MASTER THESIS

SEAMLESS SERVICE CONTINUITY IN CLOUD BASED LTE SYSTEMS

Triadimas Arief Satria

TELEMATICS
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.

18/11/13

Abstract

Cloud computing model offers better network resource utilization by pooling shared computing resources that can be rapidly provisioned and released. The increasing use of mobile devices (e.g. smart phones, tablets, etc.) has increased the complexity in provisioning cellular network resources. Applying cloud computing model in LTE systems 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.

One of the most important concepts used in mobile networks is service continuity. Mobile users moving from one sub-network to another sub-network should be able to seamlessly continue retrieving content and use services (e.g. video streaming, gamming, VoIP, etc.) that they want. In cloud based LTE systems, services are hosted on Virtual Machines (VMs) that can move and migrate across multiple networks to locations that are the best to deliver services to mobile users. The migration of VMs and/or service/functionality instances running on such VMs should happen without losing service continuity.

In this thesis, Content Centric Networking (CCN) is evaluated and verified whether it is possible to be implemented in cloud based LTE systems as a solution for service continuity. Several integration options of CCN in LTE systems and the proposed solutions to support VM and/or content migration are described in detail. Two sets of experiments that focus on "Content Migration Support" and "VM and Content Migration Support" are implemented and performed using ns-3 LENA simulation environment together with ndnSIM. The simulation results demonstrate that the proposed solution can support seamless service continuity when content migration occurs. However, in order to support service continuity when VM migration occurs, the implementation of solutions such as mobility prediction systems are needed to predict the movement of users and determine when the VM migration should be triggered in advance.

Acknowledgement

Alhamdulillah, all praises to Allah SWT for the strengths and His blessing in completing this thesis - the final project of my two-year Master program in the University of Twente, the Netherlands. This research was performed as a part of EU FP7 Mobile Cloud Networking (MCN) project under direct supervision of Dr. ir. Georgios Karagiannis from Design and Analysis of Communication Systems (DACS) group, University of Twente.

First and foremost, I would like to express my highest appreciation to my supervisor, Dr. ir. Georgios Karagiannis for the insightful feedbacks, dedicated knowledge, and continuous support of my Master thesis research. His guidance helped me in all the time of research and writing of this thesis.

I would also show my appreciation to my co-supervisor, Morteza Karimzadeh for his encouragement, inspiration, and support during this project. Additionally, I sincerely thank to Dr. Ir. Geert Heijenk for the comments during the starting and mid-term presentations which help me a lot to accomplish this research. Moreover, I am thankful to my best partner in this research, Luca Valtulina for the extensive discussion, close collaboration and great time exchanging ideas.

I gratefully acknowledge the Ministry of Communication and Information Technology of the Republic of Indonesia for the funding sources. Without their scholarship, it would be impossible for me to experience these wonderful two years journey of Master program in Telematics.

Finally, I take this opportunity to express my gratitude towards my beloved family: both of my parents, my wife Vanya Vabrina Valindria, and my son Rhapsody Radva Rasheed, for their prayers, care and love during this Master program in the University of Twente. To those who indirectly contributed in this research, your kindness means a lot to me. Thank you very much.

18 November 2013

Triadimas Arief Satria

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	3rd Generation Partnership Project
API	Application Programming Interface
ASAP	Aggregate Server Access Protocol
BE	Base Exchange
CBR	Constant Bit Rate
CC	Cloud Controller
CCN	Content-Centric Networking
CDF	Cumulative Distribution Function
CDN	Content Delivery Network
CM	Container Manager
CN	Correspondent Node
CS	Content Store
DL-TFT	Downlink Traffic Flow Template
DONA	Data-Oriented Network Architecture
DRP	Directory Relay Protocol
EEU	Enterprise End User
eNodeB	Evolved NodeB
ENRP	Endpoint Name Resolution Protocol
EPC	Evolved Packet Core
EPCaaS	Evolved Packet Core as a Service
EPS	Evolved Packet System
ESP	Encapsulated Security Payload
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FIB	Forwarding Information Base
FMA	Flow Mapping Agent
FMT	Flow Mapping Table
Gbps	Gigabit per second
GPRS	General Packet Radio Access
GTP	GPRS Tunneling Protocol
HI	Host Identifier
HIP	Host Identity Protocol
HIT	Host Identity Tag
HR	Home Registrar
HSS	Home Subscriber System

IaaS	Infrastructure as a Service
ICN	Information Centric Networking
ICN/CDNaaS	Information Centric Networking/Content Delivery Network as a Service
ILNP	Identifier-Locator Network Protocol
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
InP	Infrastructure Provider
IP	Internet Protocol
IPsec	IP Security
ISP	Internet Service
LENA	LTE-EPC Network Simulator
LMA	Local Mobility Anchor
LTE	Long Term Evolution
MAC	Medium Access Control
MAG	Mobile Access Gateway
Mbps	Megabit per second
MCDN	Mobile Content Distribution Networks
MCN	Mobile Cloud Networking
MME	Mobility Management Entity
MN	Mobile Node
MOBaaS	Mobility and Bandwidth Availability prediction as a Service
mRVS	Mobile Rendezvous Server
MSM	Mobility Support Manager
NAS	Non-Access Stratum
NBRP	Name-Based Routing Protocol
ndnSIM	Named Data Networking Simulator
NDO	Named Data Object
NEMO	Network Mobility
NetInf	Network of Information
NGNM	Next Generation Mobile Network
NIST	National Institute of Standards and Technology
NRS	Name Resolution Service
ns-3	Network Simulator 3
NVE	network virtualization environment
NVS	Network Virtualization Substrate
ONF	Open Networking Foundation
PaaS	Platform as a Service

PBU	Proxy Binding Update
PDCP	Packet Data Convergence Protocol
P-GW	Packet Data Network Gateway
PIT	Pending Interest Table
PMIPv6	Proxy Mobile IPv6
PMIPv6	Proxy Mobile IPv6
POP	Point of Presence
PPBP	Poisson Pareto Burst Process
PSIRP	Publish-Subscribe Internet Routing Paradigm
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RANaaS	Radio Access Network as a Service
RBID	Radio Bearer ID
RH	Resolution Handler
RI	Rendezvous Identifier
RLC	Radio Link Control
RMN	Reconfigurable Mobile Network
RNTI	Radio Network Temporary Identifier
RRC	Radio Resource Control
RSerPool	reliable server pooling
RTT	Round Trip Time
RVS	Rendezvous Server
SA	Security Association
SaaS	Software as a Service
SCN	Service-Centric Networking
SCTP	Stream Control Transmission Protocol
SDN	Software-Defined Networking
S-eNodeB	Source eNodeB
S-GW	Serving Gateway
SI	Scope Identifier
SIP	Session Initiation Protocol
SO	Service Orchestrator
SP	Service Provider
SPI	Security Parameter Index
TCP	Transmission Control Protocol
TEID	Tunnel Endpoint Identifier

T-eNodeB	Target eNodeB
TRIAD	Translating Relaying Internet Architecture integrating Active Directories
UDP	User Datagram Protocol
UE	User Equipment
URI	Uniform Resource Identifier
VBS	Virtual Base Station
VCP	Virtualization Controlling Platform
VLAN	virtual local area network
VM	Virtual Machine
VoIP	Voice over IP
VPN	virtual private network
VR	Visited Registrar
WNC	Wireless Network Cloud

Chapter 1

Introduction

Cloud computing is a model which is very popular because of its various benefits and conveniences provided. The cloud computing model offers better network resource utilization by pooling shared computing resources that can be rapidly provisioned and released. Virtualization technologies are used to realize resource sharing and dynamic resource provisioning. Basically, service models of cloud computing can be divided into three models, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). For business owners, cloud computing is a promising technology that provides many benefits, such as eliminating investment costs used to build a cloud infrastructure, reducing operating cost, business risk, and maintenance costs.

Various benefits offered by cloud computing have attracted researchers for applying this concept into cellular systems, such as Long Term Evolution (LTE). Nowadays, the increasing use of mobile devices (e.g. smart phones, tablets, etc.) has increased the complexity in provisioning cellular network resources. Even though LTE promises a faster and more efficient data network, its architecture is still highly centralized that can lead to very high bandwidth requirements on core network equipments. Long communication paths between users and servers can increase delay and waste network resources. 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. LTE systems that use the mobile cloud networking concept can be denoted as cloud based LTE systems.

One of the most important concepts used in mobile networks is service continuity. Mobile users moving from one sub-network to another sub-network should be able to seamlessly continue retrieving content and use services (e.g. video streaming, gamming, VoIP, etc.) that they want. In cloud based LTE systems, services are hosted on Virtual Machines (VMs) that can move and migrate across multiple networks to locations that are the best to deliver services to mobile users. The migration of VMs and/or service/functionality instances running on such VMs should happen without losing service continuity.

1.1 Cloud Computing Concepts

Cloud computing has recently emerged as a new paradigm for hosting and delivering services over the Internet. The definition of cloud computing according to National Institute of Standards and Technology (NIST) [1] is as follow: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers,

storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

There are five essential characteristics of cloud computing according to NIST. They are:

- *On-demand self-service*

  Computing resources (e.g. server time, network storage, etc.) can be provisioned automatically without requiring human interaction at the service provider's side.

- *Broad network access*

  The computing resources are delivered over the network and accessed by client applications with heterogeneous platforms (e.g. mobile phones, laptops, and PDAs).

- *Resource pooling*

  A cloud service provider's computing resources are pooled to serve multiple consumers using multi-tenant, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. The pool-based model causes physical computing resources become invisible to consumers, so that consumers do not have control or knowledge over the exact location of the resources.

- *Rapid elasticity*

  Provisioning of computing resources can be done rapidly and elastically. Consumers can use them to scale up and release them to scale down whenever they want.

- *Measured Service*

  Cloud systems have abilities to control, monitor, and optimize the usage of computing resources by leveraging a metering capability. Therefore cloud systems can provide information of the resource usage for both the provider and consumer of the utilized service.

Cloud computing leverages virtualization technologies to realize resource sharing and dynamic resource provisioning. Virtualization is used to abstract away the details of physical hardware. It provides the capability of pooling resources and dynamically allocating or reallocating virtual resources to satisfy consumers' needs. It is a useful technology that can be used to improve service capacity.

