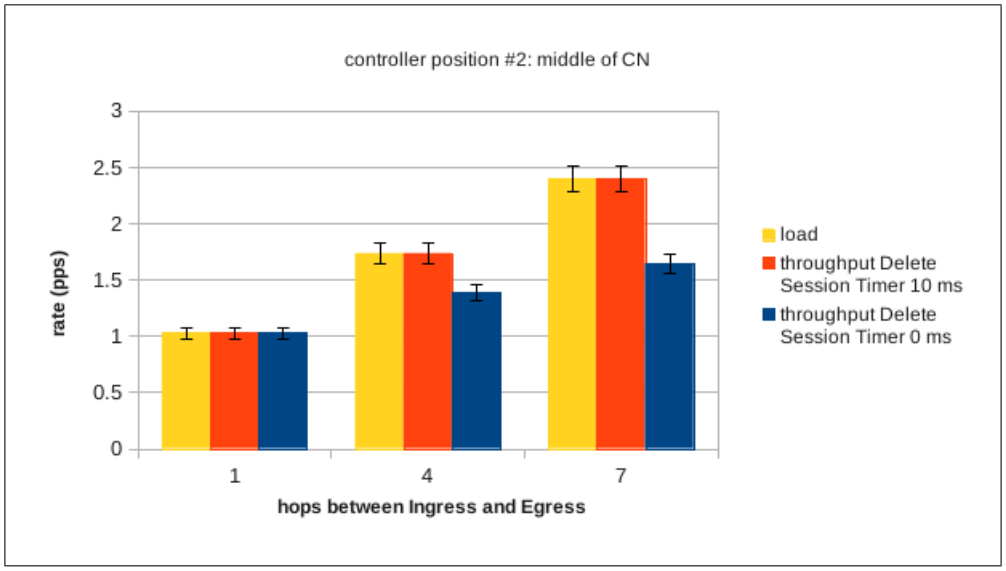


Figure 5.5: Load and throughput of X2 tunnel with NAT Controller positioned in the middle of the core network

A similar behavior has been observed for the case when the NAT Controller is located in the middle of the operator's transport network and the collected results are shown in Figure 5.5. As happened in the previous scenario, when the distance between Ingress and Egress NAT routers increases so does the latency in setting up DMM traffic redirection and thus the load of the traffic forwarded via the

X2 tunnel. A slightly bigger amount of packets are lost when the Delete Session Timer is setup to 0 ms, hinting that overall the latency in creating DMM traffic redirection is increased but 10 extra milliseconds have been proven to be enough to correctly deliver all the received downlink packets via the X2 tunnel once that the path switch procedure has been completed.

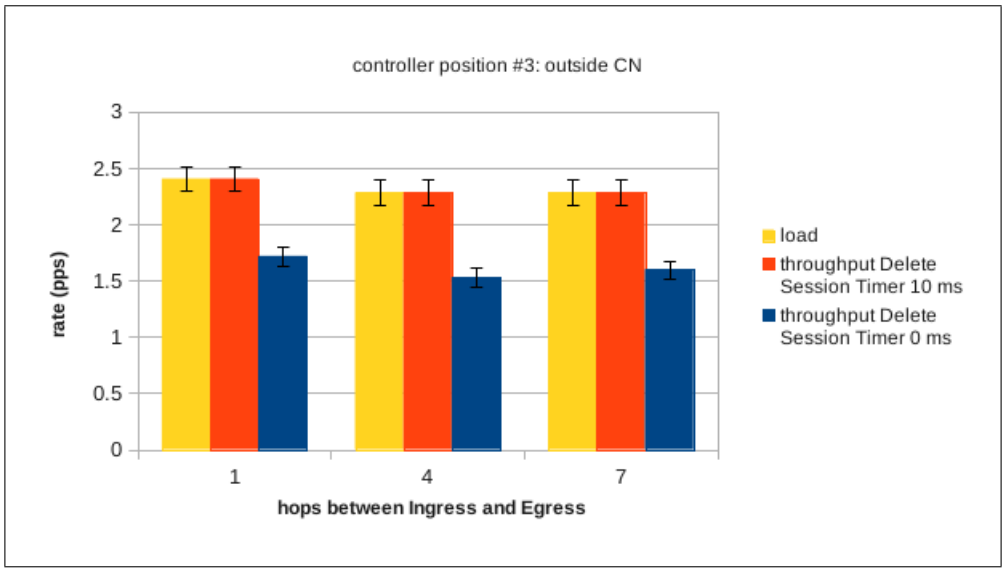


Figure 5.6: Load and throughput of X2 tunnel with NAT Controller positioned outside of the operator's transport network

Following the trend observed so far, the load of traffic forwarded via the X2 tunnel reaches the same level in all network topology when the NAT Controller is placed outside the operator's transport network, as it can be seen from Figure 5.6. When the Ingress NAT routers are positioned close to the EPC components or halfway in the operator's core network, their distance to the NAT Controller increases substantially and thus the latency in setting up DMM traffic redirection.

Once again the 10 extra milliseconds have been proven to be enough to correctly deliver all the received downlink packets via the X2 tunnel once that the path switch procedure has been completed.

To study the impact that the lack of X2 data forwarding capability can have on the system in the studied scenarios, the total load and throughput of the system have been collected both for the case when X2 data forwarding is available and the UE session is kept active in the source S-/P-GW also during *handover execution* phase and for the case when X2 data forwarding is not used during handover. Figures 5.7, 5.8 and 5.9 show the results with different NAT Controller positions.

In all cases few packets are lost when they are not forwarded via the X2 path with the gap between the throughput of the system with X2 data forwarding and the throughput of the system without X2 data forwarding increasing when the NAT Controller is placed more further away from the EPC part of the network. These results come as a confirmation on what has been observed studying the throughput

of the X2 path, demonstrating that the latency in setting up DMM traffic redirection is higher in the scenario where the NAT Controller is placed outside of the operator's core network thus increasing the probability of having traffic forwarded via the X2 path.

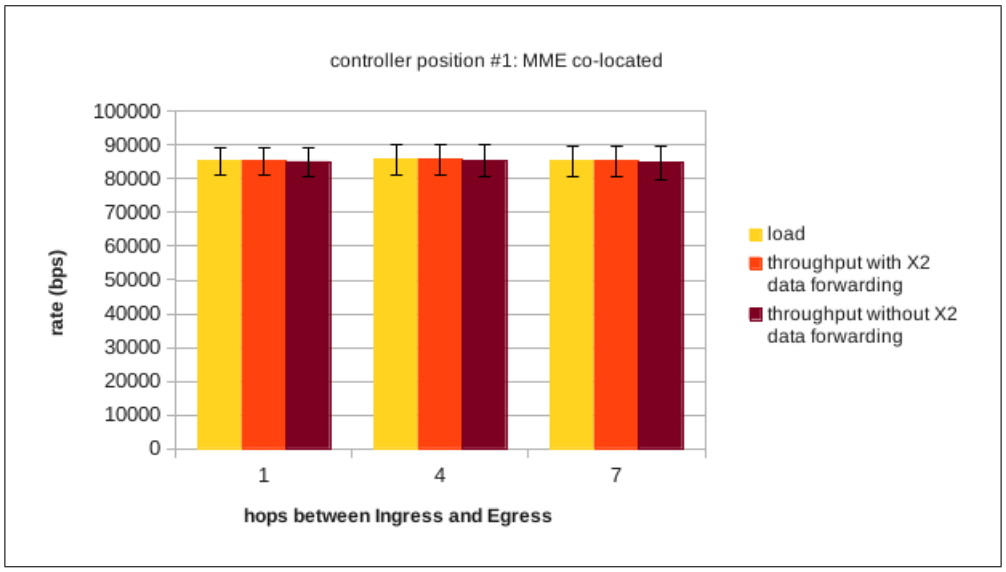


Figure 5.7: Load and total throughputs of the system with and without X2 data forwarding capabilities with NAT Controller co-located with the MME

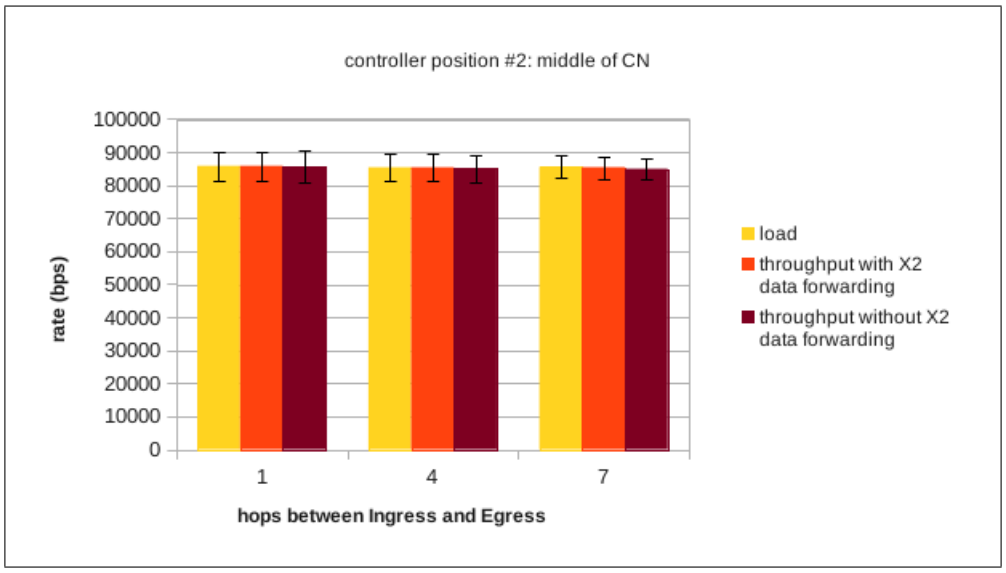


Figure 5.8: Load and total throughputs of the system with and without X2 data forwarding capabilities with NAT Controller positioned in the middle of the core network

As already suggested, in order to offer seamless mobility to UE also in the case when no X2 data forwarding is available in the access network, a function to predict the mobility of UEs should be

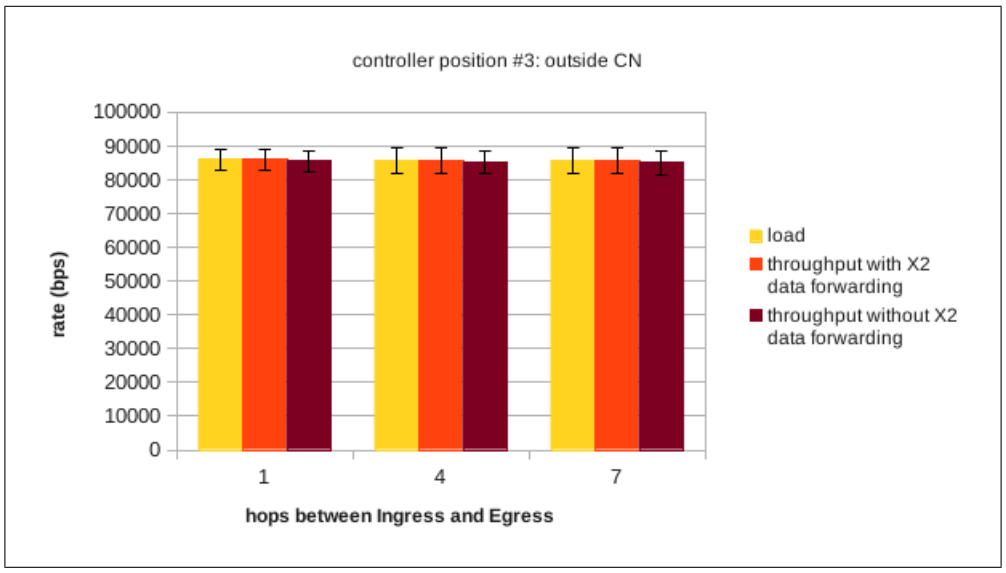


Figure 5.9: Load and total throughputs of the system with and without X2 data forwarding capabilities with NAT Controller positioned outside of the operator's transport network

present in the network to setup DMM traffic redirection prior to the trigger of the handover procedure. Moreover since the UE session(s) have not been setup yet in the target eNodeB and target S-/P-GW until the conclusion of the *handover completion* phase a buffering mechanism needs to be implemented in the Ingress or Egress NAT routers to store downlink data packets directed to the moving UE. Once that the handover procedure has been completed, the buffer data traffic can be forwarded via the implemented DMM redirected path. In this way resources can be saved in the source and target eNodeB since buffering of downlink data traffic occurs in the transport network above the EPS. Indeed packets will not need to be reorder in the target eNodeB which is what happens in current standard procedure to cope with packets delivered via the X2 path. Overall this procedure can increase the total throughput of the system while decreasing the latency of downlink data packets and saving resources in both RAN and CN.

5.1.1.4 Summary

Double NAT has been proven to offer seamless mobility to all moving UEs in all the studied network topology deployments. The best results in terms of latency in configuring traffic redirection in the DMM transport network were obtained when the NAT Controller was co-located with the MME entity due to the low latency overhead of signaling between the EPC and DMM-plane.

When X2 data forwarding was implemented in the network a Delete Session Timer of 10 ms has been always proven to be enough to empty the source eNodeB's X2 buffer before that the setup of DMM traffic redirection was completed. Furthermore depending on the deployment of the DMM transport network, not using X2 data forwarding after the conclusion of the path switch request procedure can

cause some packets to be lost and therefore compromising the seamlessness of the mobility procedure to the UEs.

As expected when X2 data forwarding has been disabled from the network, packet loss was introduced. Double NAT-based DMM traffic redirection can be considered as an alternative to the current X2 forwarding scheme. For this purpose, a function to predict the upcoming mobility of UEs needs to be implemented in the network. This function will trigger the setup of DMM traffic redirection prior to the trigger of the radio handover procedure. To cope with the fact that UE session(s) are not yet available in the target S-/P-GW during the execution of the handover procedure, a buffering mechanism is required in the DMM transport network.

5.1.2 Partial OpenFlow

5.1.2.1 Average latencies of downlink data packets

Figure 5.10 shows the average latency of downlink data packet delivery before and after handover. When handover is completed, Partial OpenFlow is used to redirect the traffic to the current UEs' mobility anchor point (target S-/P-GW). The results from different distances (number of hops) between Ingress and Egress OpenFlow switches are shown in the graph. Since no impact is given to these metrics by the position of the OpenFlow Controller in the network and the value of the Delete Session Timer, the shown results refer to the case when the OpenFlow Controller is co-located with the MME and the Delete Session Timer is set to 10 ms.

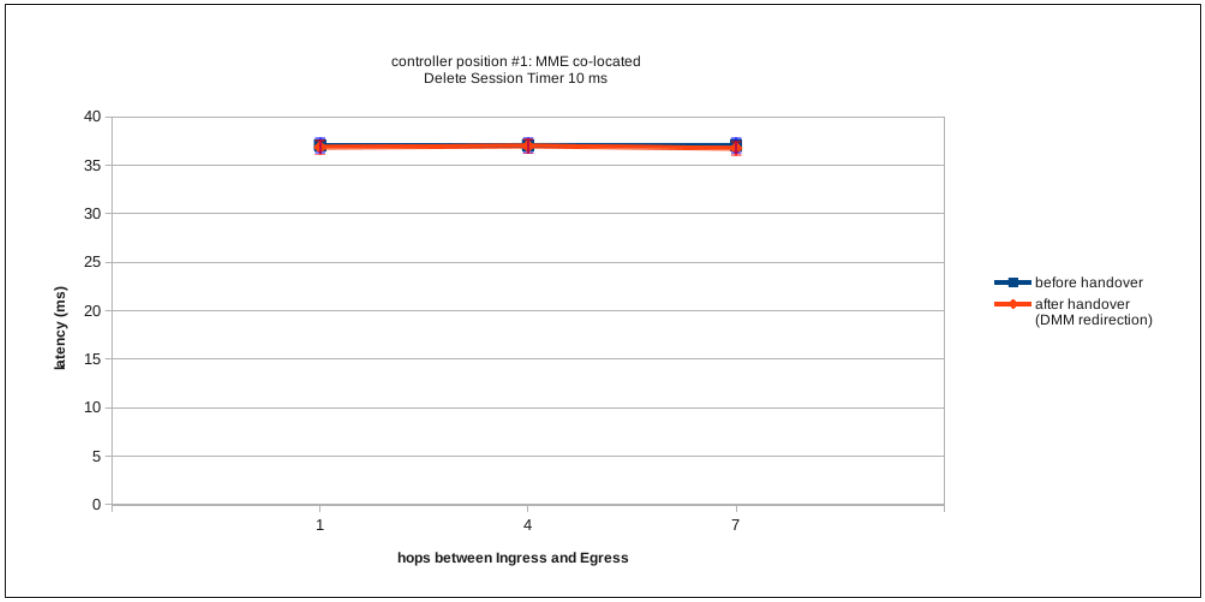


Figure 5.10: Average latency of downlink data packet delivery before and after handover. Partial OpenFlow used to redirect traffic after handover

As for the Double NAT case, the graph clearly demonstrates that DMM traffic redirection has no impact on the latency of downlink data packet delivery in the network topology used in the simulation experiments. Furthermore no variation were also observed in the collected results when the distance between Ingress and Egress OpenFlow switches varies.
The observed results were expected since the used operator's transport network topology do not offer the possibility to exploit sub-optimal routes when the UEs' downlink traffic is redirected.

Figure 5.11 shows the average latencies of downlink data packets delivered via the X2 path or via DMM traffic redirection to UEs. The results from different distances (number of hops) between Ingress and Egress OpenFlow switches are shown in the graph. The position of the OpenFlow Controller is also varied, since it may impact the load of X2 traffic. Results are shown for the case when the Delete Session Timer has been setup to 10 ms, in order to have the maximum throughput in the X2 data path.

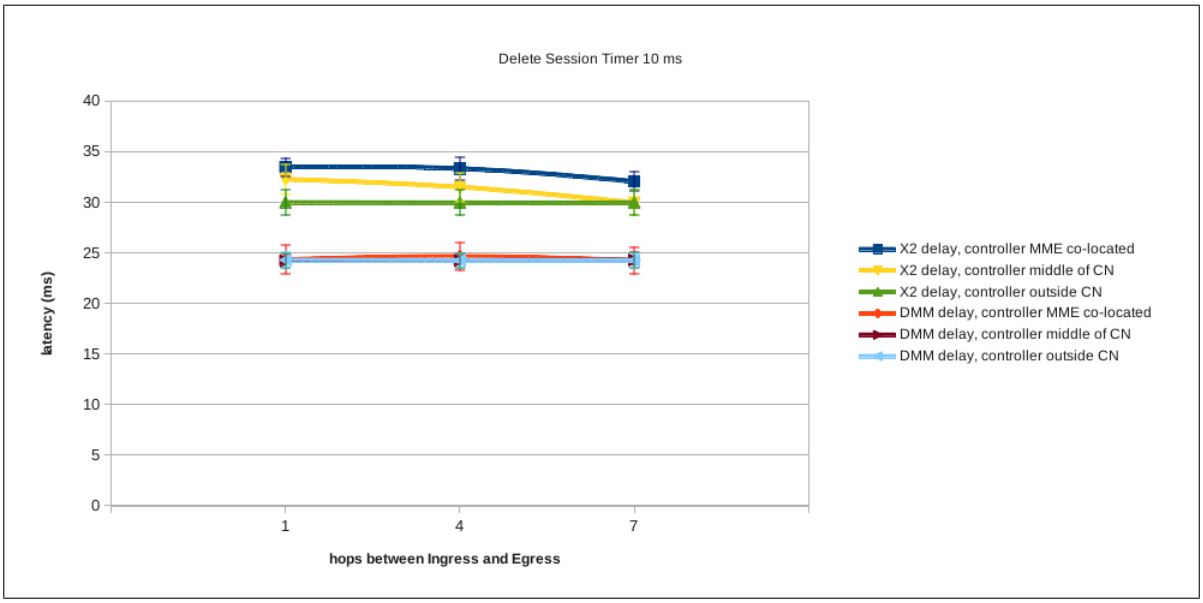


Figure 5.11: Average latency of downlink data packets delivered via X2 path and of the first packets redirected through Partial OpenFlow-based DMM transport network

The graph shows that independently from the distance between Ingress and Egress points and the positioning of the OpenFlow Controller, DMM traffic redirection outperforms X2 data forwarding. Furthermore as proposed for Double NAT, a function can be implemented in the network to predict the mobility of UEs in order to setup DMM traffic redirection prior to the trigger of the handover procedure. In this way X2 data forwarding can be avoided in the network and resources can be saved by operators in both RAN and CN. Moreover a buffering mechanism will be needed in the DMM transport network to store downlink data packets during the execution of the handover procedure.

5.1.2.2 CDF of latency of first DMM-redirected data packet

The CDF of latency of the first DMM-redirected data packet after the completion of the handover procedure when Partial OpenFlow is used as DMM solution is shown in Figure 5.12.

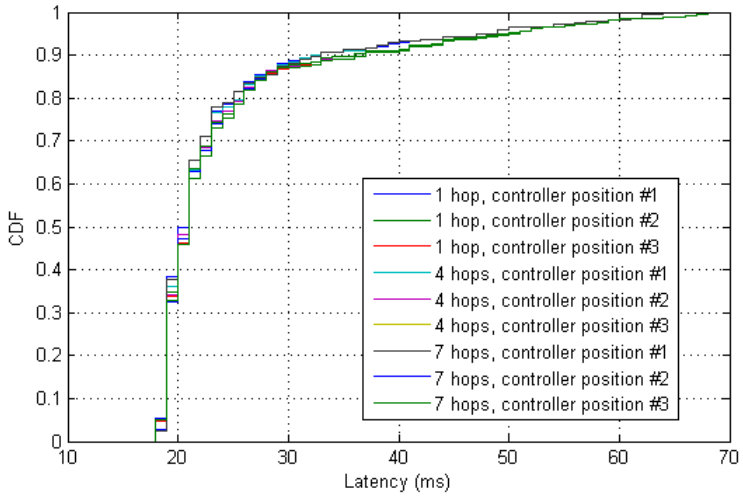


Figure 5.12: CDF of latency of first downlink data packets redirected through Partial OpenFlow-based DMM transport network

Being VoIP the studied UEs' traffic, 150 ms is accepted as a maximum one-way latency. The CDF in Figure 5.12 clearly shows that in all studied scenarios there is a 100% probability of having DMM-redirected data packets which do not overcome the maximum latency threshold of a VoIP session. Furthermore more than 95% of the observed packets has a latencies below the 50 ms backbone providers SLA maximum latency for VoIP traffic [80].

5.1.2.3 Throughputs

The impact of placing the Ingress and Egress OpenFlow switches at different distances (number of hops) with respect to the positions of the OpenFlow Controller is studied by analyzing the load and throughput of data traffic forwarded via the X2 path. In fact when DMM traffic redirection has not been setup yet upon completion of the UE radio handover, downlink data traffic will still be delivered to the source eNodeB which will then forward it to the target eNodeB via the X2 tunnel. In other words the load of the X2 forwarded traffic gives a good indication of how long is DMM traffic redirection setup with respect to the variation in positioning both Ingress OpenFlow switches and OpenFlow Controller. As explained in Section 4.2.1, the Egress OpenFlow switches have a fixed position as the next hop router on each S-/P-GW SGi interface in all simulated scenarios.
Data forwarding via the X2 path can occur during the *handover execution* and *handover completion* phase. Whether X2 data forwarding during the *handover execution* phase is not a choice for operators,

setting the Delete Session Timer to 0 ms gives them the possibility to dismantle the X2 tunnel as soon as the path switch procedure has been completed. In standard condition the Delete Session Timer is setup to 10 ms. In order to observe the impact that this choice can have on the UE downlink traffic, results from both scenarios have been collected.

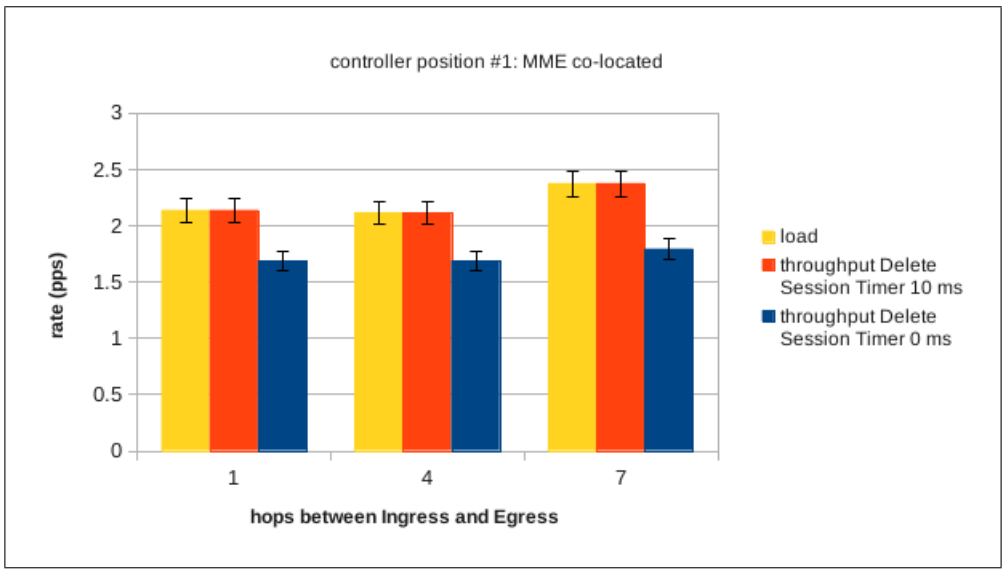


Figure 5.13: Load and throughput of X2 tunnel with OpenFlow Controller co-located with MME (Partial OpenFlow)

Differently from what was happening in the Double NAT case, Figure 5.13 shows that independently from the distance between Ingress and Egress OpenFlow switches the X2 tunnel is equally used when the OpenFlow Controller is co-located with the MME.

If the latency in setting up the OpenFlow tables of both Ingress and Egress OpenFlow switches increases, so does the probability of having X2 data forwarding after the completion of the path switch phase. This situation can be observed in Figure 5.13. Independently from the distance between Ingress and Egress OpenFlow switches no packet loss is present if the Delete Session Timer is setup to 10 ms, meaning that those 10 extra ms are enough to cope with the latency needed to setup DMM traffic redirection. Although only few, some packets are lost when the Delete Session Timer is setup to 0 indicating that the decision on whether or not use X2 data forwarding after the completion of the path switch procedure is dependent on the size of the network and the approximate distance from the Ingress and Egress OpenFlow switches to the OpenFlow Controller.

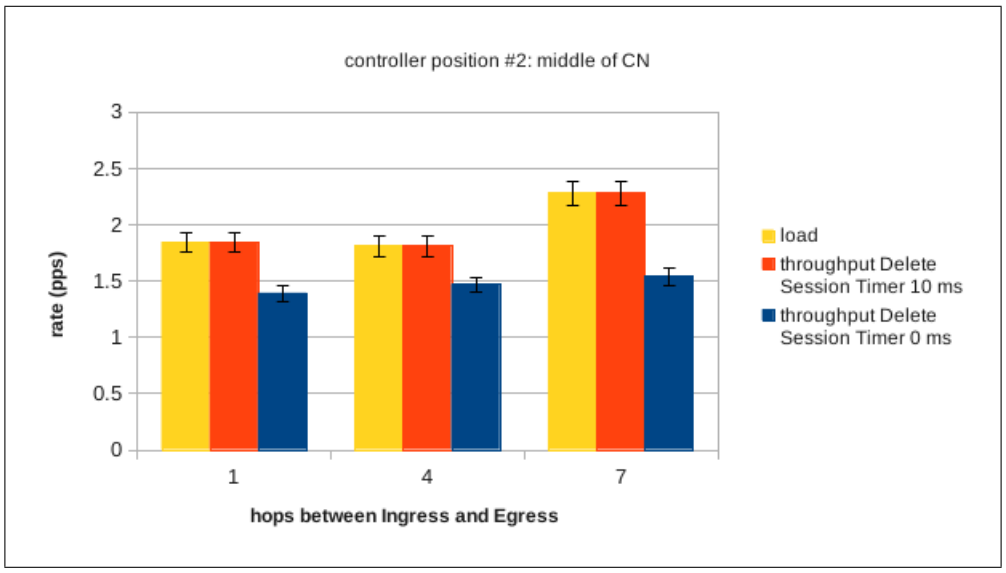


Figure 5.14: Load and throughput of X2 tunnel with OpenFlow Controller positioned in the middle of the core network (Partial OpenFlow)

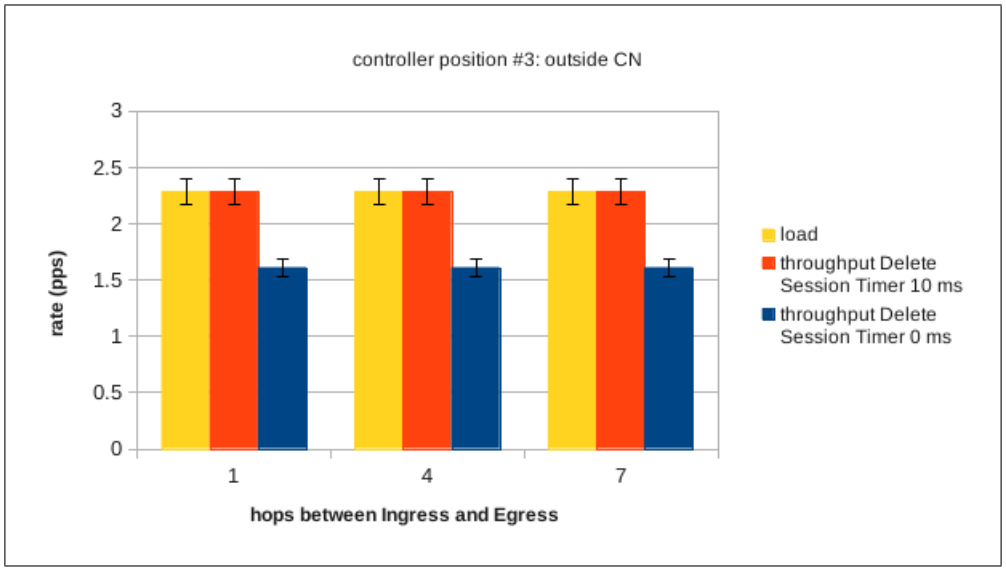


Figure 5.15: Load and throughput of X2 tunnel with OpenFlow Controller positioned outside of the operator's transport network (Partial OpenFlow)

A similar behavior has been observed for both the case when the OpenFlow Controller is located in the middle of the operator's transport network and when the OpenFlow Controller is located outside of the operator's transport network. The collected results are shown in Figure 5.14 and Figure 5.15 respectively. From the two graphs it can be observed that when the OpenFlow Controller is placed in the middle of the core network DMM traffic redirection is setup quicker and thus the load of the X2 forwarded traffic is slightly decreased. When the OpenFlow Controller is positioned outside of the

operator's transport network the same X2 traffic load has been observed independently from the distances between Ingress and Egress OpenFlow switches. This is due to the increased distance between the OpenFlow Controller and the Egress OpenFlow switches.

In both cases the 10 extra milliseconds have been proven to be enough to correctly deliver all the received downlink packets via the X2 tunnel once that the path switch procedure has been completed.

To study the impact that the lack of X2 data forwarding capability can have on the system in the studied scenarios, the total load and throughput of the system have been collected both for the case when X2 data forwarding is available and the UE session is kept active in the source S-/P-GW also during *handover execution* phase and for the case when X2 data forwarding is not used during handover. Figures 5.16, 5.17 and 5.18 show the results with different positioning for the OpenFlow Controller.
In all cases packets are lost when they are not forwarded via the X2 path with the gap between the throughput of the system with X2 data forwarding and the throughput of the system without X2 data forwarding increasing when the OpenFlow Controller is placed more further away from the EPC part of the network and reaching its maximum point in the scenario when the OpenFlow Controller is placed outside of the operator's transport network.