Cloud computing offers some advantages which are attractive to business owners. In cloud computing, service providers can eliminate investment costs used to build a cloud infrastructure. They can rent resources from the cloud and pay for the usage. By cloud computing, service providers can reduce their operating cost since resources in a cloud can be allocated and reallocated on demand. They can release the resource easily when service demand is low. Moreover cloud computing can

reduce business risk and maintenance costs. Service providers do not need to manage the cloud infrastructure by themselves, so that they can shift their business risks, such as hardware failures, to infrastructure providers. Besides, they also do not need to spend a significant amount of money for hardware maintenance purposes. Furthermore, services in cloud environment are highly scalable. Computing resources are pooled to serve multiple consumers, so that when there is an increase in service demand, service providers can expand their services rapidly to large scales in order to serve their consumers. Resource pooling mechanism also causes better use of resources. In addition, cloud computing allows multiple services from different service providers can be integrated easily to meet consumers' demands.

1.1.1 Cloud Computing Layered Architecture

Figure 1 shows the layered architecture of cloud computing which consists of four layers, namely hardware layer, infrastructure layer, platform layer, and application layer.

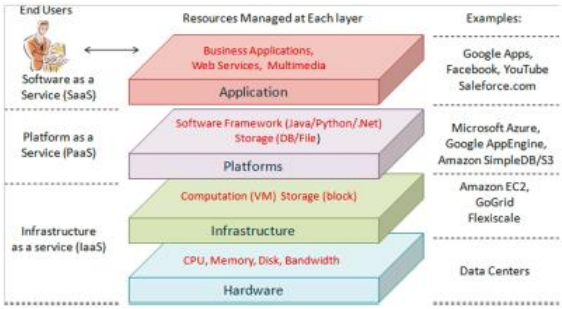

Figure 1: Cloud layered architecture, copied from [2]

The hardware layer is the layer used to manage physical resources (e.g. servers, routers, switches, storages, etc.) of cloud computing systems. The infrastructure layer is a virtualization layer that provides a pool of computing resources by partitioning the physical resources using virtualization technologies. The platform layer is the layer where operating systems and application frameworks are located. It can be used to provide Application Programming Interface (API) support for implementing storage, database, and business logic of applications. The application layer is the layer where cloud applications (e.g. business applications, web services, multimedia, etc.) are located. Every layer in cloud computing is loosely coupled to each other. Therefore, it allows cloud computing to support a wide range of application requirements.

1.1.2 Cloud Computing Service Models

Service models of cloud computing can be classified into three service models, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Figure 1 shows the service models of cloud computing.

- Infrastructure as a Service (IaaS)

  In this service model, computing resources (e.g. servers, storage, networks, etc.) are delivered as a service. IaaS allows consumers to provision resources on demand. Consumers can manage operating systems and applications through virtualization technologies, but they do not need to manage the underlying cloud infrastructure. The underlying cloud infrastructure is controlled and managed by the IaaS provider.

- Platform as a Service (PaaS)

  PaaS provides a platform, including operating system support and software development frameworks, which can be used by consumers to deploy their applications on the infrastructure offered by service provider. In this case, the deployed applications have to be created using programming languages or tools supported by the platform provided by service provider. Consumers can manage and control the deployed applications and possibly the configurations of application hosting environment, but the underlying cloud infrastructure, such as servers, operating systems, storage, etc., is managed by the service provider.

- Software as a Service (SaaS)

  This service model provides on demand application services over the internet. The applications can be accessed through various client devices connected to internet. The underlying cloud infrastructure including servers, networks, storage, operating systems, or even some configuration of the application itself is handled by the service provider.

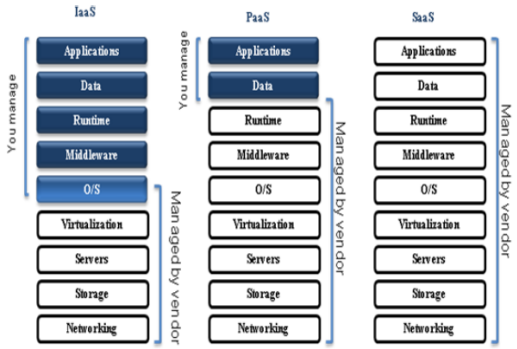

Figure 2: Cloud service models, copied from [3]

1.1.3 Cloud Computing Deployment Models

According to NIST, cloud computing can be deployed using several cloud models such as private cloud, community cloud, public cloud, and hybrid cloud.

- Private cloud

The cloud infrastructure is designed and operated exclusively for a single organization. It may be built and manage by the organization itself or an external provider. This cloud model offers an ability to optimize resource utilization, and a better control over reliability and security of cloud system.

- Community cloud

  The cloud infrastructure is shared by several organizations. They share cloud infrastructure as well as policy, security requirements, and compliance consideration. The cloud infrastructure may be managed by a third party or organizations in the community.

- Public cloud

  The cloud infrastructure is shared to the general public and it is managed by cloud service provider. Service providers can get benefit from this cloud model because there is no initial capital investment on the cloud infrastructure. The cost to build and maintain the cloud infrastructure is handled by the infrastructure provider.

- Hybrid cloud

  The cloud infrastructure is a combination of two or more cloud models (private, community, or public). In this cloud model, some service infrastructure runs in a private cloud and some others run in a public cloud. It provides on demand service expansions as well as a tighter control and security over application data compared to public cloud. Hybrid cloud is used by organizations to address some limitations in private and public clouds.

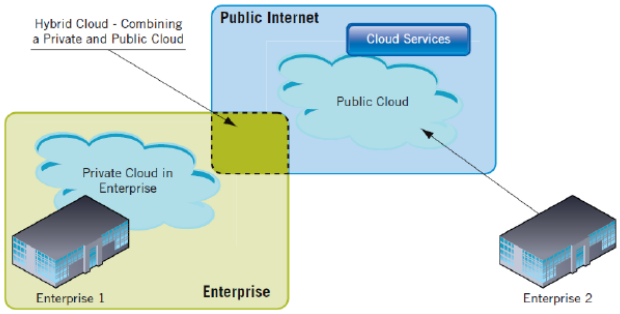

Figure 3: Private, public, and hybrid clouds, copied from [3]

1.2 Applying Cloud Computing Model in LTE Mobile Networks

Long Term Evolution (LTE) is a wireless communication technology, an evolution of the GSM/UMTS, that promises a faster and more efficient data network. It has been considered as one of the major solutions for next generation mobile systems.

Although LTE promises a faster and more efficient data network, its architecture is still highly centralized that can lead to very high bandwidth requirements on core network equipments. It also causes long communication paths between users and servers that can increase delay and waste

network resources. Implementation of cloud computing concept in LTE mobile networks can 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.

Several studies, e.g., [4] investigated how the cloud computing model could be applied into the LTE cellular system. These studies concluded that "Network Virtualization" has a vital role to apply cloud computing concept in a LTE network infrastructure. Virtualization of resources in LTE mobile networks is needed to enable multiple mobile network operators to create their own virtual network on the same infrastructure.

1.2.1 Network Virtualization

Norbert Niebert et.al, [5] proposed the use network virtualization, embedded in an architectural framework to allow the co-existence of diverse network designs and paradigms in future internet. They also explain a systematic approach to instantiating virtual networks that involves resource discovery, resource description, resource provisioning, and virtualization management. Network Virtualization will play a vital role in diversifying the future internet into separate virtual networks that are isolated from each other, and can contain operator-specific protocols and architectures which can be totally different from other co-existing virtual networks [6]. Network virtualization is considered as a mechanism that can be used to optimize the utilization of network resources, provide flexibility, promotes diversity, and promise security and increased manageability. In [4], a process of virtualizing a network which for example consists of a radio access network, mobile core network, virtual local area network (VLAN), a virtual private network (VPN), active and programmable networks and overlay networks is defined as a "network cloudification".

Figure 4 shows an example of network virtualization environment (NVE) which consists of two Virtual Networks (VN1 and VN2) with different architectures managed by different Service Providers (SPs). The virtualization of individual resources is the basis of network virtualization depicted in the figure. VN is the basic entity in NVE which is managed by a single SP even though the underlying physical resources might be provided by different Infrastructure Providers (InPs). It consists of a collection of virtual nodes connected together by a set of virtual links to form a virtual topology, which is essentially a subset of the underlying physical topology [6]. In a network virtualization environment, the roles of traditional service providers are divided into two different entities, namely Infrastructure Provider (InP), who manages the physical infrastructure, and Service Provider (SP), who leases resources from multiple InPs to deploy VNs and offers end-to-end services to end users. A service provider can provide network services to other service providers and also can create child VNs by partitioning its resources and act as a virtual InP by leasing those child networks to other SPs.

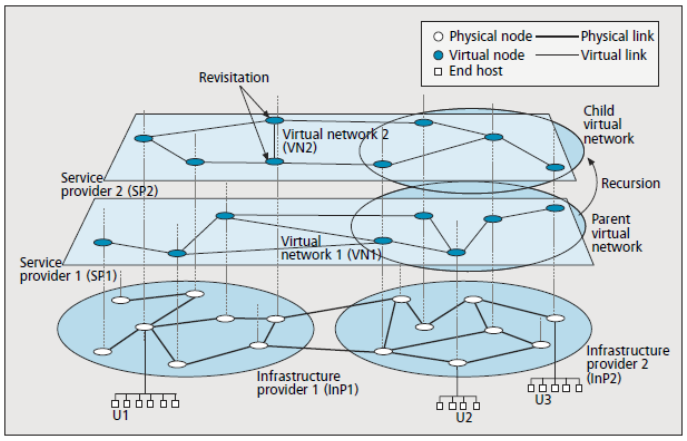

Figure 4: Network virtualization environments, copied from [6]

In the example above, there are two different InPs which are connected by physical links. SP1 operates and manages VN1 by using physical resources provided by Both InP1 and InP2. On the other hand, SP2 operates and manages VN2 by combining physical resources provided by InP1 with a child VN from service provider SP1. From the figure, we can see that SP1 provides end-to-end services to end users U2 and U3, and SP2 provides end-to-end services to end users U1 and U3.

1.2.2 Virtualization in LTE Mobile Networks

Several researches have been conducted to design architectures or network virtualizations that enable implementation of cloud computing concept in mobile communication networks. Z. Zhu et.al, [8] proposed the structure of a Virtual Base Station (VBS) pool as a step towards realizing the broader notion of a wireless network cloud (WNC). Adoption of software radio in wireless networking allows the virtualization of base stations and consolidation of virtual base stations into central pools. A. Khan et.al, [9] proposed a new reconfigurable mobile network (RMN) architecture designed for flexibility and reconfigurability. They implemented network sharing mechanisms using virtualization technologies. R. Kokku et.al, [10] described the design and implementation of a network virtualization substrate (NVS) for effective virtualization of wireless resources in cellular networks. They demonstrated the efficacy of NVS through a prototype implementation and detailed evaluation on a WiMAX testbed.

The researches mentioned above show some possible ways for virtualizing network resources in mobile communication networks. However there is still no clarity whether the solutions offered above can provide a good performance if they are implemented in LTE mobile networks.

Figure 5 shows a basic architecture of LTE Evolved Packet System (EPS) with Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access. The EPS consists of Evolved Packet Core (EPC) and E-UTRAN, and it is designed to be a purely packet switched system. It is used to provide IP connectivity using E-UTRAN. The User Equipment (UE) is connected to EPC over E-UTRAN that

consists of LTE base station called eNodeB (Evolved NodeB). The EPC is connected to the external networks that can include IP Multimedia Subsystem (IMS) via Packet Data Network Gateway (PDN GW).

Figure 5: Basic EPS architecture with E-UTRAN access, copied from [11]

In Figure 5, the EPC is composed by four network elements, namely Serving Gateway (Serving GW), Packet Data Network Gateway (PDN GW), Mobility Management Entity (MME), and Home Subscriber System (HSS). According to [11], the functions of those elements are as follows:

- **HSS:** a database that contains user-related and subscriber-related information. It provides support functions in mobility management, call and session setup, user authentication and access authorization.
- **Serving GW (S-GW):** the point of interconnection between the radio-side and the EPC. It serves the UE by routing the incoming and outgoing IP packets. It is the anchor point for the intra-LTE mobility (i.e. in case of handover between eNodeBs) and between LTE and other 3GPP accesses. It is logically connected to the other gateway, the PDN GW.
- **PDN GW (P-GW):** the point of interconnection between the EPC and the external IP networks. It routes packets to and from the PDNs. It also performs various functions such as IP address / IP prefix allocation or policy control and charging.
- **MME:** a network entity that handles the signalling related to mobility and security for E-UTRAN access. The MME is responsible for the tracking and the paging of UE in idle-mode. It is the termination point of the Non-Access Stratum (NAS).

Virtualization of components in E-UTRAN, EPC, and also operator service platforms (e.g. IMS) is needed to apply cloud computing concept in LTE mobile networks. According to [4], more than 80% of the network infrastructure is software based and already cloud computing enabled, so that virtualization of network infrastructure such as eNodeB in E-UTRAN or Serving GW in EPC could be similar to server virtualization in large data storage systems.

P. Bosch et.al, [12] introduced Virtual Telco aimed to simplify the management of deployed telecommunication services by using a cloud computing approach. Its main objective is to replace the

costly dedicated hardware implementing several centralized control plane functions and other services with distributed solutions that may be allocated on-demand over a pool of dependable, dynamically contracted computing and networking resources that are easy to manage. They proposed the case of the distributed mobility management entity (MME) for next-generation LTE cellular networks.

Y. Zaki et.al, [6] proposed a general framework for virtualizing the wireless medium that can be implemented in LTE. They virtualized the eNodeB by adding "Hypervisor" on top of the physical resources. The Hypervisor is responsible for allocating the physical resources and also the air interface resources (LTE spectrum) between different virtual eNodeBs or virtual operators. Figure 6 shows the Virtualized LTE eNodeB protocol stack.

The Hypervisor schedules the air interface resources between the different virtual operators based on the information collected from individual eNodeB, such as user channel conditions, loads, priorities, QoS requirements and information related to the contract of each of the virtual operators. The "Spectrum configuration and Bandwidth estimation" ,in the Figure 6, is responsible for setting the spectrum that the virtual eNodeB is supposed to operate in as well as estimating the required bandwidth of the virtual operator. The "Spectrum allocation unit" is responsible for scheduling the spectrum among the different virtual eNodeBs.

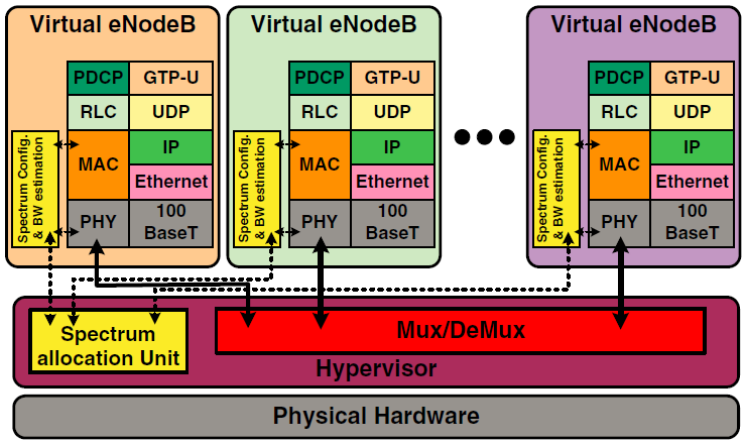

Figure 6: Virtualized LTE eNodeB protocol stack, copied from [6]

Virtualization of EPC and operator service platform, e.g., IMS, can be done in the same manner, as described in [6]. By virtualizing LTE's components in e-UTRAN, EPC, and also operator service platforms, the cloud computing concept can be implemented and can provide many benefits in LTE mobile networks (see also [53]).

1.3 Problem Statements and Research Questions

One of the features that need to be supported in mobile networks is service continuity. In cloud based LTE systems, mobile users moving from one sub-network to another sub-network should be able to

seamlessly continue retrieving content and use services (e.g. video streaming, gamming, VoIP, etc.) that they want. A migration of IP sessions should not provide any impact on service continuity; thus the mobile user will be able to continue getting services seamlessly from the network. Services or content requested by users should be able to migrate across networks to locations close to users in order to minimize delay and maximize throughput.

In cloud based LTE systems, virtualized EPS components are hosted on Virtual Machines (VMs) that can be moved and migrated across multiple networks to locations that are the best to deliver services to mobile users. The migration of VMs should happen without losing service continuity.

The main research question in this assignment is:

"How could seamless service continuity be implemented and evaluated in cloud based LTE systems?"

In order to answer the main research question, this question is divided into five sub-questions as follows:

1. What are the requirements that need to be satisfied by a service continuity solution when it is applied in cloud based LTE systems?
2. Which service continuity solutions could be implemented in cloud based LTE systems?
3. Which architecture/framework could be used to support service continuity in cloud based LTE systems?
4. How could the service continuity solution be applied in cloud based LTE systems?
5. How could the service continuity solution be evaluated and verified whether it is seamless?

1.4 Organization of Report

This report is organized as follows:

Chapter 2 describes how the concept of service continuity can be supported in cloud based LTE systems. This section provides the answers to research sub-questions (1), (2), and (3). In particular, it presents the requirements that need to be satisfied by a service continuity solution, the possible service continuity solutions that could be implemented in cloud based LTE systems, and the architecture of cloud components used to support service continuity.

Chapter 3 discusses the integration options of the CCN (Content Centric Networking) concept into the cloud based LTE systems. This section provides the answer to research sub-question (4).

Chapter 4 describes the accomplished simulation experiments, including simulation environment and assumptions, simulation topology and parameters, performance metrics and experiment scenarios. This chapter provides the answer to research sub-question (5).

Chapter 5 describes the results of simulation experiments and provides evaluation of the simulation results. This chapter evaluates the service continuity solution in supporting mobility, content migration and VM (container) migration.

Chapter 6 provides the conclusions and the recommendations for future works.

Chapter 2

Service Continuity in Cloud Based LTE Systems

In this chapter, the requirements that need to be satisfied by a service continuity solution when it is applied in cloud based LTE systems are described. Moreover, an overview of service continuity solutions in mobile networks will be presented. Furthermore, the architecture of cloud components which is used to support service continuity will be explained.

2.1 Requirements of Service Continuity in Cloud Based LTE Systems

Service continuity is an important feature that needs to be supported in order to provide services to mobile users without interruption. A mobile user moving from one sub-network to another sub-network should not lose service continuity. A migration of IP sessions should not provide any impact on service continuity; thus the mobile user will be able to continue getting services seamlessly from the network.

In cloud based LTE systems, services are hosted on Virtual Machines (VMs) that can be moved and migrated across multiple networks to locations that are the best to deliver services to mobile users. The migration of VMs and/or service/functionality instances running on such VMs should happen without losing service continuity. Therefore a service continuity solution is able to support user mobility if it can support migration of services, which includes supports for:

- *IP address continuity*: when a user moves to another sub network, the application will not observe a change of IP address on the received IP packets.
- *Session continuity:* session continuity is a combination of IP address continuity and service context migration. Service context migration means when a user moves to a new location, the service context used by the function in previous location should be able to be migrated and used by the same function in the new location.
- *Content continuity:* the requested content can migrate and can be delivered from a location close to a mobile user. Content migration has to be supported in order to maintain content continuity. Content migration can be defined as a migration or duplication of content to a new location due to user mobility.
- *Storage continuity*: storage should be able to migrate to a new location which is close to a mobile user. Storage migration can be defined as a migration of storage to a new location or the use of new storage in a new location for storing content.

- *Function continuity:* the same function in a new location can be run using context used by the same function in the previous location. Function migration needs to be supported in order to maintain function continuity. Function migration can be defined as a migration of function to a new location or an execution of function in a new location.

2.2 Service Continuity Solutions in Mobile Networks

In this section, we introduce several existing mobility management solutions and verify what their ability in supporting service continuity is. These solutions are: Information-Centric Networking (ICN), Software-Defined Networking (SDN), Identifier-Locator Network Protocol (ILNP), Host Identity Protocol (HIP), Session Initiation Protocol (SIP), Stream Control Transmission Protocol (SCTP) with dynamic address reconfiguration extension, Proxy Mobile IPv6 (PMIPv6), and Mobile Content Distribution Networks (MCDN).