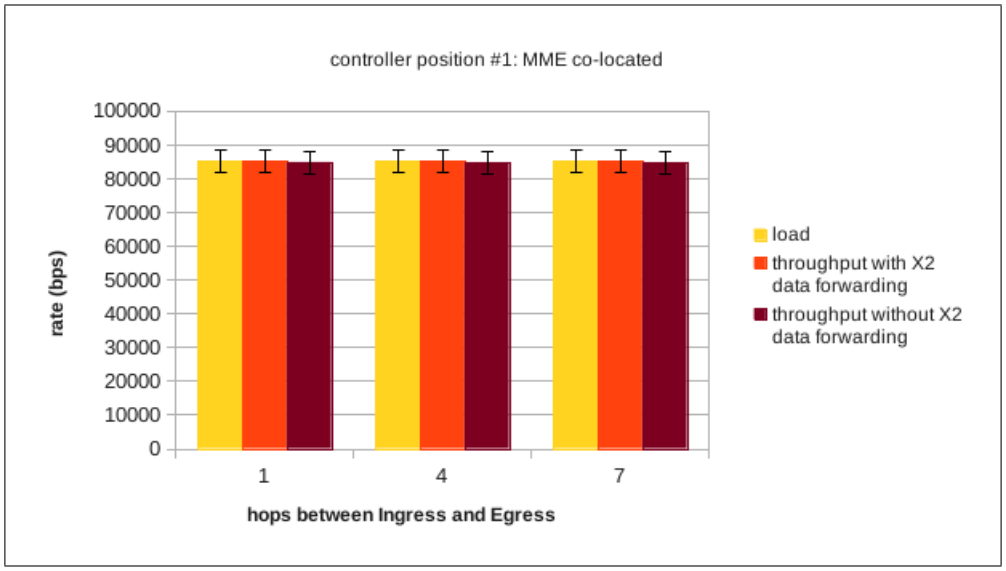


Figure 5.16: Load and total throughputs of the system with and without X2 data forwarding capabilities with OpenFlow Controller co-located with the MME (Partial OpenFlow)

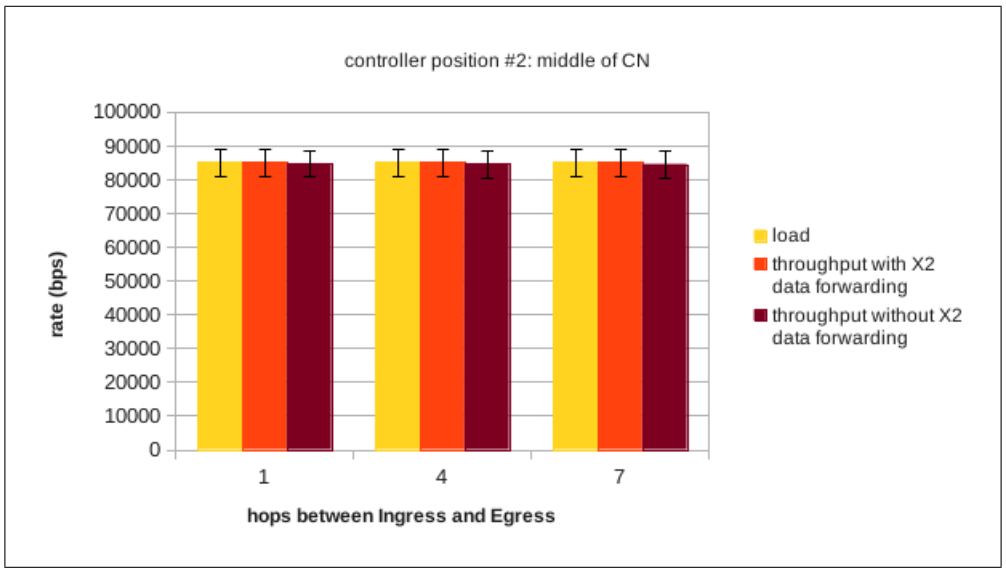


Figure 5.17: Load and total throughputs of the system with and without X2 data forwarding capabilities with OpenFlow Controller positioned in the middle of the core network (Partial OpenFlow)

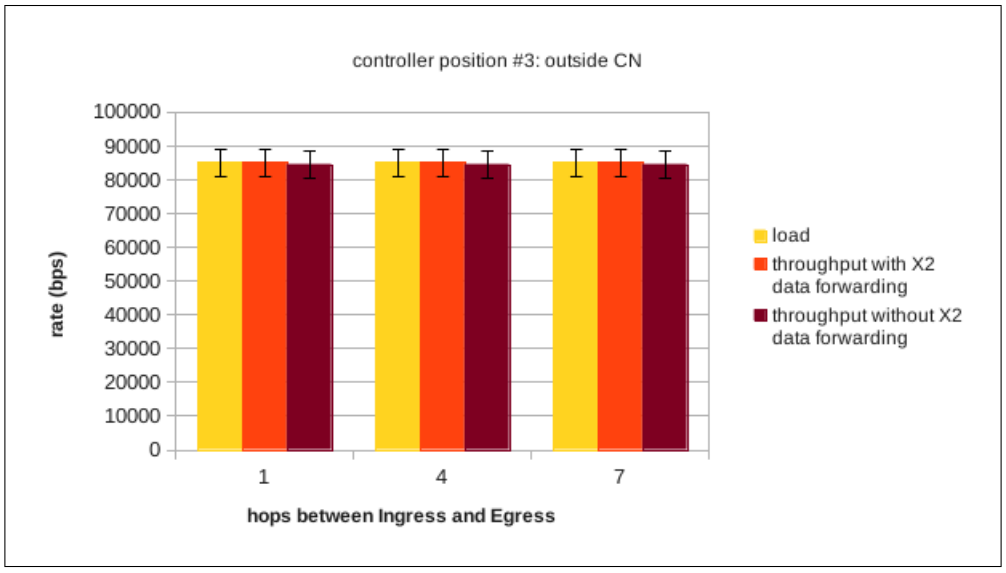


Figure 5.18: Load and total throughputs of the system with and without X2 data forwarding capabilities with OpenFlow Controller positioned outside of the operator's transport network (Partial OpenFlow)

As already mentioned, in order to offer seamless mobility to UE also in the case when no X2 data forwarding is available in the access network, a function to predict the mobility of UEs should be present in the network to setup DMM traffic redirection prior to the trigger of the handover procedure. Moreover since the UE session(s) have not been setup yet in the target eNodeB and target S-/P-GW until the conclusion of the *handover completion* phase, a buffering mechanism needs to be implemented in the DMM transport network or at the S-/P-GW level to store downlink data packets directed to the

moving UE. Once that the handover procedure has been completed, the buffer data traffic can be forwarded to the moving UE. In this way resources can be saved in the source and target eNodeB since buffering of downlink data traffic occurs in the transport network above the EPS. Indeed packets will not need to be reorder in the target eNodeB which is what happens in current standard procedure to cope with packets delivered via the X2 path. Overall this procedure can increase the total throughput of the system while decreasing the latency of downlink data packets and saving resources in the RAN.

5.1.2.4 Summary

Seamless mobility has been provided to all moving UEs in all the studied network topology deployments when Partial OpenFlow was the used DMM solution. The best results in terms of latency in configuring traffic redirection in the DMM transport network were obtained when the OpenFlow Controller has been positioned in the middle of the operator's core network and the distance between Ingress OpenFlow switches and Egress OpenFlow switches was lower than or equal to 4 hops. The position of the OpenFlow Controller had no impact for the case when the largest DMM transport network topology deployment was used in the simulations.

In all the studied network topologies and independently on the positioning of the OpenFlow Controller, setting the Delete Session Timer to 0 ms caused packet loss to be introduced in the network, compromising the seamlessness of the mobility procedure to UEs. X2 data forwarding is therefore required also for a period of time following the conclusion of the path switch request procedure. Extra 10 ms has been always proven to be enough to empty the source eNodeB's X2 buffer before that the setup of DMM traffic redirection was completed.

As expected, disabling X2 data forwarding from the network caused packet loss. DMM traffic redirection through a Partial OpenFlow transport network can be considered as an alternative to the current X2 forwarding scheme. A function to predict the mobility of UEs would need to be present in the network in order to setup DMM traffic redirection prior to the trigger of the handover procedure. In this way resources can be saved by operators in both RAN and CN. Downlink data packets will then need to be buffered in the transport network above the EPS during the execution of the EPS handover.

5.1.3 Full OpenFlow

Differently from the previous two solutions, in the Full OpenFlow case the distance between Ingress and Egress points of the DMM transport network is kept fixed. The worst case scenario with average 7 hops between Ingress and Egress points has been used in the experiment in order to have a higher density of *OpenFlow-full* switches. The used topology is depicted in Figure 4.8 of Section 4.2.1.

5.1.3.1 Average latencies of downlink data packets

Figure 5.19 shows the average latency of downlink data packet delivery before and after handover. When handover is completed, Full OpenFlow is used to redirect the traffic to the current UEs' mobility anchor point (target S-/P-GW). The results from different positions of the OpenFlow Controller in the network are shown in the graph. Since no impact is given to the results by the value of the Delete Session Timer, the shown results refer to the case when it has been set to 10 ms.

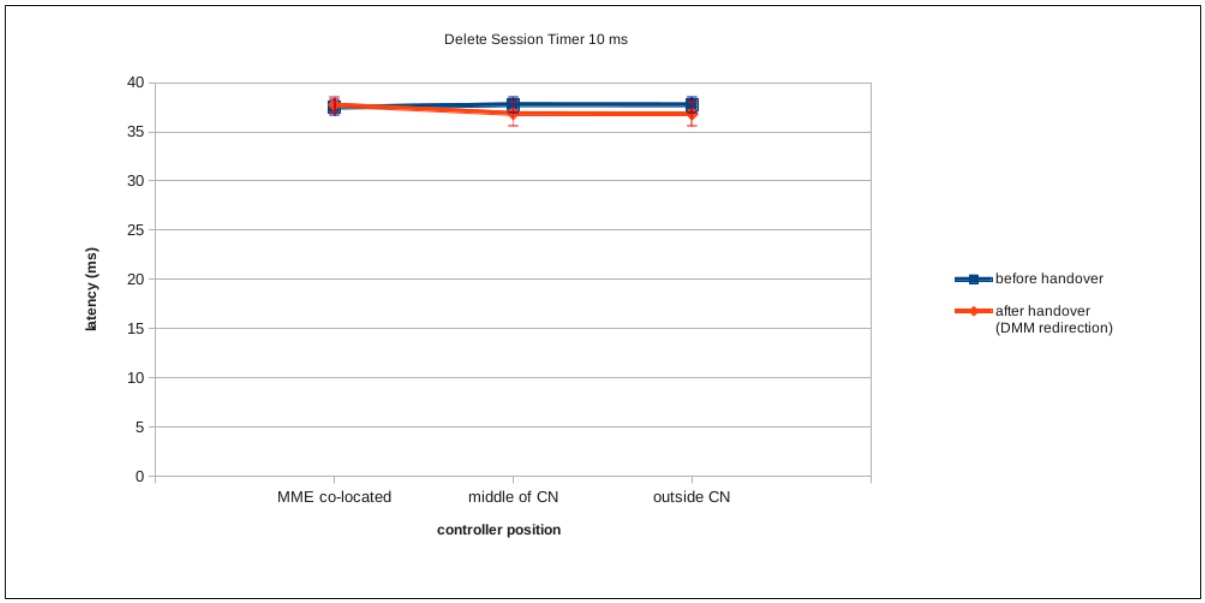


Figure 5.19: Average latency of downlink data packet delivery before and after handover. Full OpenFlow used to redirect traffic after handover

As for the Double NAT and Partial OpenFlow cases, the graph clearly demonstrates that DMM traffic redirection has no impact on the latency of downlink data packet delivery in the network topology used in the simulation experiments.
The observed results were expected since the used operator's transport network topology do not offer the possibility to exploit sub-optimal routes when the UEs' downlink traffic is redirected.

Figure 5.20 shows the average latencies of downlink data packets delivered via the X2 path or via DMM traffic redirection to UEs. The results from different positions of the OpenFlow Controller are shown in the graph. The position of the OpenFlow Controller can affect the X2 traffic load. Results are shown for the case when the Delete Session Timer has been setup to 10 ms, in order to have the maximum throughput in the X2 data path.

The graph shows that independently from positioning of the OpenFlow Controller, DMM traffic redirection outperforms X2 data forwarding. The X2 data forwarding capability can be removed from the network if a UEs mobility prediction system would be available and used to setup DMM traffic

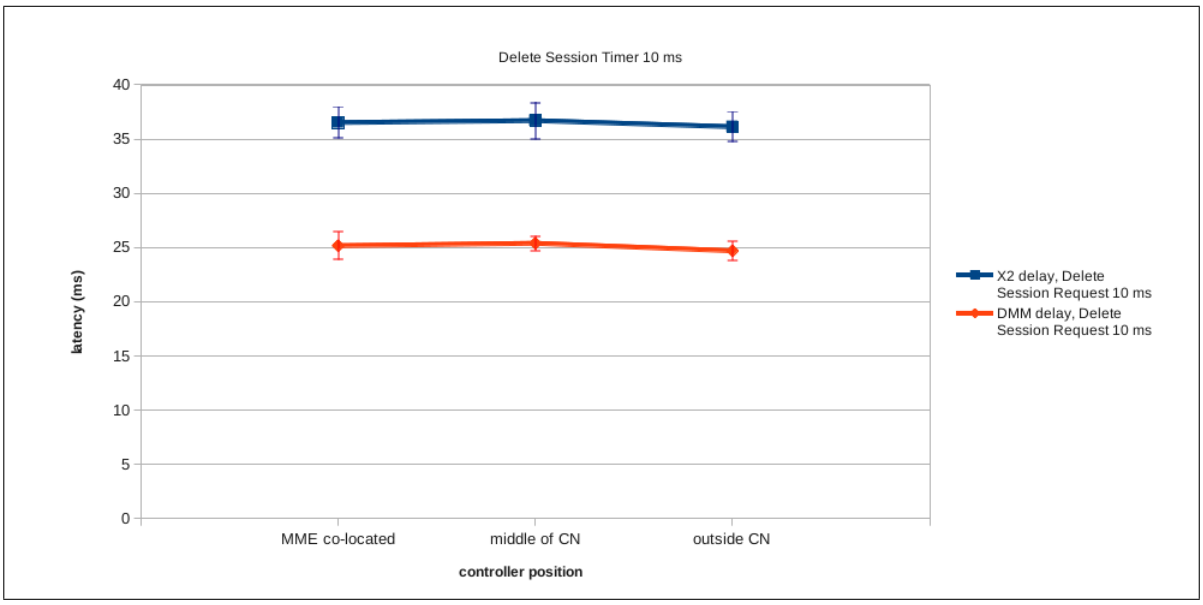


Figure 5.20: Average latency of downlink data packet delivered via X2 path and of the first packets redirected through Full OpenFlow-based DMM transport network

redirection prior to the trigger of the handover procedure. In this way operators can save resources in both RAN and CN. Moreover a buffering mechanism will be needed in the DMM transport network to store downlink data packets during the execution of the handover procedure.

5.1.3.2 CDF of latency of first DMM-redirected data packet

The CDF of latency of the first DMM-redirected data packet after the completion of the handover procedure when Full OpenFlow is used as DMM solution is shown in Figure 5.21.