2.2.1 Information-Centric Networking

The information-centric networking (ICN) is an approach in internet architectures based on named data objects (NDOs). It changes the focal point of the network architecture from the "end host" to "information" (content or data) [13]. The ICN architectures leverage in-network storage for caching, multiparty communication through replication, and interaction models that decouple senders and receivers [14].

The NDO, such as web page, document, video, or other information, is independent of location, storage method, application program, and transportation method. A unique name for each NDO is required in ICN to identify objects independent from its location. Information about the source of an object is also useful to associate with the name. ICN has an Application Programming Interface (API) which is used as an interface for publishing or getting NDOs. By using the ICN API, a producer can publish NDOs to internet, and a consumer can get them from internet.

Figure 7 shows an ICN communication model for client side where a client can get the requested object from any network nodes holding a copy of the object; since ICN leverages in-network caching. ICN secures the object itself, independent of the node that delivers it, that could be untrusted.

B. Ahlgren et.al, [14] described some advantages of the ICN approach as follows:

- The ICN approach offers scalable and cost-efficient content distribution since it leverages in-network caching, so that requests for NDOs can be served by any network nodes holding a copy of requested NDOs.
- The ICN approach has persistent and unique naming of NDOs, and also a service model that decouples senders and receivers, so that it does not have a problem with name-object binding.

For instance, when an object is moved, the client still can get that object. It is different from current condition where most content URIs in current network are object locators that appoint to a web server serving requests from a client. In this case, when an object is moved, the site can be unreachable.

- The ICN approach has an interesting security model. It provides name-data integrity and origin verification of NDOs, independent of the immediate source. It enables ubiquitous caching with retained name-data integrity and authenticity.
- The ICN approach supports mobility (IP address continuity) and multihoming. A mobile client just need to send requests for NDOs to a new access and the requests will be served by a network node that might be different from the previous network node. A multi-homed client can similarly choose to send a request on any one, several, or all accesses.
- The ICN approach provides better reliability and better performance compared to current network that needs end-to-end connection management to origin servers since it leverages optimized hop-by-hop transport and in-network caching.

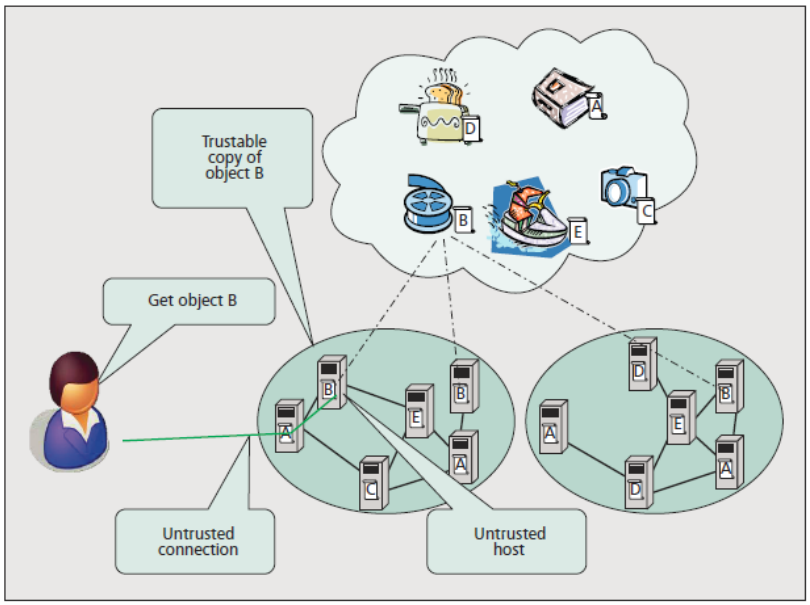

Figure 7: ICN communication model: client side, copied from [14]

Currently there are several ICN approaches that have been developed, such as Data-Oriented Network Architecture (DONA) [15], Content-Centric Networking (CCN) [16], Publish-Subscribe Internet Routing Paradigm (PSIRP) [17], Network of Information (NetInf) [18], Translating Relaying Internet Architecture integrating Active Directories (TRIAD) [19] and Service Centric Networking (SCN) [22]. The following subsections provide a brief overview of those approaches which is mostly based on [14].

2.2.1.1 Data-Oriented Network Architecture

Figure 8 illustrates how a requester can get the requested data from a source in Data-Oriented Network Architecture (DONA). DONA [15] uses the route-by-name paradigm for name resolution and relies on a new network entity called resolution handler (RH). Each domain has one RH that maintains a registration table that maps a name to both the next-hop RH and the distance to the copy. Any node authorized to serve data will register to its local RH. NDOs are published into the network by the sources. To get the requested data, a requester will send a FIND packet that will be routed by name toward the appropriate RH (steps 1–4). The request is routed by name in a hierarchical fashion. The RH resolution infrastructure will route the request and find a copy of the content closest to the requester. As a response, the requested data will be sent back by the source to the requester. The data can be sent to the requester either through a direct route (step 9) or through the reverse RH path (steps 5–8) by enabling caching. In DONA, a NDO name is possible to be registered before the NDO content is available. A wildcard registration can be performed by content providers in the RH to make queries to be directed to them without the need to register specific objects. There is expiry time in register commands, so that a renewal of registration is needed when the time is expired.

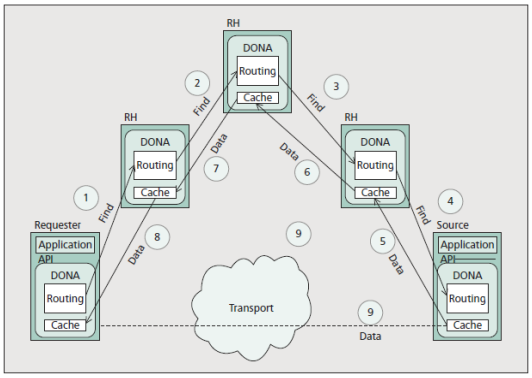

Figure 8: DONA overview, copied from [14]

DONA supports IP address continuity, content continuity, and storage continuity. In case of mobility, users can deregister from their previous location and re-register to their new location. Afterwards a mobile client just needs to send requests for NDOs to a new access; since DONA uses NDO names as addresses instead of host addresses, IP address continuity can be supported. DONA leverages in-network caching, so that the requests can be responded by any RH holding a cached copy of the requested NDOs; therefore content continuity and storage continuity can be supported. However, while IP address continuity, content continuity, and storage continuity are supported, DONA does not support session continuity and function continuity since it does not provide mechanisms to support service context migration and function migration.

2.2.1.2 Content-Centric Networking

Content-Centric Networking (CCN) uses routing protocols to distribute information about a location of NDO published on nodes. CCN uses public key cryptography to provide NDO security. There are several ways to establish trust in keys, such as by using a PKI-like certificate chain based on the naming hierarchy, or by using information provided by a friend.

There are two types of CCN packet, namely Interest, which contains a request for NDO, and Data, which contains a response for an Interest. Step 1 - 3 in Figure 9 shows how an Interest packet is forwarded to a source through CCN routers. A CCN router has a Forwarding Information Base (FIB) which contains information where Interest packets have to be forwarded. Moreover, a CCN router also has a Pending Interest Table (PIT) that keeps state for each outstanding Interest, so that when a router receives multiple Interests for the same NDO, Interest aggregation can be done and only the first Interest is forwarded to the source. Interests are mapped to network interface where corresponding Interests have been received from. The requested data will be routed back on the reverse Interest path (steps 4–6). Moreover, a CCN router also has a cache that can be used to cache NDOs received by the router, so that subsequent received Interests for the same object can be satisfied from that cache (steps 7–8).

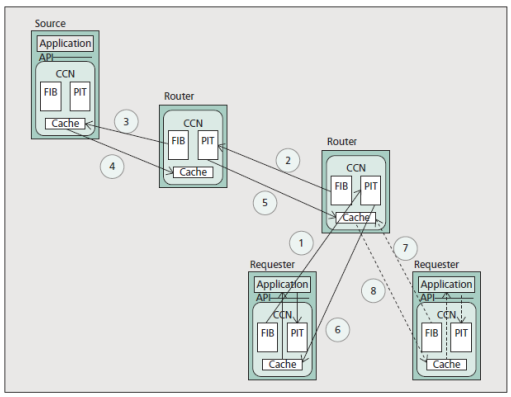

Figure 9: CCN Overview, copied from [14]

CCN supports IP address continuity, content continuity, and storage continuity. Mobile users moving from a sub-network to other sub-network can continue to issue NDOs. They just need to send requests for NDOs to a new access; IP address continuity can be supported since CCN uses NDO names as addresses instead of host addresses. The requests can be satisfied by a CCN router which is close to the user and holds a copy of requested NDOs, hence content continuity and storage continuity can be supported. However, while IP address continuity, content continuity, and storage continuity are supported, CCN does not support session continuity and function continuity since it does not provide mechanisms to support service context migration and function migration.

2.2.1.3 Publish-Subscribe Internet Routing Paradigm

Figure 10 shows the overview of Publish-Subscribe Internet Routing Paradigm (PSIRP). In PSIRP [17], an NDO source publishes NDOs to the network (step 1). The publication belongs to a particular named scope. A client/requester can subscribe to NDOs (step 2). A rendezvous system matches the publication and subscription (step 3). The rendezvous system is a policy-enforcement point and a mechanism to support freedom of choice for network end points. The scope identifier (SI) and the rendezvous identifier (RI) specified by the subscription request are used together to name the desired NDO. The identifiers are used as an input to a matching procedure in the rendezvous system. The output of this matching procedure is a forwarding identifier (FI) which will be sent to the NDO source. That FI is used by the NDO source to forward the requested data to the requester (steps 5–7). PSIRP routers use a Bloom filter contained in the FI to select the interfaces on which to forward an NDO. A Bloom filter is a simple space-efficient randomized data structure for representing a set in order to support membership queries [20].

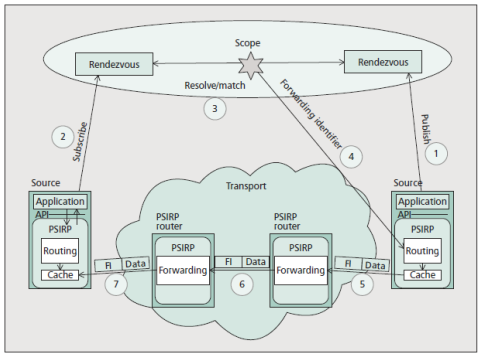

Figure 10: PSIRP overview, copied from [14]

PSIRP supports IP address continuity, content continuity, and storage continuity. In case of mobility, a user just needs to send requests for NDOs to a new access without need to keep an association to a specific source of NDOs. PSIRP uses NDO names as addresses instead of host addresses, so that IP address continuity can be supported. In PSIRP, NDOs can be cached in multiple caches within the scope of rendezvous point, so that the requests can be responded by any nodes holding a copy of requested NDOs; therefore content continuity and storage continuity can be supported. However, while IP address continuity, content continuity, and storage continuity are supported, PSIRP does not support session continuity and function continuity since it does not provide mechanisms to support service context migration and function migration.

2.2.1.4 Network of Information (NetInf)

Network of Information (NetInf) [18] has two models that can be used to retrieve NDOs. NDO sources can publish NDOs by either registering a name/locator binding with a name resolution service

(NRS), or announcing routing information in a routing protocol, depending on the model used in the local network. A NetInf node holding a copy of an NDO (including in-network caches and user terminals) can optionally register its copy with an NRS, thereby adding a new name/locator binding. The illustration of NetInf can be seen in the Figure 11. In case of an NRS is available, a requester can use the NRS to resolve an NDO name into a set of available locators (steps 1–2). Afterwards the requester can get a copy of the data from the NDO source (steps 3–4). Besides, the requester can also directly send a GET request with the NDO name to the potential node holding a copy of requested data. That request will be forwarded by router(s) using name based routing (steps 5–6). After the node holding a copy of data is reached, the data will be sent to the requester (steps 7–8).

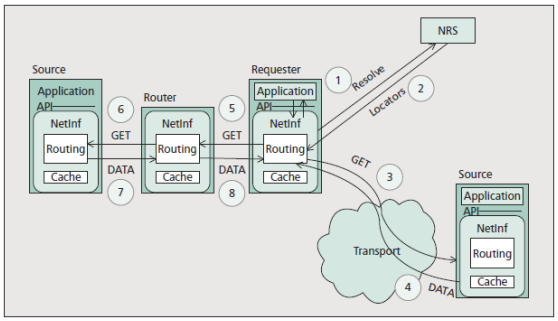

Figure 11: NetInf overview, copied from [14]

NetInf supports IP address continuity, content continuity, and storage continuity. In case of mobility, a mobile client just needs to send requests for NDOs to a new access; since NetInf uses NDO names as addresses instead of host addresses, IP address continuity can be supported. The requests can be responded by a node that might be different from the previous node. NDOs can be cached and can be delivered to users from the closest location; therefore content continuity and storage continuity can be supported. However, while IP address continuity, content continuity, and storage continuity are supported, NetInf does not support session continuity and function continuity since it does not provide mechanisms to support service context migration and function migration.

2.2.1.5 Translating Relaying Internet Architecture integrating Active Directories (TRIAD)

Translating Relaying Internet Architecture integrating Active Directories (TRIAD) defines an explicit content layer that provides scalable content routing, caching, content transformation and load balancing, integrating routing and transport connection setup [19]. TRIAD identifies endpoints by using names. In the content layer, Web Uniform Resource Locator (URL) is used as the format for content identification to provide compatibility with the World Wide Web. The content layer is implemented by content routers that can forward requests to content servers holding the content, content servers that provide content services, and content caches storing a copy of content.

TRIAD uses a name lookup and connection setup protocol, Directory Relay Protocol (DRP), at content layer to setup a transport connection. A client can send a lookup request to the directory system, then it will route the name request to a content server that stores the requested content (NDO). The location of NDO is returned to the client, and then the NDO is retrieved using standard HTTP/TCP. TRIAD uses Name-Based Routing Protocol (NBRP) [21] to perform routing by name. NBRP distributes name suffix reachability to content routers. Routing information is maintained locally with next hops and destinations specified using names and name suffixes. Name mapping information is distributed by NBRP to ensure availability, distribute the name lookup load, and provide faster name lookup response.

TRIAD supports IP address continuity, content continuity, and storage continuity. A host visiting a guest network will get a temporary guest name in that network that allows the host getting service from the network. A mobile host that moves to a new location (network) can continue requesting for NDOs even though its address may change; therefore IP address continuity can be supported. The mobile host just needs to acquire a guest name in the guest network and register its real name with its guest network and its guest name with its home network; then the transport connections will rebind based on the name identification. Furthermore, TRIAD architecture leverages in-network caching that allows requests for NDOs to be served by any network nodes storing a copy of requested NDOs in its cache; therefore content continuity and storage continuity can be supported. However, while IP address continuity, content continuity, and storage continuity are supported, TRIAD does not support session continuity and function continuity since it does not provide mechanisms to support service context migration and function migration.

2.2.1.6 Service Centric Networking (SCN)

Service-Centric Networking (SCN) [22] is a new ICN paradigm for the future internet, in which routing and forwarding are based on service identifiers. SCN is an extension of CCN which is designed by using an object oriented approach, in which content and service are considered as an object. In SCN, data or content not only can be retrieved but also can be processed before being delivered to users. Services and contents processed by services can reside at different locations in a network. Services typically are located in servers, while data or contents are located in data storages; therefore in order to determine the most appropriate server, SCN not only consider the distance between client and server, but also the distance between server and required data.

In SCN, services are represented as functions to be invoked by users. By using object oriented approach, both functions and data are integrated into objects. For invoking functions, methods are called among objects. In SCN, clients can request for both services and contents by using object names. There are three types of objects, namely pure content objects, pure service objects, and combined content and service objects (see Figure 12).

- *Pure content objects* represent data, such as images, video, files, etc. Pure content objects only support read methods which are the default methods to be called if there is no other given method.
- *Pure service objects* represent service functions which do not have association with particular data. A client can invoke these objects to process its individual data by specifying the content or location of the data in the service invocation as additional parameters.
- *Combined content and service objects* represent a combination of both services and content data. A client can send a request for a service by using the object name as an address. Afterwards that request will be routed to the object storing the data, and the data can be directly processed on the node hosting the object by using the given method.

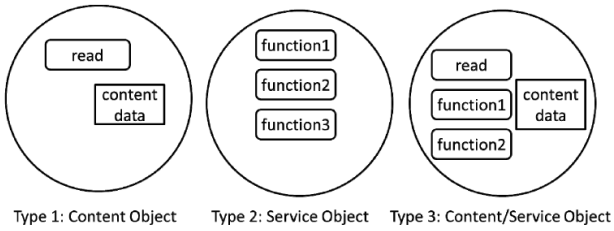

Figure 12: **SCN object types, copied from [22]**

In SCN, services can be invoked one after the other. SCN uses a routing header in which all objects to be visited by a service request are listed (see Figure 13) Objects listed in the routing header (Objectname1-N) can be visited in sequence or in parallel depending on the dependency of corresponding functions to each other. The list of parameters (Parameters1-N) is located in the data part of the request, and they will be used by methods to be called.

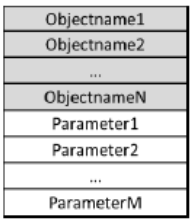

Figure 13: **Combining services using routing header, copied from [22]**

SCN supports stateful services that require both client and server to share some client-specific context. In stateful services, clients need to communicate to the same server (e.g. e-banking). SCN provides session_establishment method used to establish sessions between a client and a server to ensure that the client can communicate to the same server. The session_establishment method can be sent in an interest message to a service object before service requests and results are exchanged. After the session is established, the client's request can be forwarded to the server, and the server's response

can also be forwarded to the client. To support the session concept, the underlying CCN infrastructure is modified. SCN introduces session table as an addition to forwarding information base, content store, and pending interest table. The session table should contain entries for all Interests and Data messages that will be exchanged between a client and a server. An object name, a session identifier, and a session timeout should be included in each session table entry.

Since SCN is an extension of CCN, it also supports IP address continuity, content continuity, and storage continuity. In addition it also can support session continuity and function continuity. In case of mobility, when a user moves to a new location, the user just needs to send a request for a service by using an object name that identifies the service; IP address continuity can be supported since SCN uses object names as addresses instead of host addresses. Afterwards the request will be routed to the closest server supporting this service that might be different from the previous server. SCN uses automatic server selection in determining the most appropriate server used to handle requests (see Figure 14). CCN routers used in SCN can cache data resulted from the service call and then use these data to satisfy subsequent requests sent by mobile users; hence content continuity and storage continuity can be supported. In SCN, the service context used by the function in previous location can be migrated and used by the same function in the new location; therefore session continuity which is a combination of IP address continuity and service context migration can be supported. Furthermore, the same function in a new location can be run using context used by the same function in the previous location, so that function continuity can be supported.

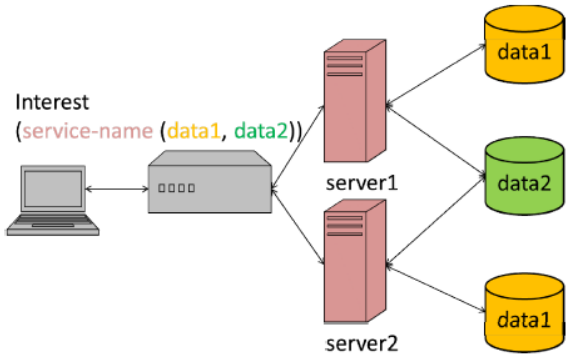

Figure 14: **Automatic Server Selection, copied from [22]**

2.2.2 Host Identity Protocol (HIP)

Host Identity Protocol (HIP) introduces a new layer between the network and transport layers (see Figure 15) which maps the host identifiers to network addresses and vice versa [23]. In the HIP architecture, the end-point names and locators, which are both represented by IP addresses in the current Internet architecture, are separated. IP addresses will act as locators, while the host identifiers take the role of end-point identifiers. The host identifier (HI) in HIP is the public key of an

asymmetric key-pair that can be used to verify signatures without access to certificates or a public-key infrastructure. The existence of Host Identity Layer ensures that transport layer connections are no more bound to IP addresses, so that if a location of hosts change, the connections do not have to be broken.