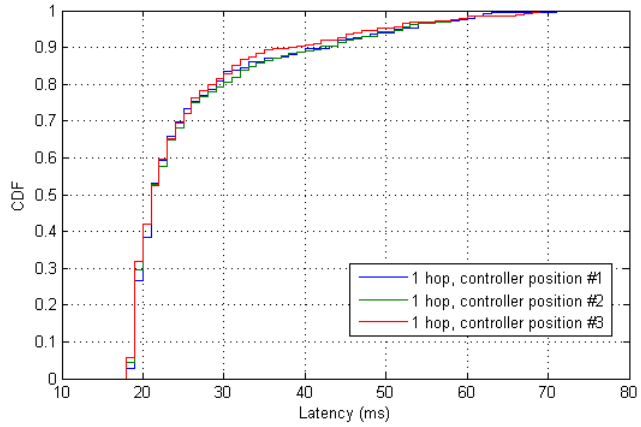


Figure 5.21: CDF of latency of first downlink data packets redirected through Full OpenFlow-based DMM transport network

The CDF in Figure 5.21 clearly shows that in all studied scenarios there is a 100% probability of having DMM-redirected data packet which do not overcome the maximum one-way latency threshold of a VoIP session (150 ms). Furthermore more than 92% of the observed packets has a latencies below the 50 ms backbone providers SLA maximum latency for VoIP traffic [80].

5.1.3.3 Throughputs

The load and throughput of data traffic forwarded via the X2 path are studied to analyze the impact of placing the OpenFlow Controller at different positions inside or outside the operator's network. In fact when DMM traffic redirection has not been setup yet upon completion of the UE radio handover, downlink data traffic will still be delivered to the source eNodeB which will then forward it to the target eNodeB via the X2 tunnel. In other words the load of the X2 forwarded traffic gives a good indication of how long is DMM traffic redirection setup with respect to the variation in positioning of the OpenFlow Controller.

Data forwarding via the X2 path can occur during the *handover execution* and *handover completion* phase. Whether X2 data forwarding during the *handover execution* phase is not a choice for operators, setting the Delete Session Timer to 0 ms gives them the possibility to dismantle the X2 tunnel as soon as the path switch procedure has been completed. In standard condition the Delete Session Timer is setup to 10 ms. In order to observe the impact that this choice can have on the UE downlink traffic, results from both scenarios have been collected.

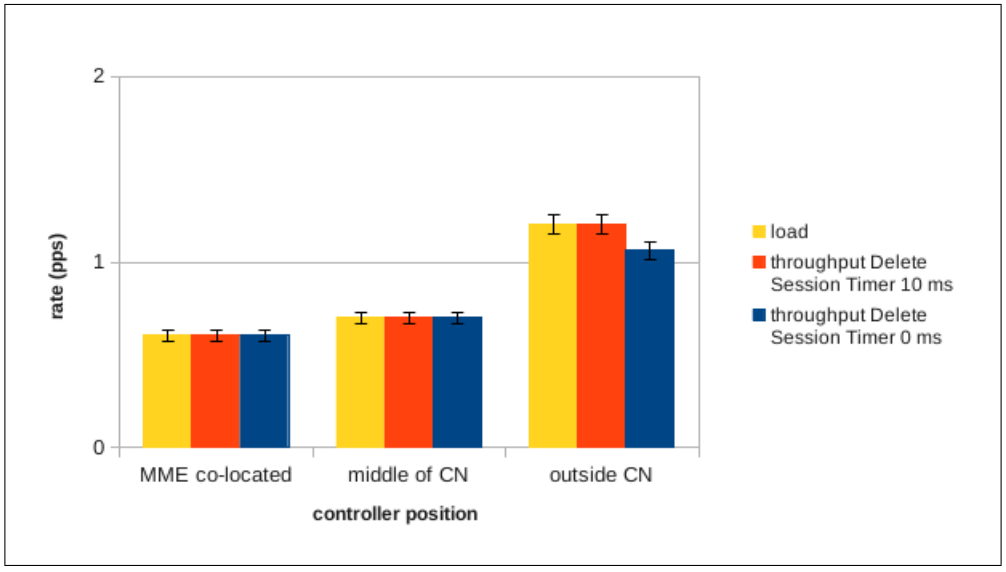


Figure 5.22: Load and throughput of X2 tunnel with different OpenFlow Controller positioning (Full OpenFlow)

Figure 5.22 shows the load and throughputs of the X2 forwarded traffic. The Full OpenFlow transport network is setup more quickly when the OpenFlow Controller is placed inside the operator's transport

network. The load of the X2 tunnel is lower compared to what has been observed in all scenarios for both Double NAT and Partial OpenFlow solutions. These results demonstrate the efficiency of the Full OpenFlow solution in the studied network topologies. A comparisons between the three proposed DMM solutions will be carried out in the next subsection.

When the OpenFlow Controller is placed outside the operator's core network, the observed X2 traffic load doubles respect to the case when the OpenFlow Controller is co-located with the MME. Furthermore some packet loss is experienced when X2 data forwarding is not used after the completion of the path switch request, hinting that DMM traffic redirection has not been fully setup yet when the UE has been completely handed over to the target eNodeB.

When the OpenFlow Controller is placed inside the operator's transport network the X2 buffer is emptied prior to the completion of the path switch procedure demonstrating that the time required to setup DMM traffic redirection is less than the time required to complete the X2-based handover procedure.

To study the impact that the lack of X2 data forwarding capability can have on the system in the studied scenarios, the total load and throughput of the system have been collected both for the case when X2 data forwarding is available and the UE session is kept active in the source S-/P-GW also during *handover execution* phase and for the case when X2 data forwarding is not used during handover. Figures 5.23 shows the results with different positioning for the OpenFlow Controller.

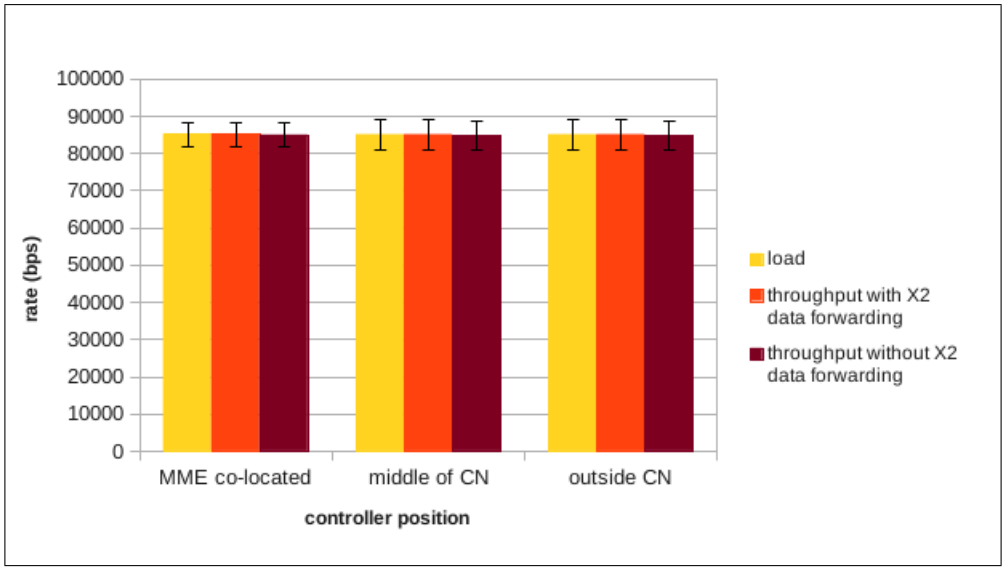


Figure 5.23: Load and throughputs of the system with and without X2 data forwarding capabilities with different OpenFlow Controller positioning (Full OpenFlow)

In all cases packets are lost when they are not forwarded via the X2 path with the gap between the throughput of the systems with X2 data forwarding and without X2 data forwarding reaching its maximum when the OpenFlow Controller is placed outside of the operator's transport network.

5.1.3.4 Summary

Full OpenFlow provided seamless mobility to all moving UEs in the studied network topology deployment. The best results in terms of latency in configuring traffic redirection in the DMM transport network were obtained when the OpenFlow Controller has been positioned inside the operator's core network.

Furthermore with the OpenFlow Controller in its optimal positions X2 data forwarding can be avoided after the completion of the path switch request procedure. Only for the case when the OpenFlow Controller was placed outside the operator's CN, 10 extra ms were required to empty the source eNodeB's X2 buffer prior to the completion of DMM traffic redirection setup.

As already proven in the Double NAT and Partial OpenFlow experiments, disabling X2 data forwarding from the network caused the introduction of some packet loss. A Full OpenFlow-based DMM redirection scheme can be considered as an alternative to the current X2 forwarding mechanism. A function to predict the mobility of UEs would need to be implemented in the network in order to setup DMM traffic redirection prior to the trigger of the handover procedure. Moreover a buffering mechanism will be needed in the DMM transport network to store downlink data packets during the execution of the EPS handover.

At first glance Full OpenFlow provided the best results in terms of efficiency and latency in establishing DMM traffic redirection compared to Double NAT and Partial OpenFlow solutions. To confirm or refute this impression a more careful comparison is carried out in the next subsection.

5.1.4 Comparison

In this subsection the results collected for each DMM solutions will be compared. Since in the Full OpenFlow experiments a single network topology has been implemented, the Double NAT and Partial OpenFlow results from this specific scenario will be used. The studied network topology is depicted in Figure 4.8 of Section 4.2 and it refers to the case when the Ingress DMM points are located close to the operator's Internet PoPs at an average distance of 7 hops from the Egress DMM points. This topology has been selected for two reasons: 1) increase the density of the OpenFlow switches present in the network for the Full OpenFlow network and 2) it has been proven to be the worst case scenario in both Double NAT and Partial OpenFlow solutions.

The position of the DMM Controller in the operator's network has been again varied following the three placements specified in Section 4.4.1.2.

In the previous subsections it has been proven that all three DMM solutions outperformed X2 data forwarding in terms of latency of downlink packet delivery demonstrating that, if implemented prior to the trigger of the handover procedure, DMM traffic redirection can provide lower latency and save resources to operators in both RAN and core network.

In order to better understand which DMM solutions provide lower latencies in the studied network topology the average latencies of downlink data packets delivered via DMM traffic redirection is shown in Figure 5.24. The Delete Session Timer has been setup to 10 ms.

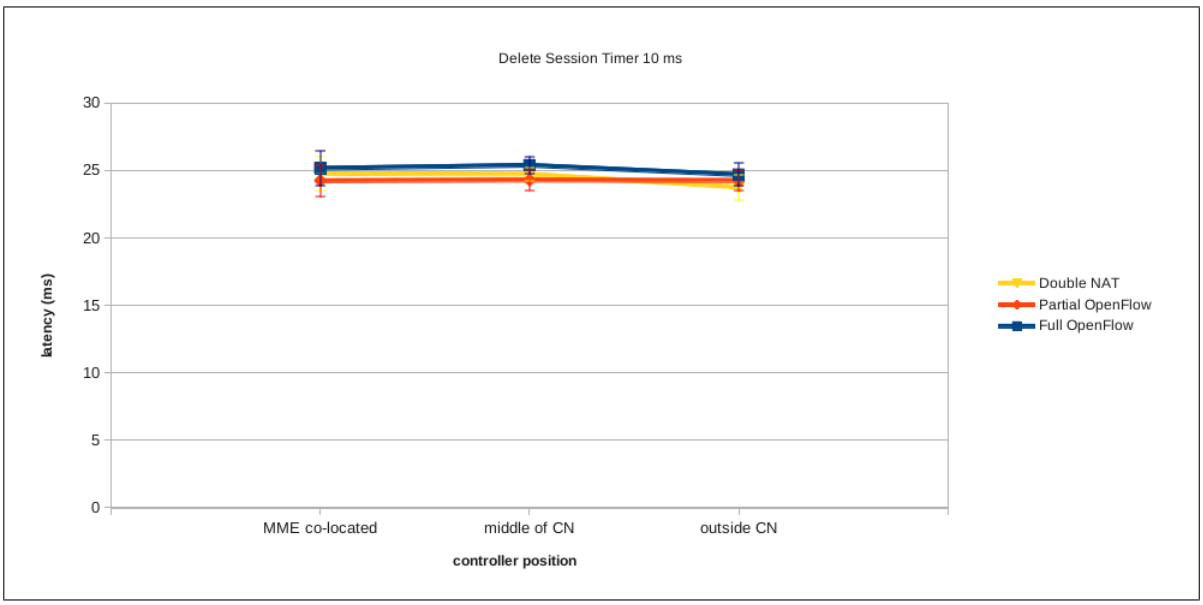


Figure 5.24: Comparison of average latency of downlink data packets delivered via DMM traffic redirection when Double NAT, Partial OpenFlow or Full OpenFlow is used as DMM solution

The graph shows that no significant difference has been observed in the average latency of the first DMM-redirected downlink data packets for each of the proposed solution. As expected, no impact is given to the studied latencies by the position of the DMM Controller in the operator's transport network.

As already done in the analysis of the single DMM solutions performances, the throughput of the X2 forwarded traffic is studied to evaluate the latency of DMM traffic redirection setup procedure with respect to the variation in positioning of the DMM Controller.

In order to observe the impact that the lack of X2 data forwarding after the completion of the path switch request procedure can have on the UE downlink traffic, results have been collected using two different values of the Delete Session Timer: 10 ms and 0 ms.

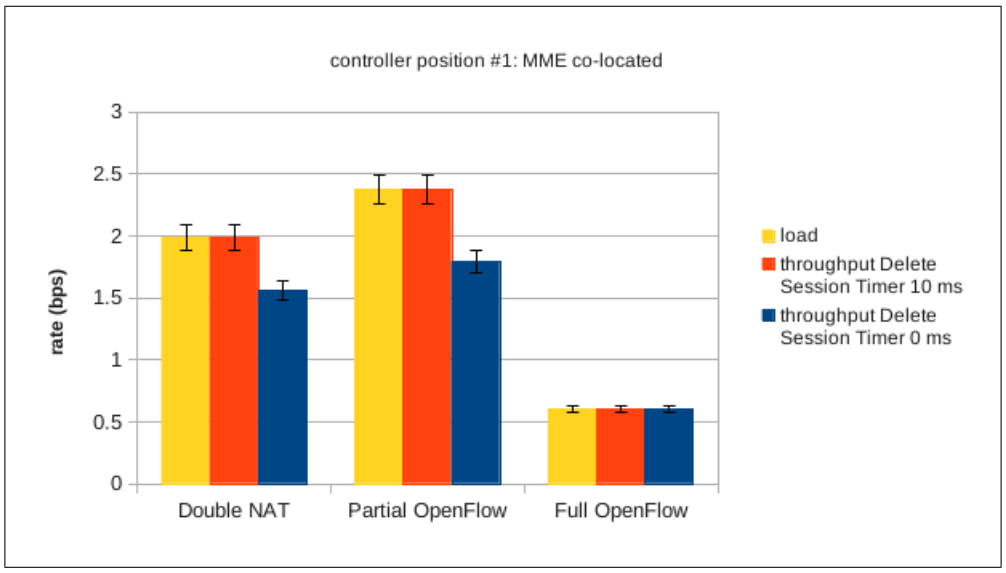


Figure 5.25: Comparison of throughput of X2 tunnel with DMM Controller co-located with MME when Double NAT, Partial OpenFlow or Full OpenFlow is used as DMM solution

Figure 5.25 shows that when the DMM Controller is co-located with the MME entity, the X2 traffic load when Full OpenFlow is the used DMM solution is much lower than for the Double NAT and Partial OpenFlow cases. Furthermore only in the Full OpenFlow case DMM traffic redirection is setup prior to the completion of the path switch request procedure since, differently from the other two solutions, no packet loss were present when the Delete Session Timer has been setup to 0 ms.

Slightly better results can be observed for the Double NAT solution compared to Partial OpenFlow in this scenario.

Similar results has been observed for the case when the DMM Controller is located in the middle of the operator's core network as shown in Figure 5.26. Also in this scenario DMM traffic redirection has been proven to be completed more quickly when Full OpenFlow is used in the transport network while extra 10 ms were needed in both Double NAT and Partial OpenFlow cases.

In this scenario, slightly better results can be observed for Partial OpenFlow compared to Double NAT, showing once more the similarity between the two solutions.