HIP uses an initial four-packet handshake mechanism, called Base Exchange (BE), to establish end-to-end connection and set up keying material for the communication. An IPSec Encapsulated Security Payload (ESP) and Security Association (SA) pair are constructed between the endpoints using a Diffie-Hellman authenticated key exchange during the BE [25]. The built-up ESP SAs are bound to HIs, and packets traveling in the network are identified and mapped to the correct SA using the Security Parameter Index (SPI) value in the IPSec header and the destination IP address in the IP header.

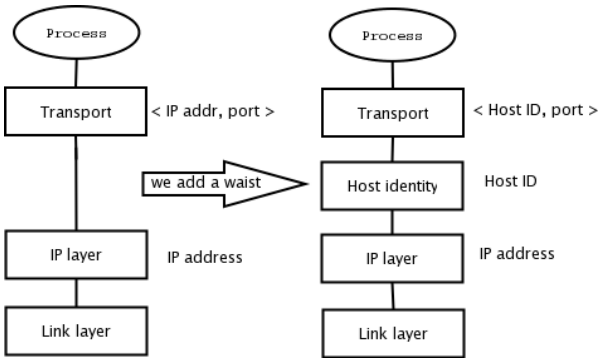

Figure 15: The Host Identity layer, copied from [24]

HIP does not support session continuity, content continuity, storage continuity, and function continuity; it only provides mechanisms to support IP address continuity as described below.

a) HIP Mobility with Single SA Pair

HIP has a parameter called LOCATOR used to allow a HIP node to update existing HIP associations if the node changes its network point of attachment by sending an UPDATE packet (see Figure 16) to its correspondent nodes (CNs). CNs that receive an UPDATE packet will update its HI - IP address mapping, so that the communication between moving node and its CNs can continue. The changes in IP layer are transparent to a HIP node since transport layer connections are bound to HIs. It makes HIP mobility management and multihoming provisioning can be handled easily.

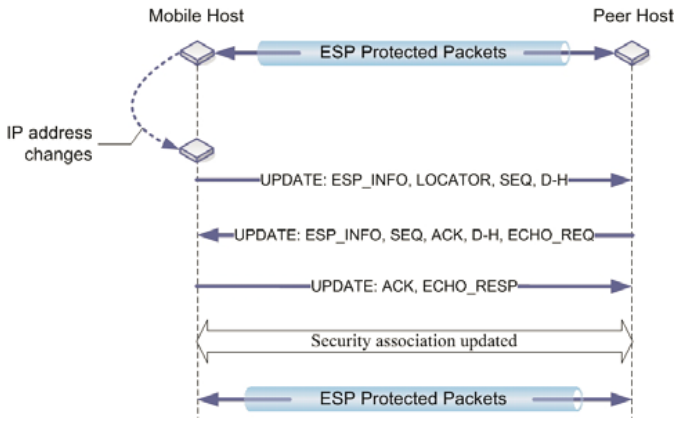

Figure 16: HIP mobility with single SA pair, copied from [26]

b) Mobility with Rendezvous Mechanism

In some complex cases, e.g. caused by frequent location updates or simultaneous mobility nodes, the simple end-to-end readdressing functionality and HIP architecture are not adequate. To overcome this problem, a new network entity called HIP Rendezvous Server (RVS) [27] was introduced. RVS is used to map HIs onto a set of IP addresses. A mobile node entering the network should register its IP address in a network directory known by all its potential CNs. The use of RVS as a second global name service, besides DNS, is the solution used for tracking the frequent address changes of mobile nodes [28].

Figure 17 shows HIP Rendezvous mechanism which is a mechanism used to locate mobile hosts based on their Host Identity Tag (HIT), a 128-bit representation for a Host Identity. When a mobile node (MN) enters a new network, the MN will update its entry at the RVS and reports the IP address of the RVS at the DNS. A CN will perform a DNS lookup for the IP address of the MN if the CN wants to communicate with the MN. Afterwards the DNS will inform the IP address of the MN's RVS to the CN. Then the CN sends the I1 packet to the RVS with the HIT of the MN to initiate the HIP connection. The RVS will add a FROM parameter, that represents the IP address of the CN, and forward the packet to the MN. After receiving the packet, the MN will send the R1 packet directly to the CN, and afterwards the BE will be accomplished in the regular way.

In [26] S. Novaczki et.al, described a Host Identity Protocol (HIP) extension called HIP-NEMO (NEtwork MObility), which is based on hierarchical topology, signalling delegation and connection tracking to enable secure and efficient network mobility support in the HIP layer. It also does not support session continuity, content continuity, storage continuity, and function continuity; it only provides mechanisms to support IP address continuity in case of network mobility. The Network Mobility (NEMO) Basic Support protocol [29] is used to enable mobile networks to attach to different points in the Internet and allow session continuity for every node in a mobile network when the network moves. HIP-NEMO was developed for scenarios where a complete network changes its point

of attachment instead of a single node. Their proposal supports multihoming and provides secure connectivity and reachability for every node and nested subnet in the moving network.

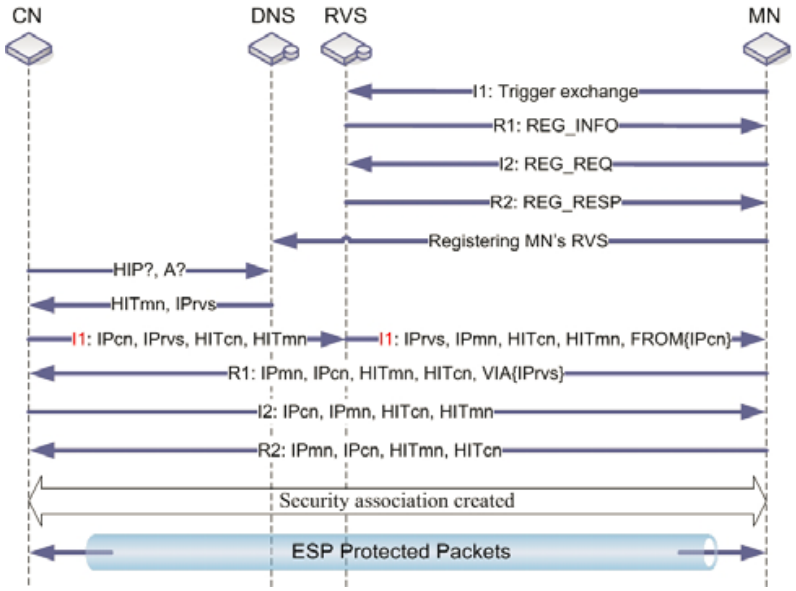

Figure 17: HIP Rendezvous mechanism, copied from [26]

All benefits offered by HIP are also inherited since the proposal is based on HIP. HIP-NEMO scales well with large and complex mobile networks. However, it introduces some packet overhead since inter mobile Rendezvous Server (mRVS) data flow needs to be tunneled. The root mRVS of nested environments will not be overloaded by huge signalling process management since one mRVS is responsible only for mobile network nodes (MNNs) that are connected directly to it. HIP-NEMO also provides a micromobility-like service for nested mobile networks. However their proposal also has some disadvantages. HIP-NEMO does not support completely seamless NEMO support for MNNs. It also can generate a large signalling overhead if many moving networks (MNets) change its point of attachment, since all Rendezvous Servers (RVSs) and Correspondent Nodes (CNs) of all MNNs have to be updated.

2.2.3 Identifier-Locator Network Protocol

Randall Atkinson et.al, [30] proposed the Identifier Locator Network Protocol (ILNP), a naming architecture which addresses issues of mobility, multi-homing, and end-to-end IP security. It evolves the naming in the internet by splitting an address into two different entities, an Identifier (I) used for end-to-end identity and a Locator (L) used for routing and forwarding packets. It provides an integrated solution to the issues mentioned above without changing the core routing architecture, while offering incremental deployability through backward compatibility with IPv6. A new network-layer protocol (ILNPv6) derived from IPv6 was introduced here.

The Locator, L, is used at the network layer, and the bits that hold the value of L are not visible above the network layer. The Identifier, I, is limited to a common, non-topological, end-to-end identifier used by transport-layer protocols. I is never used for routing, however the value of Identifier, I, is visible at the network-layer, so that a common identifier can be used by all transport-layer protocols. Changes in the value of L will not impact any upper-layer protocols since the transport-layer state is bound only to the Identifier, I, not to a network-layer address. With ILNP, new DNS record types for Locators and Identifiers are required.

A Locator names a single IP sub-network, not a specific host interface, while an Identifier names a (virtual or physical) node and is not tied to a specific host interface or network location. A host may have multiple Identifiers concurrently and may use multiple Identifiers simultaneously. However, any single transport-layer session must maintain the same value of I throughout its lifetime. An identifier is not required to be globally unique; however it must be unique within the scope of any particular Locator which it is used. A host may have several Locators at the same time. Packet delivery on the final hop uses the whole of an ILNP address, so that mechanisms such as ARP (IPv4) or Neighbor Discovery (IPv6) can be adapted for use easily.

ILNPv6 does not require significant changes to already deployed IPv6 backbone routers. ILNPv6 can be deployed incrementally since it is backwards compatible with IPv6. The use of Network Address Translation (NAT) will not impede the deployment of new services and protocols over ILNP. With ILNP, end-to-end security using IP Security (IPsec) can work with mobility, multi-homing, and NATs. However it still needs some improvement particularly in the areas of operational scalability, implementation considerations, and performance optimization.

ILNP does not support session continuity, content continuity, storage continuity, and function continuity; it only provides mechanisms to support IP address continuity as described below.