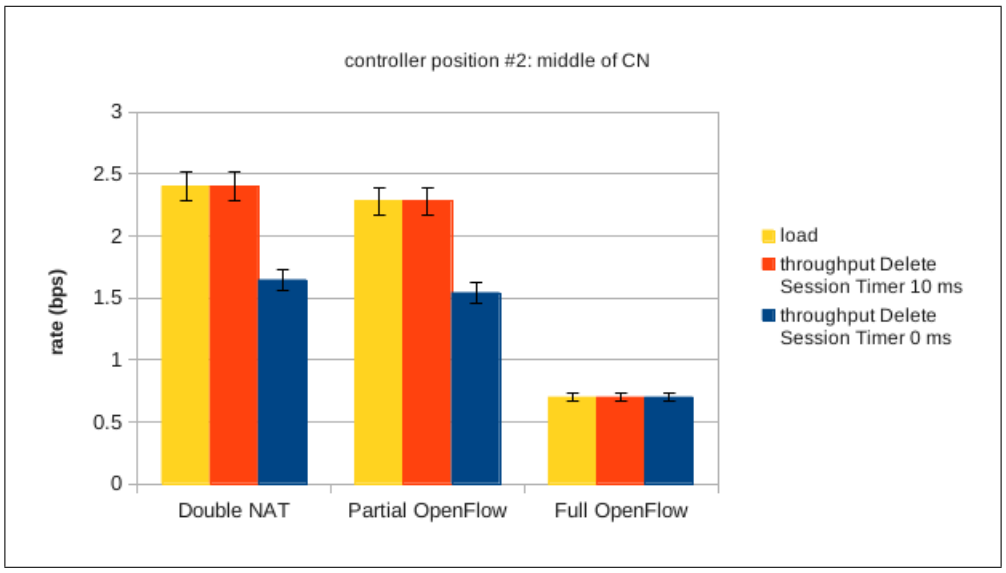


Figure 5.26: Comparison of X2 tunnel throughput with DMM Controller positioned in the middle of the core network when Double NAT, Partial OpenFlow or Full OpenFlow is used as DMM solution

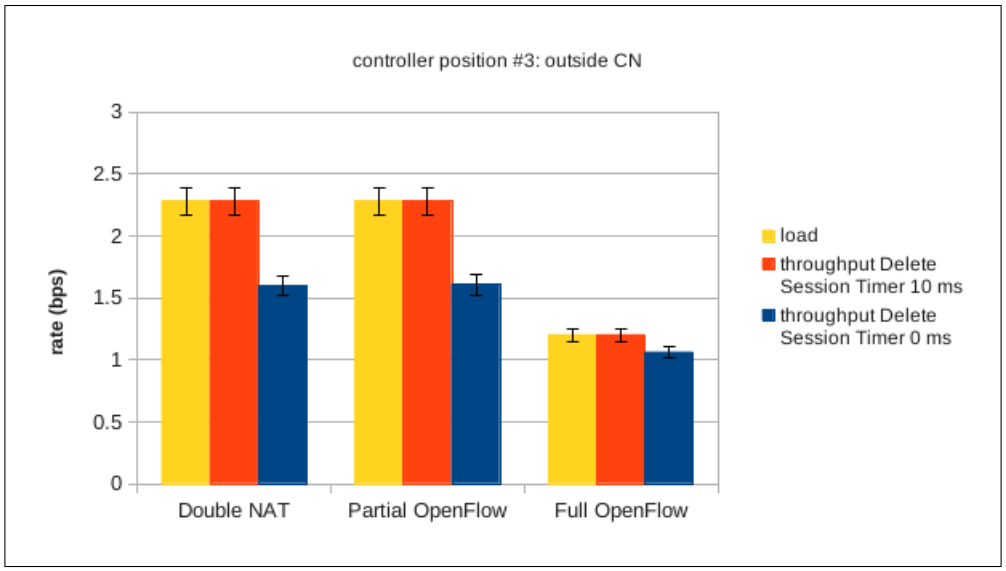


Figure 5.27: Comparison of X2 tunnel throughput with DMM Controller positioned outside of the operator's core network when Double NAT, Partial OpenFlow or Full OpenFlow is used as DMM solution

Figure 5.27 shows the results when the DMM Controller is positioned outside of the operator's CN. Once more Full OpenFlow stands up as the best solution in terms of latency in configuring DMM traffic redirection although 10 extra ms were in this case due to the higher distance between the OpenFlow Controller and the Egress part of the DMM transport network.

Double NAT and Partial OpenFlow showed similar results in this scenario.

The results observed for the X2 throughput can be seen as a surprise. Theoretically in the Full OpenFlow case, more OpenFlow switches will need to be setup to implement traffic redirection towards the UE's current mobility anchor point while in Double NAT and Partial OpenFlow only the Ingress and Egress DMM points need to be signaled by the DMM Controller. But being the Ingress and Egress DMM points positioned at the edges of the operator's transport network, the latency of the signaling messages directed to them will be equal for all three DMM solutions. Furthermore in the Full OpenFlow case when the OpenFlow switches positioned at the edge of the network receive the *Modify-State* message from the OpenFlow Controller, there is a high probability that the OpenFlow switches which lay in between them have already received the signaling and thus setup their OpenFlow tables.

Explained the above it appears clear that, due to the higher number of components which needs to be signaled, in the Full OpenFlow case it might happen that the DMM transport network is not in a consistent state until the setup of DMM traffic redirection has been completed. As a confirmation the average latency of downlink data packets delivered via X2 path has been analyzed for each solution and the results are shown in Figure 5.28.

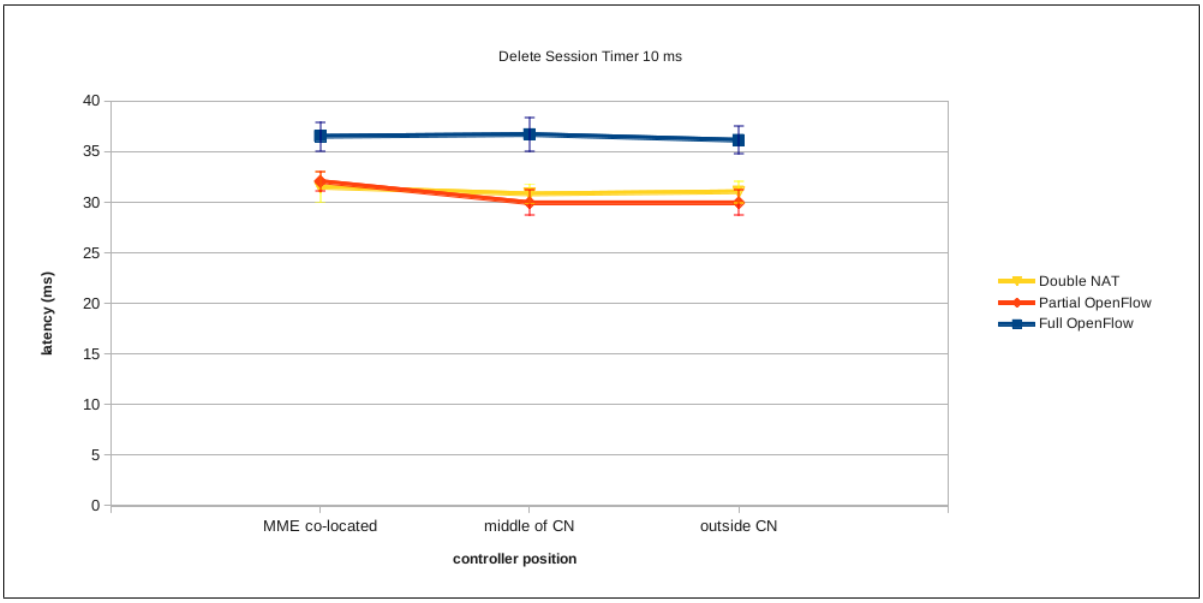


Figure 5.28: Average latency of downlink data packets delivered via X2 path when Double NAT, Partial OpenFlow or Full OpenFlow is used as DMM solution

The average latency of downlink data packets forwarded via the X2 path is significantly higher when Full OpenFlow is deployed in the network. Since the setup of the DMM transport network to implement the correct traffic redirection occurs in parallel with the forwarding of downlink data packets from the remote hosts to the EPS, in the case of Full OpenFlow is possible that parts of the network have been already setup to forward the specific flows' traffic to the target S-/P-GW while the OpenFlow tables of switches belonging to other parts of the network have yet to be updated. In other words

it exists a window of time in which the states of the OpenFlow switches in the network are inconsistent. Obviously the described situation leads to a case of sub-optimal routing and the results are well represented in 5.28.

Partial OpenFlow and Double NAT have shown similar results also with respect to this performance metric.

The impacts of not having X2 data forwarding capability in the network can be seen in Figure 5.29 where the downlink packet loss ratio of the system is depicted for all three DMM solutions.

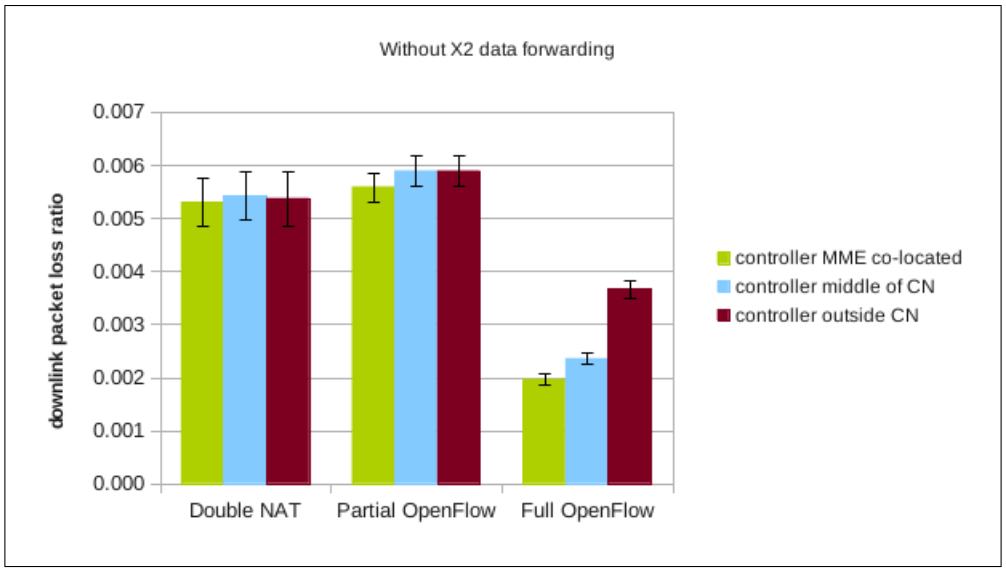


Figure 5.29: Comparison of downlink packet loss ratio of the system when Double NAT, Partial OpenFlow or Full OpenFlow is deployed as DMM solution and no X2 data forwarding capability is present in the network

Following what has been observed so far, when no X2 data forwarding capability is present in the network a higher and similar downlink packet loss ratio is observed for the case when Double NAT and Partial OpenFlow are the deployed DMM solutions while Full OpenFlow stands out as the more reliable solution. These results are directly related to what seen so far in this Section where the Full OpenFlow network has been proven to be the best solution in terms of efficiency in implementing DMM traffic redirection in the studied network topology.

If compared with what depicted in Figure 5.28, the results shown in Figure 5.29 demonstrate the existence of a trade-off between the average latency in the delivery of downlink data packets via X2 tunnel and the downlink packet loss ratio in case the X2 tunnel is not implemented at all. Moreover the results depicted in Figure 5.29 make even more clear the necessity of a mobility prediction function in the network as a solution to the lack of X2 data forwarding capability.

For all three DMM solutions the minimum packet loss is observed when the DMM Controller is co-located with the MME entity due to the low latency of signaling between the EPC and the DMM-plane.

To conclude the comparison, Figure 5.30 shows the level of the signaling load of the system when each of the proposed DMM solutions is deployed in the network.

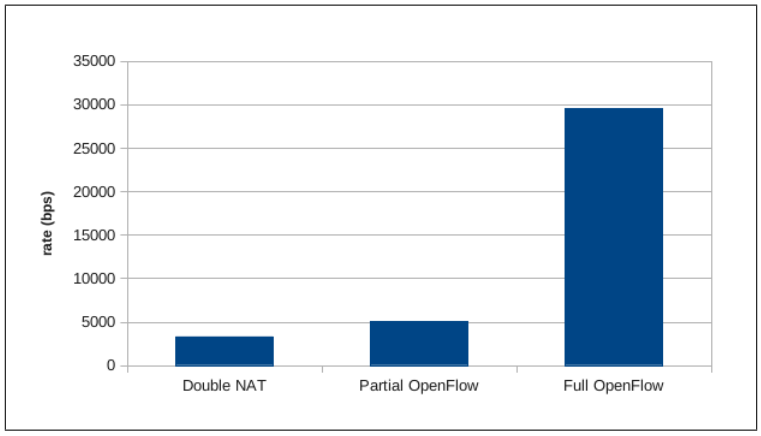


Figure 5.30: Comparison of signaling load of the system when Double NAT, Partial OpenFlow or Full OpenFlow is deployed as DMM solution

The results in the graph refer only to the load of the DMM-related signaling traffic required to setup DMM traffic redirections for 30 moving UEs while the signaling traffic exchanges between EPS entities are not taken into consideration.

Unsurprisingly the signaling load required by the Full OpenFlow solution is significantly higher compared to the one needed by Double NAT and Partial OpenFlow solutions. This is clearly given to the fact that a higher number of network entities need to be signaled by the OpenFlow Controller when Full OpenFlow is the deployed DMM solution.

Although with a small margin, Double NAT requires a lower load of signaling traffic to setup the same amount of DMM traffic redirections compared to the Partial OpenFlow solution. The reason for this result is the smaller size of the *Rule Update* messages compared to the size of the *Modify-State* messages used by OpenFlow.

5.1.4.1 Summary

In the studied network topology, Full OpenFlow stood out as the more efficient solution in terms of latency in configuring traffic redirection in the DMM transport network. The comparison showed a strong resemblance in results when Double NAT and Partial OpenFlow is the deployed DMM solution in the network. This come as a confirmation on the similarity between these two solutions.

Due to the higher number of components which needs to be signaled, in the Full OpenFlow case it might happen that the DMM transport network is not in a consistent state until the setup of DMM traffic redirection has been completed leading to a case of sub-optimal routing for X2 forwarded

downlink data traffic. Anyway the observed latencies are well within the one-way maximum threshold for a VoIP session confirming that seamless mobility management is still provided to moving UEs.

5.2 Handover with virtual S-/P-GW migration

This set of experiments aims to demonstrate that mobility prediction is needed in the network to provide seamless mobility to UEs which are handed over to a target cell requiring the migration of the serving S-/P-GW virtualized entity. In particular the case when UEs are forced to be handed over to a target cell which momentarily does not have any functional serving S-/P-GW has been studied.

In this scenario a VM implementing S-/P-GW functionalities needs to be migrated to a data center in a location closer to the edge of the network. The migrated VM is not transferred via the operator's transport network since the migration is assumed to happen on a separated platform. For this reason the signaling and operations needed to migrate a VM are not discussed in this report and have been implemented in the simulations only as an additional delay to the handover procedure, named as Virtualization Platform delay (or VP delay).

Since the target eNodeB does not have any serving S-/P-GW upon handover termination, UEs' uplink and downlink traffic will be buffered in the target data center and locally delivered to the target S-/P-GW virtualized entity once that the migration procedure is completed.

The experiments are performed using the network topology depicted in Figure 4.8 of Section 4.2.1. The DMM Controller is positioned in what has been referred to as *MME co-located* position (or position #1) in Section 4.4.1.2 since this position gives a lower latency in the signaling between MME and the DMM Controller. The Delete Session Timer has been setup to 10 ms.

The readers are advised to refer to Section 4.4.3 for a complete definition of this set of experiments.

To verify the impact of VM migration on the seamlessness of UE mobility which requires S-/P-GW relocation, the average latency of the first downlink data packet received from the target eNodeB is evaluated. The proposed DMM solutions are used to redirected the traffic to the target S-/P-GW and the results for each solutions when the migrated VM have a size of 128 MB and 256 VM have been collected and are shown in Figure 5.31.

The graph clearly shows the impact that the VM migration has on the latency of the first packet received after the completion of the migration procedure. Whereas throughout Section 5.1 the observed downlink data packet latencies were always in the order of few tens of milliseconds, Figure 5.31 shows results in the order of seconds. The measured average latencies are only few milliseconds different from the Virtualization Platform (VP) delays simulated in the experiments. For this reason no substantial differences can be seen in the graph when each of the three proposed DMM solutions is deployed in the network.

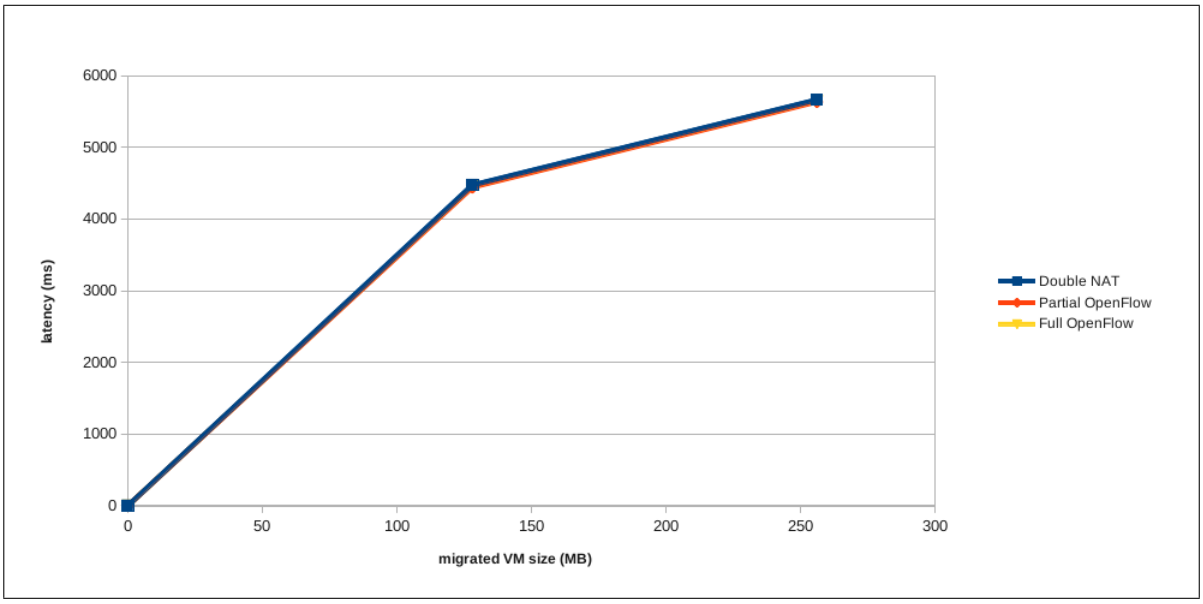


Figure 5.31: Average latency of first downlink data packet received after the completion of the VM migration when Double NAT, Partial OpenFlow or Full OpenFlow is the deployed DMM solution in the network

Obviously none of the observed latencies were below the standard one-way delay of a VoIP session making, as expected, the implemented mobility scenario non seamless.

Unfortunately the huge simulation time, required to simulate the cases when the migrated VM has a size bigger than 256 MB, made impossible to collect the results of this simulation prior to the writing of this report. Anyway in the case when VM of a size of 128 MB and 256 MB are migrated, the measured latencies of the first downlink data packet received through DMM redirection are only few milliseconds different from the simulated VP delays. For this reason in Figure 5.32, the VP delays summarized in Table 4.6 of Section 4.2.5 are also plot in order to show the expected latencies for migrated VM with a size bigger than 256 MB.

The results shown in this Section clearly demonstrated that a mobility prediction function is indeed required in the network to cope with the high latencies introduced by the migration of a VM entity or function which tasks cannot be temporarily replaced. Moreover the anticipation required by this function to predict the necessity to migrate a VM increases proportionally with the size of the VM which has to be migrated and can reach 1 minute with VM bigger than 2 GB.

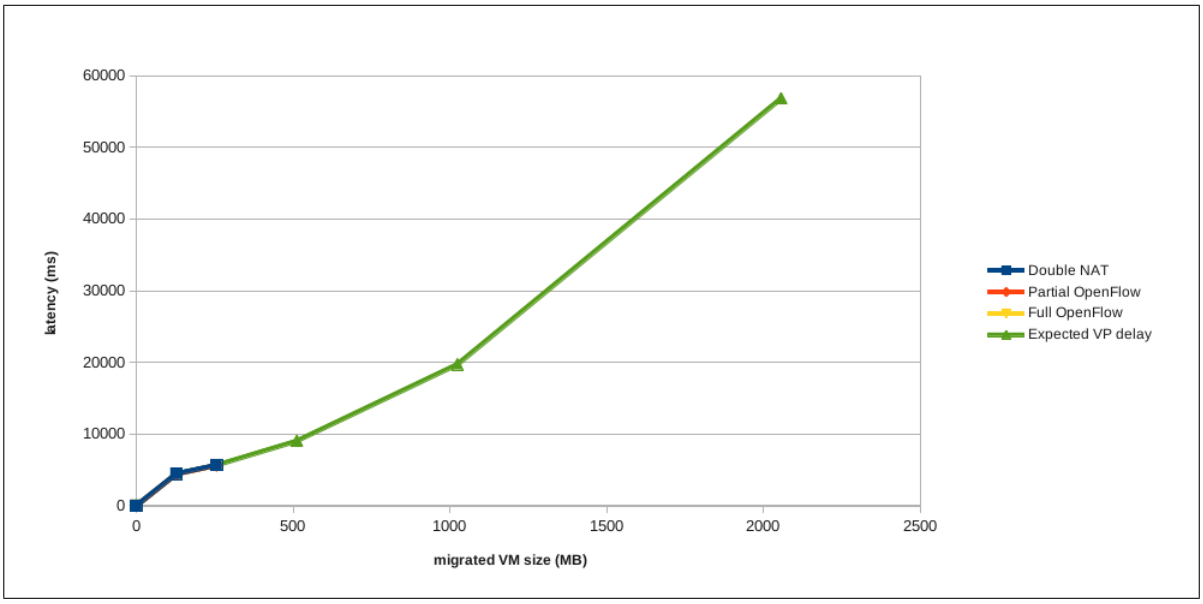


Figure 5.32: Average latency of first downlink data packet received after the completion of the VM migration when Double NAT, Partial OpenFlow or Full OpenFlow is the deployed DMM solution in the network

5.3 Chapter summary

In this Chapter, the evaluation of the proposed DMM solutions has been carried out. Some different sets of experiment conducted in NS3 LENA simulation environment to evaluate the performance of the proposed solutions have been discussed. Double NAT, Partial OpenFlow and Full OpenFlow have been proven to introduce a similar latency in redirecting downlink data packets to the relocated UE's mobility anchor point. The 100% of the measured latencies of DMM-redirected downlink VoIP data packets is lower than the standard maximum delay threshold for a VoIP session for each of the three deployed DMM solutions. Furthermore, for all three solutions the latency in redirecting downlink traffic to the target eNodeB via the target S-/P-GW is substantially lower than the latency of downlink data traffic forwarded to the target eNodeB via the X2 tunnel. For this reason DMM traffic redirection can be exploited as a countermeasure when X2 data forwarding is not available in the network given the fact that a mobility prediction function is deployed.

Indeed it has been demonstrated that Double NAT, Partial OpenFlow and Full OpenFlow offer seamless mobility management when used as the deployed DMM solution. When compared using the same network topology, Full OpenFlow clearly stood out as the more efficient solution due to its lowest overhead in configuring traffic redirection in the DMM transport network.

As expected, independently from the size of the DMM transport network and the distance between Ingress DMM points and Egress DMM points, the DMM Controller has been proven to take advantage from a positioning closer to the part of the operator's network where the virtualized EPS entities are

running. In fact a higher total throughput (and thus lower downlink packet loss ratio) and a lower use of the X2 tunnel has been observed in all scenarios where the DMM Controller has been co-located with the MME since signaling between these two entities was reduced to a case of local communication.

A more quickly setup of the DMM transport network results also in a higher probability of not using X2 data forwarding once that the path switch request procedure has been completed. In this way resources can be saved in the RAN since X2 tunnel can be dismantled as soon as the UE session(s) has been transferred to the target S-/P-GW. This behavior has been observed in all studied scenarios and independently on the used DMM solution.

Given the above, no substantial differences were encountered studying the results when Double NAT or Partial OpenFlow is the DMM solution deployed in the network, demonstrating the similarity of the two solutions. This together with the results observed for the Full OpenFlow solution, have proven once more the high efficiency and extensibility of the OpenFlow protocol.

The need of a mobility prediction function in the network when the virtualized S-/P-GW entity has to be migrated upon UE movement has been validated by means of proving the drastic effects that VM migration will have on the current standard handover procedures. Moreover the same mobility prediction scheme can be exploited to replace X2 data forwarding with DMM traffic redirection in the network. For this scope a buffering mechanism would need to be implemented to store downlink data packets directed to the moving UE in the DMM transport network. Resources can be saved in the source and target eNodeB since buffering of downlink data traffic will occur in the transport network above the EPS. Furthermore packets will not need to be reorder in the target eNodeB differently from what done in current standard procedure to cope with packets delivered via the X2 path. Overall this procedure can increase the total throughput of the system while decreasing the latency of downlink data packets and saving resources in both RAN and CN.

Chapter 6

Conclusions and future work

This Chapter discusses the conclusions and the future work. First the research questions are answered. Then the overall conclusions are given and discussed. Finally some recommendations for future work are presented.

6.1 Conclusions

Distributed Mobility Management (DMM) refers to mobility solutions designed to tackle the limitations of the currently deployed centralized mobility management approaches. DMM allows a cheaper, more distributed and efficient network deployment to operators by developing the concept of a flatter system, in which the mobility anchors are placed closer to the user, distributing the control and data infrastructures among the entities located at the edge of the access network.

On the other hand the use of Cloud Computing concept in LTE mobile networks could be a good solution to increase LTE's performance by building a shared distributed LTE mobile network that can optimize the utilization of resources, minimize communication delays, and avoid bottlenecks. Furthermore the use of Cloud Computing technologies and sharing the Cloud Computing infrastructure among different network operators will allow the development of novel DMM architectures.

In this report the requirements, architecture design and modifications required to the current standard mobility procedure to support DMM approaches have been presented. Three DMM solutions based on different technologies, protocols and/or network deployments have been described and their seamlessness has been assessed and verified for the case when UEs mobility causes their mobility anchor point to be relocated. A combination of literature study and simulation-based evaluation has been conducted to answer the main research question: "How can seamless DMM be implemented and evaluated in cloud based LTE system?"

Several sub-questions have been defined in Section 1.4, and the answers are given as the following:

1) Which requirements need to be satisfied by a DMM solution when applied in cloud based LTE systems?

Answer: a literature study has been conducted to answer the first sub-question. The requirements to be satisfied by a DMM solution when applied in cloud based LTE system are described in Chapter 2 of this report and are as follows: distributed network-based deployment, provision of transparent mobility support above the IP layer when needed, IPv6 as the primary deployment environment, consider the reuse and extension of existing mobility protocols, ability to co-exist with existing network deployments and end hosts, security and multicast awareness, allowing the split of data flows belonging to same users and support the separation between data plane and control plane.

2) Which architecture/framework can be used for the support of DMM in cloud based LTE systems?

Answer: literature study has also been used to address the second sub-question. A functional framework introducing a set of functional entities required to support IP address continuity in a distributed network deployment is described in Chapter 2. Two DMM architecture deployment variants using the previously introduced functional entities has also been proposed in Chapter 2. A third functional architecture has been designed and described in the same Chapter. This architecture is used in all the DMM solutions proposed in this report and it is based on traffic redirection above the EPS. To avoid encapsulation overhead, alternative forwarding techniques have been used instead of the popular IP tunneling to deliver the UE's downlink traffic to the currently used mobility anchor.

3) Which of the existing DMM solutions can be applied in cloud based LTE systems?

Answer: Chapter 3 introduced two main approaches to perform traffic redirection in the transport network above the EPS. Three solutions derived from these approaches have been selected and compared to give a preliminary indication of their possible impacts on current operators' networks.
The first solution utilizes Network Address Translation (NAT) at both ends of the operator's transport network, hence Double NAT, to solve the routing above the EPS.
Both second and third solutions are derived from the popular OpenFlow protocol and they utilize different features provided by OpenFlow to implement traffic redirection above the EPS. The first OpenFlow-based solution, implements NAT at both ends of the operator's transport network as done by the Double NAT solution. A special feature of OpenFlow switches named *Set-Field* action is used for this purpose. Only the switches placed at the edges of the operator's transport network will need to be *OpenFlow-hybrid* switches. This solution has been named Partial OpenFlow.
The second OpenFlow-based solution implement a DMM transport network entirely composed by *OpenFlow-full* switches. Differently from the other two solutions, no modifications are required to the downlink IP flows packet since flow-based routing is used in the network instead of Layer-3 routing. This solution is referred to as Full OpenFlow.

4) How can the DMM solution be implemented in cloud based LTE systems?

Answer: This sub-question is partially answered in Section 2.2 where the modifications required to support IP address continuity in the current standard 3GPP's mobility procedures are explained in details. The procedures and messages used in the interaction between the EPC MME entity and the DMM Controller (which is the entity entitled to manage the DMM transport network) compose the remaining part of this answer and they have been defined in Chapter 3. Since DMM traffic redirection occurs in the network above the EPS, a single signaling message (*Make Path* message) is required to be transmitted by the MME to the DMM Controller to correctly initiate the DMM transport network setup.

5) How can the seamlessness of the DMM solution be assessed and verified?

Answer: Chapter 4 motivates the choice of NS3 LENA simulation environment and discusses the implementation of the proposed DMM solutions. Furthermore some different sets of experiment conducted in NS3 LENA simulation environment to evaluate the performances of the proposed solutions are also described in Chapter 4. The simulation results and analysis for the conducted experiments are presented in Chapter 5.
All three DMM solutions have been proven to offer seamless Distributed Mobility Management, in all studied network topologies, to UEs being handed over between neighboring cells and whose movements require the relocation of their mobility anchor point. Although it introduces a higher signaling overhead in the network, Full OpenFlow stood out as the best DMM solution in terms of efficiency in configuring DMM traffic redirection, reliability and capability to cope with the absence of the X2 data forwarding feature in the network.
Furthermore DMM traffic redirection outperformed X2 data forwarding in terms of latency of downlink data delivery for each of the proposed DMM solutions. For this reason DMM traffic redirection can be considered as a valid alternative to X2 data forwarding if a function to predict the mobility of UEs would be present in the network.

Besides answering the aforementioned research questions, some extra conclusions have been drawn from the simulation experiments. The best positioning for the DMM Controller in the operator's transport network has been evaluated and a placement closer to the part of the network where the virtualized EPS entities are implemented has been proven to be the best choice due to the low latency of the signaling between the EPC MME and the DMM Controller.

The impact of VM migration on the proposed DMM solutions have been studied and it demonstrated that solutions like mobility prediction systems are indeed needed in the network to provide seamless mobility to UEs which are handed over to a target cell requiring the migration of the serving S-/P-GW virtualized entity.

6.2 Future work

For future work, more experiments should be done to verify the performance of the proposed DMM solutions in different network topologies. The deployed scenarios should target the possibility to exploit sub-optimal routes for traffic which is redirected via the DMM transport network. Furthermore a higher number of moving UEs should be used together with an increased number of simulation runs in order to refine better the obtained results.

For the Full OpenFlow network solution, experiments in smaller DMM transport network deployments should be carried out to confirm the results already obtained in the implemented worse-case network topology scenario.