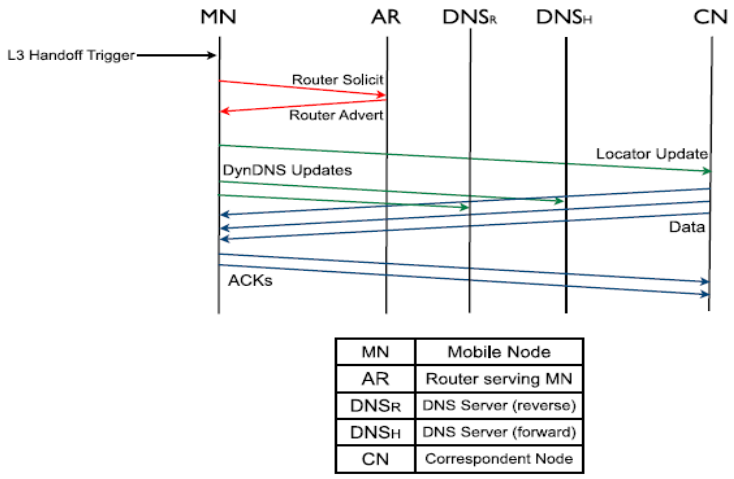

Figure 18: ILNPv6 handoff time-sequence diagram, copied from [30]

Host Mobility - A mobile node will update its Locator record in the DNS when the node moves to another subnetwork (see Figure 18). New sessions will be established directly to its current location, so that the Home Agent used in Mobile IP and Mobile IPv6 is not needed. Afterwards the mobile node sends authenticated (ICMP) Locator Update messages to all current correspondents informing of the node's new Locator. These LU messages are authenticated by the recipients, and then if the authentication succeeds, they will update their local Identifier/Locator cache. If a node does not respond, then the node's correspondents can make a DNS forward lookup on that node's domain name to learn its current set of Locators. In this way, the Foreign Agent used in Mobile IP is also not needed since all packets are transmitted directly from sender to receiver.

Network mobility - Network mobility is essentially handled by the same mechanism as host mobility. A node will change its value of L by discovering a suitable value locally from Router Advertisements (RA) whenever it moves from one subnetwork to another. An ILNP node may hold and use more than one value of L concurrently if it is multi-homed. With ILNP, a mobile network can be seen as a special case of multi-homing since values of L can be changed as site connectivity changes.

2.2.4 SCTP Protocol with Dynamic Address Reconfiguration Extension

T. Dreibholz et.al [31] proposed a new scheme for mobility management for IP-based networks that relies on the transport protocol SCTP (Stream Control Transmission Protocol) with the extension for dynamic address reconfiguration, and the reliable server pooling (RSerPool) protocol suite. The proposed solution is transparent for application, and it does not require changes in the network infrastructure.

SCTP [32] is a transport protocol that provides a more flexible data delivery by separating reliable transfer of messages between endpoints from the actual delivery to the user process. It supports multi-homed endpoints with more than one IP address, so that it provides improved network level fault tolerance. Heartbeat control chunks will be sent by SCTP endpoints regularly to all idle destination addresses of the peer endpoint in order to monitor the reachability of their peers.

The RSerPool protocol suite is used to provide server redundancy using server pools. The RSerPool architecture has three classes of elements, namely Pool Elements (PEs), Pool Users (PUs), and Name Servers. Pool Elements (PEs) are a pool of servers providing the same service within the pool. Pool Users (PUs) are clients being served by one PE. Name Servers (NSs) are nodes that provide a translation. The combination of RSerPool with the network fault tolerant transport protocol SCTP makes single points of failure of systems can be eliminated. It allows dynamic registration and deregistration of PEs. If a PE fails, the PU can fail-over to a different PE of the same pool.

A pool or a set of servers providing the same service is identified by a pool handle. A server can become a PE for a specific pool by registering itself at a name server (PE's home-NS) with the

corresponding pool handle. The protocol used between the PEs and the NSs is called the Aggregate Server Access Protocol (ASAP), while the protocol used by NSs within an operational scope, which is used to synchronize their name spaces, is the Endpoint Name Resolution Protocol (ENRP).

Mobile-SCTP with RSerPool does not support session continuity, content continuity, storage continuity, and function continuity; it only provides mechanisms to support IP address continuity. If one of two communicating nodes is mobile, Mobile-SCTP will be able to keep the association established between them. However if both nodes move and change their addresses simultaneously, the SCTP association may break. RSerPool provides a solution to overcome the problem by inserting a session layer between the transport layer (Mobile SCTP) and the application layer (see Figure 19). The session layer will ensure an establishment of a new transport association and trigger an application-specific failover procedure in case of the transport connection breaks caused by simultaneous mobility.

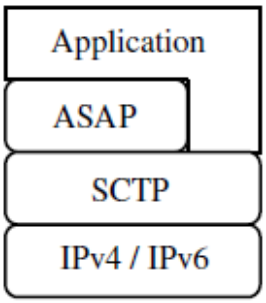

Figure 19: Protocol Stack for Mobile-SCTP with RSerPool, copied from [31]

Figure 20 shows an example of mobility scenario where simultaneous mobility occurs. We can see that in the beginning the called node registers under a unique pool handle. The caller node does an RSerPool name resolution and connects to the resolved transport address of the called node to establish the association. When simultaneous handover occurs, the called node re-registers itself with its new transport address under the same previous pool handle. The ENRP protocol ensures that all NSs of the operational scope get the updated pool data. Now the caller node can establish a new association and execute an application-specific failover procedure since the RSerPool name server can resolve the pool handle to the new transport address. Afterwards the communication between the caller node and the called node can be continued.

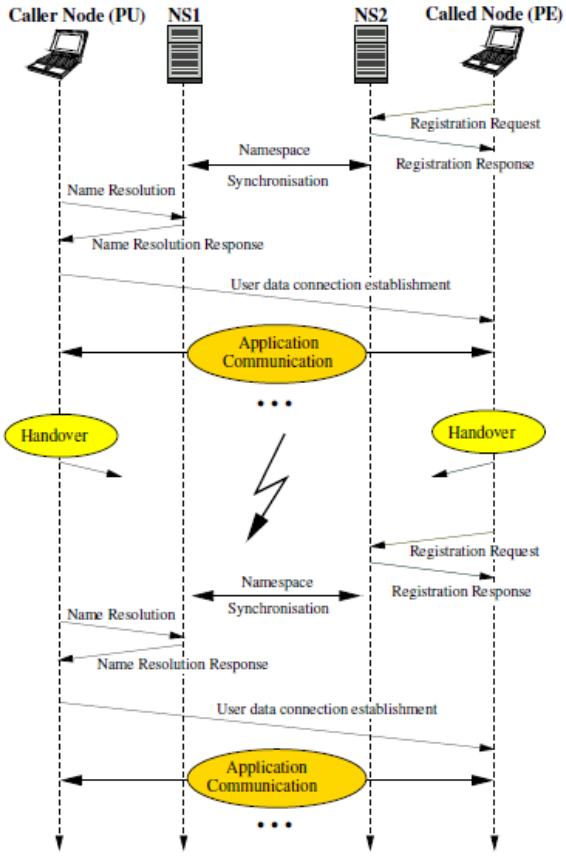

Figure 20: Mobility Example Scenario, copied from [31]

2.2.5 Session Initiation Protocol (SIP)

Session Initiation Protocol (SIP) [33] is an application-layer mobility management protocol that can be used to supports IP address continuity. SIP is transparent to the lower layer characteristics and can maintain the true end-to-end semantics of a connection. However the performance of SIP is limited by the underlying transport layer protocols, since SIP uses them to carry its signalling messages.

The main elements in SIP are user agents, proxy servers, and redirect servers. In SIP, a user is identified using SIP URI (Uniform Resource Identifier). SIP URI has a similar form to an email address such as user@userdomain, where user is the username and userdomain is the domain or numerical address. Several messages have been defined in SIP to set up sessions between user agents, such as INVITE, ACK, BYE, OPTIONS, CANCEL, and REGISTER.

SIP does not support session continuity (service context migration is not supported by SIP), content continuity, storage continuity, and function continuity; it only provides mechanisms that can be used to support IP address continuity during user mobility. SIP can establish a connection either during the start of a new session (pre-call mobility), when the mobile host (MH) has already moved to a different location, or in the middle of a session (mid-call mobility). In pre-call mobility, the MH will send a REGISTER message to re-register its new IP address with its redirect server in its home network, so

that the Home Agent (SIP proxy) will detect the new location of MH. The Correspondent Host (CH) gets information about the new location of the MH from the Home Agent and then sends an INVITE message to the MH (see Figure 21). In mid-call mobility, the MH will send an INVITE message about the terminal's new IP address and updated session parameters to the CH. After the INVITE message is received, the CN will send data to the new location where the MH is now located (see Figure 22).

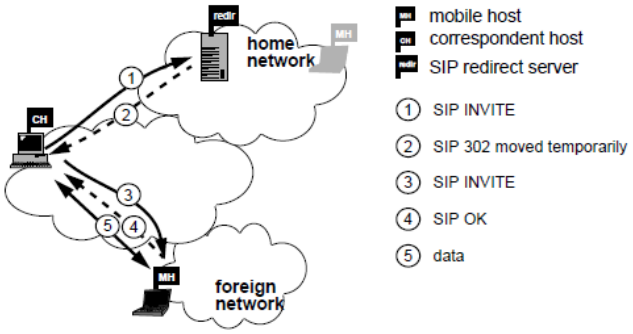

Figure 21: SIP-based pre-call location, copied from [33]

Furthermore, SIP also provides an ability of the home service provider either to maintain a control of services provided to the user in the visited network or to transfer the control to the visited network [34]. In case of mobility with control from home network, a control for user's sessions and services are maintained by the home network regardless whether the user is located in the home network or in a visited network. The user always registers to the home network. The user or mobile station (MS) sends a SIP REGISTER message to the Home Registrar (HR), then the HR send a query message to the home network AAA for requesting verification of MS credentials and rights (F2). If the HR gets a positive response from AAA (F3), it will send a 200 OK response to the MS. The complete registration process is shown in Figure 23.

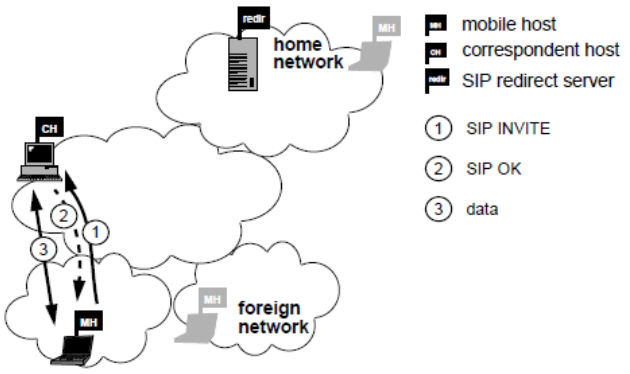

Figure 22: SIP-based hand-off in mid call, copied from [33]

In case of mobility with control from visited network, the control of mobile user's ongoing sessions is transferred to the visited network, and then the new sessions of the user will be controlled by the

visited network. The mobile user (MS) always registers to a local registrar in the visited network. The example of complete registration process with the visited network is shown in Figure 24. The MS sends a SIP REGISTER message to the Visited Registrar (VR), and then the VR send a query to the visited network AAA for requesting verification of MS credentials and rights (F2). If VR gets a positive response from the AAA (v), it will send the 200 OK response to the MS (F4).