A scenario where also the MME entity is relocated should be studied. In this case the proposed positioning of the DMM Controller can be questioned. Moreover the impact of utilizing multiple DMM Controllers within the same DMM transport network requires some further research since it may impact the latency in configuring DMM traffic redirection and the design of the signaling protocol between EPS and DMM-plane.

Last but not the least, a mobility prediction function has been proven to be required in the network to provide seamless mobility when the virtualized UE's mobility anchor point has to be migrated. Furthermore, the same function can be used for other purposes, for instance it can aim at the replacement of X2 data forwarding with one of DMM traffic redirection mechanisms described in this report.

The requirements, design and implementation of a buffer mechanism used to store uplink and downlink data packets within the DMM transport network need to be defined and evaluated. Buffering of downlink DMM-redirected data packets can occur also in the target data center where the virtualized S-/P-GW entity is to be migrated in the case when DMM traffic redirection is setup before the conclusion of the EPS handover procedure. The evaluation of this scenario is also left for further research.

Bibliography

[1] Bengt Ahlgren, Pedro A Aranda, Prosper Chemouil, Sara Oueslati, Luis M Correia, Holger Karl, Michael Sollner, and Annikki Welin. Content, connectivity, and cloud: ingredients for the network of the future. *Communications Magazine, IEEE*, 49(7):62–70, 2011.

[2] Christine de Monfreid. The LTE Network Architecture-A Comprehensive Tutorial. *Alcatel-Lucent White Paper. Some content may change prior to final publication*.

[3] 3GPP Technical Specification 36.300: Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 8), `www.3gpp.org`.

[4] MCN D2.2 "Overall Architecture Definition, Release 1, European Commission", deliverable 2.2, EU FP7 Mobile Cloud Networking public deliverable, November 2013.

[5] `lte.alcatel-lucent.com/locale/en-us/downloads/LTE-poster.pdf`.

[6] LENA M5 design documentation. Available online: `http://lena.cttc.es/manual/lte-design.html`.

[7] Ming Zhao and Renato J Figueiredo. Experimental study of virtual machine migration in support of reservation of cluster resources. In *Proceedings of the 2nd international workshop on Virtualization technology in distributed computing*, page 5. ACM, 2007.

[8] Worldwide Interoperability for Microwave Access (WiMAX), `www.wimax.com`.

[9] Long Term Evolution (LTE), `www.3gpp.org/LTE`.

[10] David Johnson, Charles Perkins, Jari Arkko, et al. Mobility support in IPv6, IETF RFC 3775, June 2004.

[11] Sri Gundavelli, Kuntal Chowdhury, Vijay Devarapalli, Basavaraj Patil, Kent Leung, et al. Proxy Mobile IPv6, IETF RFC 5213, June 2008.

[12] 3GPP Technical Specification 29.060, General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface (Release 8), `www.3gpp.org`.

[13] Philippe Bertin, Servane Bonjour, and J-M Bonnin. Distributed or centralized mobility? In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE, 2009.

[14] H Anthony Chan, Hidetoshi Yokota, Jiang Xie, Pierrick Seite, and Dapeng Liu. Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues. *Journal of Communications*, 6(1):4–15, 2011.

[15] László Bokor, Zoltán Faigl, and Sándor Imre. Flat architectures: Towards scalable future internet mobility. *The future internet*, pages 35–50, 2011.

[16] Mobile Cloud Networking (MCN) Project, `www.mobile-cloud-networking.eu`.

[17] Arjan J Staring. Applying the Cloud Computing Model in LTE based Cellular Systems. 2012.

[18] What is the EPS Bearer? `3gpp.wikispaces.com/What+is+the+EPS+Bearer`.

[19] Overview of 3GPP Release 8 V0.2.9 (2013-01), `www.3gpp.org/Release-8`.

[20] 3GPP Technical Specification 24.301: Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3 (Release 8), `www.3gpp.org`.

[21] Overview of 3GPP Release 9 V0.2.8 (2013-01), `www.3gpp.org/Release-9`.

[22] 3GPP Technical Specification 23.402: Architecture enhancements for non-3GPP accesses (Release 8), `www.3gpp.org`.

[23] C-RAN, `labs.chinamobile.com/cran/`.

[24] Venmani Daniel Philip, Yvon Gourhant, and Djamal Zeghlache. OpenFlow as an Architecture for e-Node B Virtualization. *e-Infrastructure and e-Services for Developing Countries*, pages 49–63, 2012.

[25] Cisco VN-Link: Virtualization-Aware Networking, white paper, `www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns892/ns894`.

[26] VRouter, `nrg.cs.ucl.ac.uk/vrouter`.

[27] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M Frans Kaashoek. The click modular router. *ACM Transactions on Computer Systems (TOCS)*, 18(3):263–297, 2000.

[28] Ashiq Khan, Dan Jurca, Kazuyuki Kozu, Wolfgang Kellerer, and Masami Yabusaki. The Reconfigurable Mobile Network. In *Communications Workshops (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[29] Yasir Zaki, Liang Zhao, Carmelita Goerg, and Andreas Timm-Giel. LTE wireless virtualization and spectrum management. In *Wireless and Mobile Networking Conference (WMNC), 2010 Third Joint IFIP*, pages 1–6. IEEE, 2010.

[30] Yasir Zaki, Liang Zhao, Carmelita Goerg, and Andreas Timm-Giel. A Novel LTE Wireless Virtualization Framework. *Mobile Networks and Management*, pages 245–257, 2011.

[31] Yasir Zaki, Liang Zhao, Carmelita Goerg, and Andreas Timm-Giel. LTE mobile network virtualization. *Mobile Networks and Applications*, 16(4):424–432, 2011.

[32] MCN D3.1, "Infrastructure Management Foundations - Specifications and Design for Mobile Cloud framework", deliverable 3.1, EU FP7 European Commission, EU FP7 Mobile Cloud Networking public deliverable, November 2013.

[33] MCN D4.1, "Mobile Network Cloud Component Design", deliverable 4.1, European Commission, EU FP7 Mobile Cloud Networking public deliverable, November 2013.

[34] Rajkumar Buyya, Mukaddim Pathan, and Athena Vakali. *Content Delivery Networks*, volume 9. Springer, 2008.

[35] 3GPP Technical Specification 23.829: Local IP Access and Selected IP Traffic Offload; Release 10 V1.3 (2010), "`www.3gpp.org`.

[36] Overview of 3GPP Release 10 V0.1.7 (2013-01), `www.3gpp.org/Release-10`.

[37] Overview of 3GPP Release 11 V0.1.3 (2013-01), `www.3gpp.org/Release-11`.

[38] Overview of 3GPP Release 12 V0.0.6 (2013-01), `www.3gpp.org/Release-12`.

[39] 3GPP Technical Specification 23.859: LIPA Mobility and SIPTO at the Local Network; Release 11 and 12 (2012), "`www.3gpp.org`.

[40] IETF Distributed Mobility Management (DMM) Working Group. Archive: `http://tools.ietf.org/wg/dmm/`.

[41] Dapeng Liu, Hidetoshi Yokota, Pierrick Seite, Jouni Korhonen, and H Anthony Chan (editor). Requirements for Distributed Mobility Management. IETF Internet draft (work in progress), 2013.

[42] Marco Liebsch, Georgios Karagiannis, and Pierrick Seite. Distributed Mobility Management-Framework & Analysis. IETF Internet draft (work in progress), 2013.

[43] 3GPP Technical Specification 23.401: General Packet Radio Service enhancements for Evolved Universal Terrestrial Radio Access Network access; Technical report, 2010, `www.3gpp.org`.

[44] 3GPP Technical Specification 32.251: Packet Switched (PS) domain charging; Release 11 (July 2013), www.3gpp.org.

[45] Marco Liebsch. Per-Host Locators for Distributed Mobility Management. IETF Internet draft (work in progress), 2013.

[46] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[47] The OpenFlow Switch Specification, Version 1.3.0. Available at http://archive.openflow.org.

[48] Lisa M Ellram. Total cost of ownership: an analysis approach for purchasing. *International Journal of Physical Distribution & Logistics Management*, 25(8):4–23, 1995.

[49] Theophilus Benson, Aditya Akella, and David A Maltz. Unraveling the complexity of network management. In *NSDI*, pages 335–348, 2009.

[50] Kurt Tutschku and Phuoc Tran-Gia. 23. traffic characteristics and performance evaluation of peer-to-peer systems. In *Peer-to-Peer Systems and Applications*, pages 383–397. Springer, 2005.

[51] https://github.com/lucval/dmm/.

[52] http://www.nsnam.org.

[53] http://iptechwiki.cttc.es/LTE-EPC_Network_Simulator_LENA.

[54] Network Address Translation model for NS3. http://www.nsnam.org/wiki/index.php/GSOC2012NetworkAddressTranslation.

[55] https://github.com/lucval/dmm/tree/master/nat/.

[56] OpenFlow switch support. http://www.nsnam.org/docs/release/3.13/models/html/openflow-switch.html.

[57] https://github.com/lucval/dmm/tree/master/openflow/.

[58] http://www.cs.washington.edu/research/networking/rocketfuel/.

[59] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring isp topologies with rocketfuel. *ACM SIGCOMM Computer Communication Review*, 32(4):133–145, 2002.

[60] https://github.com/lucval/dmm/tree/master/lte/.

[61] https://github.com/lucval/dmm/tree/master/csma/.

[62] NGMN Alliance, "NGMN Radio Access Performance Evaluation Methodology".

[63] Farooq Khan. *LTE for 4G mobile broadband: air interface technologies and performance*. Cambridge University Press, 2009.

[64] Giang T. Pham. Integration of IEC 61850 MMS and LTE to support smart metering communications. Master's thesis, University of Twente, August 2013. Available online at: `http://www.utwente.nl/ewi/dacs/assignments/completed/master/reports/report-Giang.pdf`.

[65] A.D Nguyen. Integration of IEC 61850 MMS and LTE to support remote control communications in electricity distributuion grid. Master's thesis, University of Twente, August 2013. Available online at: `http://www.utwente.nl/ewi/dacs/assignments/completed/master/reports/report-Dung-Nguyen.pdf`.

[66] `https://github.com/lucval/dmm/tree/master/applications/`.

[67] Doreid Ammar, Thomas Begin, and Isabelle Guerin-Lassous. A new tool for generating realistic internet traffic in ns-3. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pages 81–83. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[68] Zygmunt J Haas and Yi-Bing Lin. On optimizing the location update costs in the presence of database failures. *Wireless Networks*, 4(5):419–426, 1998.

[69] Serge P Hoogendoorn. Vehicle-type and lane-specific free speed distributions on motorways: A novel estimation approach using censored observations. *Transportation Research Record: Journal of the Transportation Research Board*, 1934(1):148–156, 2005.

[70] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[71] `https://github.com/lucval/dmm/tree/master/examples/`.

[72] 3GPP Technical Specification 36.104, LTE Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception; version 11.5.0 (Release 11), `www.3gpp.org`.

[73] Buffer sizing in the Internet,. `http://yuba.stanford.edu/buffersizing/`.

[74] Damon Wischik and Nick McKeown. Part i: Buffer sizes for core routers. *ACM SIGCOMM Computer Communication Review*, 35(3):75–78, 2005.

[75] Yashar Ganjali and Nick McKeown. Update on buffer sizing in internet routers. *ACM SIGCOMM Computer Communication Review*, 36(5):67–70, 2006.

[76] Mary Natrella. NIST/SEMATECH e-handbook of statistical methods. Available on-line: http://www.itl.nist.gov/div898/handbook/eda/section3/eda352.htm.

[77] Quality of Service for Voice over IP - Cisco. http://www.cisco.com/en/US/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.pdf.

[78] Latency and QoS for Voice over IP - SANS Institute. http://www.sans.org/reading-room/whitepapers/voip/latency-qos-voice-ip-1349?show=latency-qos-voice-ip-1349&cat=voip.

[79] Rackspace Cloud Computing. OpenStack Open Source Cloud Computing Software. http://www.openstack.org.

[80] http://www.voip-info.org/wiki/view/QoS.

Appendix A

Guideline to setup and use the source code in NS3

A.1 Installation

A.1.1 Install NS3

The NS3 installation instructions are available at:
`http://www.nsnam.org/wiki/index.php/Installation`

A.1.2 Install the modified LTE module in NS3

The source code of the modified LTE module following the modifications introduced in Section 4.2.1.1 is available at:
`https://github.com/lucval/dmm/tree/master/lte`
Change directory to the source code directory of NS3
`cd NS3_path>/src`
Delete entirely the original `lte` folder
`rm -rf lte`
Clone a copy of the modified LTE repository
`git clone https://github.com/lucval/lte.git`
Compile the module after moving back to the NS3 directory
`cd ..`
`./waf`
Waf will start compiling the new module.

A.1.3 Install the modified CSMA module in NS3

The source code of the modified CSMA module implementing gigabit Ethernet is available at:
`https://github.com/lucval/dmm/tree/master/csma`
Change directory to the source code directory of NS3
`cd <NS3_path>/src`
Delete entirely the original `csma` folder
`rm -rf csma`
Clone a copy of the modified CSMA repository
`git clone https://github.com/lucval/csma.git`
Compile the module after moving back to the NS3 directory
`cd ..`
`./waf`
Waf will start compiling the new module.

A.1.4 Install the LTE background traffic module in NS3

The source codes for the LTE UDP background traffic is available at:
`https://github.com/tgpham/gen-udp`
Change directory to the source code directory of NS3
`cd <NS3_path>/src`
Clone a copy of the LTE background traffic repository
`git clone https://github.com/tgpham/gen-udp.git`
Compile the module after moving back to the NS3 directory
`cd ..`
`./waf`
Waf will start compiling the new module.
* Note: If there is a problem with the build, the following adjustments need to be made in the `wscript` file in the modules:
Change from `headers = bld.new_task_gen(features=['ns3header'])`
to `headers = bld (features=['ns3header'])`
(remove `.new_task_gen`, as for WAF 1.7 and above it will cause errors)
The same procedure needs to be repeated for the LTE TCP background traffic module which can be cloned as follows
`git clone https://github.com/lucval/http.git`

A.1.5 Install the wired link background traffic generator in NS3

The source codes of the PPBP application can be downloaded from:
`http://perso.ens-lyon.fr/thomas.begin/NS3-PPBP.zip`

Extract the content of the downloaded archive in a temporary folder.

Move the `PPBP-application.cc` and `PPBP-application.h` files into `<NS3_path>/src/applications/model/`

Move the `PPBP-helper.cc` and `PPBP-helper.h` files into `<NS3_path>/src/applications/helper/`

Change directory to

```
cd <NS3_path>/src/applications/
```

Modify the `wscript` file adding `'model/PPBP-application.cc'` and `'helper/PPBP-helper.cc'` to the `module.source` list and `'model/PPBP-application.h'` and `'helper/PPBP-helper.h'` to the `headers.source` list.

Compile the module after moving back to the NS3 directory

```
cd ../..
./waf
```

Waf will start compiling the updated `applications` module.

A.2 Setup Rocketfuel-based network topology in NS3

An example of file representing a network topology derived using Rocketfuel can be found at:

https://github.com/lucval/dmm/blob/master/examples/topology-example

The file represents the topology used in this report and depicted in Figure 4.5 of Section 4.2. The path where the downloaded file has been placed needs to be passed to the application. Instructions on how to setup the downloaded topology can be found below.

```
/** ************ CREATE BACKBONE NETWORK TOPOLOGY ************ **/
std::string format ("Rocketfuel");
std::string input;
input = "src/topology-read/examples/topology-example";

Ptr<TopologyReader> inFile = 0;
TopologyReaderHelper topoHelp;

NodeContainer backboneNodes;

topoHelp.SetFileName (input);
topoHelp.SetFileType (format);
inFile = topoHelp.GetTopologyReader ();

if (inFile != 0)
  {
    backboneNodes = inFile->Read ();
  }

if (inFile->LinksSize () == 0)
  {
```

```
    NS_LOG_ERROR ("Problems reading the topology file. Failing.");
    return -1;
  }

NS_LOG_INFO ("creating internet stacks");
InternetStackHelper stack;
stack.Install (backboneNodes);

NS_LOG_INFO ("creating ip4 addresses");
Ipv4AddressHelper opNetAddress;
opNetAddress.SetBase ("10.0.0.0", "255.255.0.0");

int totlinks = inFile->LinksSize ();

NS_LOG_INFO ("creating node containers");
NodeContainer* nc = new NodeContainer[totlinks];
TopologyReader::ConstLinksIterator iter;
int i = 0;
for ( iter = inFile->LinksBegin (); iter != inFile->LinksEnd (); iter++, i++ )
  {
    nc[i] = NodeContainer (iter->GetFromNode (), iter->GetToNode ());
  }

NS_LOG_INFO ("creating net device containers");
NetDeviceContainer* ndc = new NetDeviceContainer[totlinks];
CsmaHelper csmah;
csmah.SetDeviceAttribute("Mtu", UintegerValue (1500));
for (int i = 0; i < totlinks; i++)
        ndc[i] = csmah.Install (nc[i]);

// it crates little subnets, one for each couple of nodes
NS_LOG_INFO ("creating ipv4 interfaces");
Ipv4InterfaceContainer* ipic = new Ipv4InterfaceContainer[totlinks];
for (int i = 0; i < totlinks; i++)
  {
      ipic[i] = opNetAddress.Assign (ndc[i]);
      opNetAddress.NewNetwork ();
  }
```

A.3 Setup ARP tables in NS3

Since gigabit Ethernet is used in the wired network, the ARP tables need to be initialized before the simulation begins. An example on how to do so is shown below with `n` being a `NodeContainer` containing all the backbone network's nodes.

```
/** ******************* SETUP ARP TABLES ****************** **/
Ptr<ArpCache> arp = CreateObject<ArpCache>();
arp->SetAliveTimeout(Seconds(3600 * 24 * 365));
for(uint16_t i=0; i<n.GetN(); i++){
  Ptr<Ipv4L3Protocol> ip = n.Get(i)->GetObject<Ipv4L3Protocol>();
  NS_ASSERT(ip!=0);
  int ninter = (int)ip->GetNInterfaces();
  for(int j = 0; j < ninter; j++) {
    Ptr<Ipv4Interface> ipIface = ip->GetInterface(j);
    NS_ASSERT(ipIface != 0);
    Ptr<NetDevice> device = ipIface->GetDevice();
    NS_ASSERT(device != 0);
    Mac48Address addr = Mac48Address::ConvertFrom(device->GetAddress ());
    for(uint32_t k = 0; k < ipIface->GetNAddresses (); k ++) {
      Ipv4Address ipAddr = ipIface->GetAddress (k).GetLocal();
      if(ipAddr == Ipv4Address::GetLoopback())
         continue;
      ArpCache::Entry * entry = arp->Add(ipAddr);
      entry->MarkWaitReply(0);
      entry->MarkAlive(addr);
    }
  }
}

for(uint16_t i=0; i<n.GetN(); i++){
  Ptr<Ipv4L3Protocol> ip = n.Get(i)->GetObject<Ipv4L3Protocol>();
  NS_ASSERT(ip!=0);
  int ninter = (int)ip->GetNInterfaces();
  for(int j = 0; j < ninter; j++) {
    Ptr<Ipv4Interface> ipIface = ip->GetInterface(j);
    ipIface->SetArpCache(arp);
  }
}
```

A.4 Setup LTE networks with two S-/P-GWs in NS3

In order to implement an LTE network with two S-/P-GWs, several steps need to be followed. The steps have to be executed in the exact order specified below. Please refer to the NS3 LENA documentation [6] for the complete source code.

1) Create the S-/P-GW nodes and install IP stack on each one of them
2) Connect them to the Internet or IMS network
3) Create the source and target eNodeBs and install Mobility Model and IP stack on each one of them
4) Instantiate the `lteHelper`
5) Instantiate the `epcHelper` for the source S-/P-GW passing the corresponding S-/P-GW node as

attribute: `Ptr<EpcHelper> epcHelper1 = CreateObject<EpcHelper> (sourceSPgw);`
6) Setup `epcHelper1` within the `lteHelper`
7) Install LTE devices to source eNodeB which will establish the S1-U tunnel with the source S-/P-GW entity
8) Create the UEs and install Mobility Model and IP stack on each one of them
9) Install LTE devices to the UEs
10) Attach UEs to the source eNodeB
11) Configure the frequency band for the target eNodeB
12) Instantiate the `epcHelper` for the target S-/P-GW passing the corresponding S-/P-GW node as attribute: `Ptr<EpcHelper> epcHelper2 = CreateObject<EpcHelper> (targetSPgw);`
13) Setup `epcHelper2` within the `lteHelper`
14) Install LTE devices to target eNodeB which will establish the S1-U tunnel with the target S-/P-GW entity
15) Create the UEs and install Mobility Model and IP stack on each one of them
16) Install LTE devices to the UEs
17) Attach UEs to the target eNodeB
18) Setup IP address continuity in the network in 6 steps:

```
Ipv4AddressHelper tunPool1 = epcHelper1->GetHelper();
Ipv4AddressHelper tunPool2 = epcHelper2->GetHelper();
Ipv4AddressHelper tunPool2new = epcHelper1->AddTunDevice(tunPool2);
Ipv4AddressHelper tunPool1new = epcHelper2->AddTunDevice(tunPool1);
epcHelper1->ReSetHelper(tunPool1new);
epcHelper2->ReSetHelper(tunPool2new);
```

19) Create the MME node and install IP stack on it
20) Setup S11 interfaces as follows:

```
lteHelper->AttachMme(mme, epcHelper1->GetUeDefaultGatewayAddress());
lteHelper->AttachMme(mme, epcHelper2->GetUeDefaultGatewayAddress());
```

A.5 Schedule handover with S-/P-GW relocation in NS3

Source and target eNodeBs need to be neighboring cells to correctly perform an handover with S-/P-GW relocation in NS3. The source and target S-/P-GW needs to be connected, by means of S11 interface, to the EPC MME entity.

The following function schedules an handover with S-/P-GW relocation:

```
void LteHelper::DoHandoverRequestWithAnchorRelocation (Ptr<Node> ue, Ipv4Address pgwAddress,
Ptr<NetDevice> sourceEnbDev, Ptr<NetDevice> targetEnbDev) { ...  }
```

X2 data forwarding can be used if desired by adding the X2 interfaces to the eNodeBs (see NS3 LENA documentation [6] for the complete source code).

A.6 Use LTE background traffic module in NS3

An example of using the LTE background traffic module is included in the `lte-BgNodes.cc` script available at `<NS3_path>/src/gen-udp/examples/`. Every section in the source code is commented to help the user with the usage of the code.
Some of the key points in using the module are listed as follows.
The LTE UE nodes are created based on the traffic mix specified in [62]:

```
// Create Voice UEs (30% of the nodes)
NodeContainer lteVoiceUeContainer;
lteVoiceUeContainer.Create((float)0.3*numberOfBgNodes);
// Create Video UEs (20% of the nodes)
NodeContainer lteVideoUeContainer;
lteVideoUeContainer.Create((float)0.2*numberOfBgNodes);
// Create Gaming UEs (20% of the nodes)
NodeContainer lteGamingUeContainer;
lteGamingUeContainer.Create((float)0.2*numberOfBgNodes);
// Create HTTP UEs (20% of the nodes)
NodeContainer lteHttpUeContainer;
lteHttpUeContainer.Create((float)0.2*numberOfBgNodes);
// Create FTP UEs (the rest)
NodeContainer lteFtpUeContainer;
lteFtpUeContainer.Create(numberOfBgNodes-lteVoiceUeContainer.GetN()
-lteVideoUeContainer.GetN()
-lteGamingUeContainer.GetN()
-lteHttpUeContainer.GetN());
```

The remote hosts are created to allow each type of traffic to be transfered:

```
// Create Voice Remote host to send/receive Voice traffic to/from
Voice UEs NodeContainer lteVoiceRemoteContainer;
lteVoiceRemoteContainer.Create(1);
// Create Video Remote host to send/receive Video traffic to/from
Video UEs NodeContainer lteVideoRemoteContainer;
lteVideoRemoteContainer.Create(1);
// Create Gaming Remote host to send/receive Gaming traffic to/from
Gaming UEs NodeContainer lteGamingRemoteContainer;
lteGamingRemoteContainer.Create(1);
// Create FTP Remote host to send/receive FTP traffic to/from FTP UEs
NodeContainer lteFtpRemoteContainer;
lteFtpRemoteContainer.Create(1);
// Create a number of HTTP Server, one for each UEs
NodeContainer lteHttpRemoteContainer;
lteHttpRemoteContainer.Create(lteHttpUeContainer.GetN());
```

The LTE background traffic server and client are defined using the `GeneralUdpServerHelper` and `GeneralUdpClientHelper`. There is also an additional parameter to specify the exact traffic type

(Video=0, Gaming uplink=1, Gaming downlink=2, VoIP=3), and it must be declared when adding `ApplicationContainer` to the node. For example, the following part of the script creates uplink and downlink video traffic.

```
// ----------------------------------------------------------------------
// Video Application, both UL and DL

// UPLINK (from UEs)
//
// Create one Video applications on remote host.
//
uint16_t lteVideoRemotePort = 5000;
GeneralUdpServerHelper lteVideoRemoteServer (lteVideoRemotePort, 0);
ApplicationContainer lteVideoUeApp = lteVideoRemoteServer.Install (lteVideoRemoteNode);
lteVideoUeApp.Start (Seconds (0.0));
lteVideoUeApp.Stop (Seconds (1000.0));
//
// Create one Video application to send UDP datagrams from UE nodes to
// Remote Video host.
//
GeneralUdpClientHelper VideoClientUe (lteVideoRemoteAddress, lteVideoRemotePort, 0);
lteVideoUeApp = VideoClientUe.Install (lteVideoUeContainer);
lteVideoUeApp.Start (Seconds (0.1));
lteVideoUeApp.Stop (Seconds (100.0));

// DOWNLINK (to UEs)
//
// Create Video applications on UE nodes.
//
uint16_t lteVideoUePort = 5000;
GeneralUdpServerHelper lteVideoUeServer (lteVideoUePort, 0);
ApplicationContainer lteVideoRemoteApp = lteVideoUeServer.Install (lteVideoUeContainer);
lteVideoRemoteApp.Start (Seconds (0.0));
lteVideoRemoteApp.Stop (Seconds (1000.0));
//
// Create one Video application to send UDP datagrams from Remote Host to VoIP UEs.
//
for (uint32_t i = 0; i < lteVideoUeInterface.GetN(); i++)
{
  GeneralUdpClientHelper VideoClientRemote (lteVideoUeInterface.GetAddress(i),
        lteVideoUePort, 0);
  lteVideoRemoteApp = VideoClientRemote.Install (lteVideoRemoteNode);
  lteVideoRemoteApp.Start (Seconds (0.1));
  lteVideoRemoteApp.Stop (Seconds (100.0));
}
```

A.7 Use PPBP-application in NS3

The parameters used by the PPBP-application to generate the wired link background traffic can be setup as follows:

```
Config::SetDefault ("ns3::PPBPApplication::MeanBurstArrivals",
        RandomVariableValue (ConstantVariable (2000)));
Config::SetDefault ("ns3::PPBPApplication::MeanBurstTimeLength",
        RandomVariableValue (ConstantVariable (0.02)));
Config::SetDefault ("ns3::PPBPApplication::BurstIntensity",
        DataRateValue (DataRate ("20Mb/s")));
```

In the above the `PPBP-application` has been setup following what specified in Section 4.2.4.2 of this report.

In order to generate the desired background traffic, install the `PPBP-application` on the node where the desired background traffic will be generated (`client`) and install a `packet-sink` application on the node where the background traffic will be stopped (`server`). Clearly the server node needs to be reachable by the client node. A simple example can be found below.

```
// Create Packet Sink on server (1001 is the UDP port used by the sink)
Address sinkLocalAddress (InetSocketAddress (Ipv4Address::GetAny (), 1001));
PacketSinkHelper packetSinkHelper ("ns3::UdpSocketFactory", sinkLocalAddress);
ApplicationContainer sinkApps;
sinkApps.Add (packetSinkHelper.Install (server));
sinkApps.Start (Seconds (0.0));
// Create PPBP-application sending packets to the sink
// (1001 is also the UDP port used by the PPBP application)
PPBPHelper clientHelper ("ns3::UdpSocketFactory", Address ());
ApplicationContainer clientApps;
AddressValue remoteAddress (InetSocketAddress (clientAddress, 1001));
clientHelper.SetAttribute ("Remote", serverAddress);
clientApps.Add (clientHelper.Install (client));
clientApps.Start (Seconds (0.1)); // Start after sink
```

A.8 Setup and use Double NAT in NS3

The instructions on how to integrate the NAT module in NS3 and on how to use it can be found at:
`http://www.nsnam.org/wiki/index.php/GSOC2012NetworkAddressTranslation`

In order to correct setup a Double NAT solution the following steps need to be done:
1) Create Ingress and Egress nodes and install the IP stack on each one of them
2) Instantiate an `Ipv4NatHelper`
3) Install Ipv4Nat capabilities on the Ingress and Egress nodes
4) Create the NAT Controller node and install the IP stack on it

5) Add the controller to the `lteHelper`: `lteHelper->CreateController(natController);`
6) Create TCP connection(s) between the Ingress NAT router(s) and the NAT controller:
`lteHelper->ConnectSocketsIngress(IngressNatNode1,IngressNatNode1Ipv4Address,IngressIpv4Nat);`
7) Create TCP connection(s) between the Egress NAT router(s) and the NAT controller:
`lteHelper->ConnectSocketsIngress(EgressNatNode1,EgressNatNode1Ipv4Address,EgressIpv4Nat, targetSPgwIpv4Address);`

An example on how to completely setup a Double NAT DMM solution in a LTE system with two S-/P-GWs is available at:
`https://github.com/lucval/dmm/blob/master/examples/double-nat.cc`

A.9 Setup and use OpenFlow in NS3

The instructions on how to integrate the OpenFlow module in NS3 and on how to use it can be found at:
`http://www.nsnam.org/docs/release/3.13/models/html/openflow-switch.html`
In order to setup a Partial OpenFlow solution the following steps need to be done:
1) Create the Ingress and Egress OpenFlow nodes and install the IP stack on each one of them
2) Instantiate an `OpenFlowSwitchHelper`
3) Create an OpenFlow controller node and a Learning Controller as follow:
`Ptr<ns3::ofi::LearningController> ofController = CreateObject<ns3::ofi::LearningController>();`
`ofController->SetAttribute ("ExpirationTime", TimeValue (Seconds(0)));`
4) Install the OpenFlow module on each Ingress and Egress node setting the instantiated Learning Controller as controller
5) Add the controller node to the `lteHelper`: `lteHelper->CreateController(controller);`
6) Create TCP connection(s) between the Ingress switch(es) and the OpenFlow controller node:
`lteHelper->ConnectSocketsIngress(IngressOfNode1,IngressOfNode1Ipv4Address);`
7) Create TCP connection(s) between the Egress switch(es) and the OpenFlow controller node:
`lteHelper->ConnectSocketsIngress(EgressOfNode1,EgressOfNode1Ipv4Address,targetSPgwIpv4Address);`
Repeat steps from 1 to 6 to setup a Full OpenFlow solution by simply implementing each OpenFlow switch as a Partial OpenFlow's Ingress switch.

An example on how to completely setup a Partial OpenFlow DMM solution in a LTE system with two S-/P-GWs is available at:
`https://github.com/lucval/dmm/blob/master/examples/partial-openflow.cc`
While an example on how to completely setup a Full OpenFlow DMM solution in a LTE system with two S-/P-GWs is available at:
`https://github.com/lucval/dmm/blob/master/examples/full-openflow.cc`