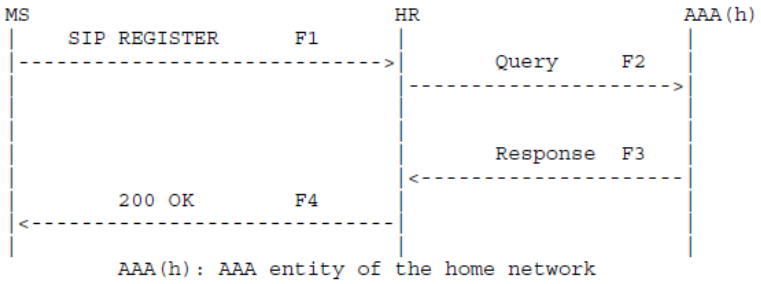

Figure 23: Complete registration with the home network [34]

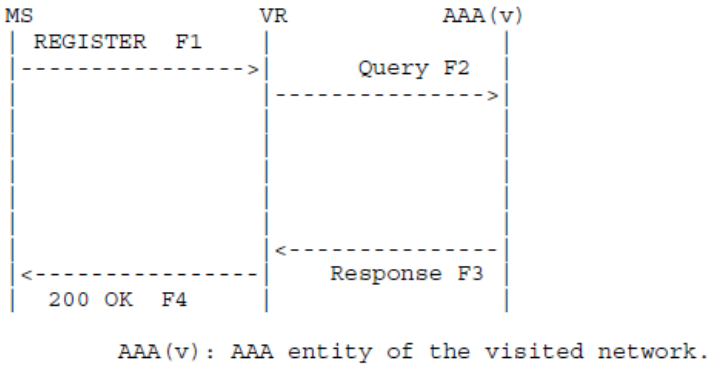

Figure 24: Complete registration process with the visited network [34]

2.2.6 Proxy Mobile IPv6 (PMIPv6)

Proxy Mobile IPv6 (PMIPv6) [35] is a network-based mobility management protocol based on the Mobile IPv6 concept [36], which is designed to support a mobile node without requiring participation of the mobile node in any IP mobility related signalling. There are two core functional entities in PMIPv6 used to track movements of mobile nodes and initiate the mobility signalling and set up the required routing state, namely Local Mobility Anchor (LMA) and Mobile Access Gateway (MAG). The LMA is the topological anchor point for the mobile node's home network prefixes used to maintain a reachability state of mobile node, while the MAG is the entity used to perform mobility management for a node attached to its access link. The MAG is responsible to detect the mobile

node's movement and to initiate binding registrations to the mobile node's LMA for updating the route to the mobile node's home address.

MPIPv6 does not support session continuity, content continuity, storage continuity, and function continuity; it only provides a mechanism to support IP address continuity. When an IPv6-enabled mobile node (MN) moves and wants to attach to a new MAG (nMAG), the MN will send a Router Solicitation (RS) message that contains its identity to the nMAG (see Figure 25). That information is then forwarded to the policy store, such as AAA-server, to get the MN's policy information. Afterwards the nMAG will send a Router Advertisement (RA), containing MN's home network prefix, the nMAG address, and other configuration parameters, to the MN. The MN will be always able to continue using its Home Address (MN-HoA) in the PMIPv6 domain since the MN always get the same home network prefix on the access link.

After RA is received by the MN, the nMAG will send a Proxy Binding Update (PBU) to the LMA in order to update the MN's current location. When the PBU Acknowledgement sent by the LMA is received by the nMAG, the nMAG will establish a tunnel to the LMA and add a default route over the tunnel, so that the LMA will be able to forward subsequent packets addressed to the MN through the nMAG. Meanwhile, the MAG on the previous link that detects the MN's detachment from the link will signal the LMA and remove the binding and routing state for that MN.

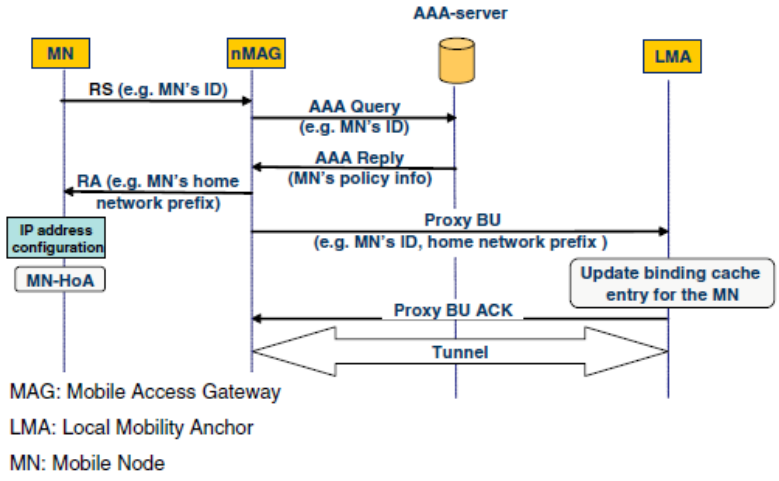

Figure 25: PMIPv6 handover operations, copied from [37]

2.2.7 Software Defined Networking (SDN)

Open Networking Foundation (ONF) defined Software Defined Networking (SDN) as an emerging network architecture where network control is decoupled from forwarding and is directly programmable [38]. The SDN architecture supports network virtualization since the underlying network infrastructure can be abstracted from the applications and network services. It provides a new

dynamic network architecture which changes traditional network platforms into rich service-delivery platforms.

In practice SDN refers more broadly to logically centralized software control [39]. The network appears to the applications and policy engines as a single logical switch since the network intelligence that maintains a global view of the network is logically centralized in software-based SDN controllers (see Figure 26). With SDN, network devices only need to accept instructions from the SDN controller without having to understand and process many protocol standards. It makes network operators can easily maintain the entire network from a single logical point. New services can be deployed, and network behavior can be changed in real time by leveraging the centralized intelligence of SDN controller. The centralized network state in the control layer provides simplicity and flexibility for network designers and network operators to configure and improve their network using automated SDN programs. Therefore SDN can enhance service continuity. .

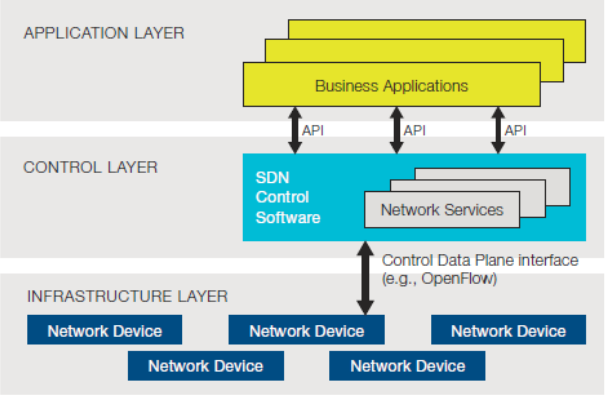

Figure 26: Software-Defined Network architecture, copied from [38]

According to its value proposition, SDN can be classified into three classes as follows [39]:

- *Flow Services SDN*
  It addresses the wealth of security and visibility applications by flow-level programmability.
- *Virtualization SDN*
  It provides virtual network connectivity for efficiency and agility.
- *Infrastructure SDN*
  It exposes network resources for continual optimization of resources and predictable handling of diverse traffic demands

OpenFlow [40] is the first standard interface designed specifically for SDN that provides an open protocol to program the flow table in different switches and routers. OpenFlow is defined between the control and forwarding layers of an SDN architecture. It allows the forwarding plane of network devices to be accessed and manipulated directly. OpenFlow-based SDN architecture provides extremely granular control since OpenFlow allows the network to be programmed on a per-flow basis.

Any OpenFlow-enabled network devices, such as switches, routers, and virtual switches, from any vendor can be controlled centrally by SDN control software. OpenFlow switch consists of three main elements, namely flow table, secure channel, and OpenFlow protocol (see Figure 27). A flow table contains information used by the switch to process the flow. Meanwhile, a secure channel is used to connect the switch to the remote controller in order to allow commands and packets to be sent between them using the OpenFlow protocol.

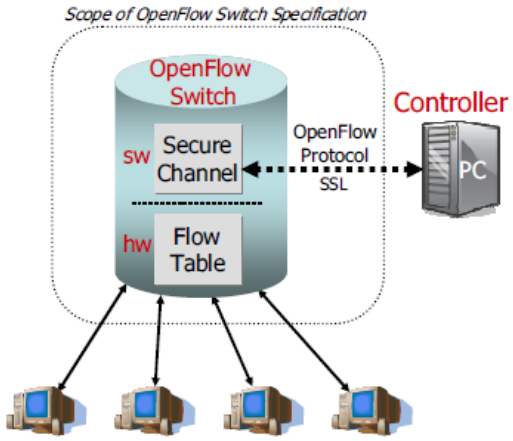

Figure 27: OpenFlow Switch, copied from [40]

By using OpenFlow-based SDN, complexity can be reduced since it provides a flexible network automation and management framework that can be used to build tools used for automating management tasks. With SDN, cloud-based applications can be managed by using intelligent orchestration and provisioning systems. New services and network capabilities can be delivered rapidly since network operators or network designers do not need to configure individual devices or wait for vendor releases. OpenFlow-based SDN provides more granular network control with the ability to apply comprehensive and wide-ranging policies at the session, user, device, and application levels, so that cloud operators can support multi-tenancy and maintain traffic isolation, security, and resource management between coexisting consumers that share the same network infrastructure. The SDN has SDN controllers that provide complete visibility and control over the network. It makes access control, traffic engineering, quality of service, security, and other policies can be enforced consistently across the wired and wireless network infrastructures. Centralized and automated management of network devices, uniform policy enforcement, and fewer configuration errors improve network reliability and security. It also offers a better user experience. The centralized network control and the availability of state information to higher level applications make an SDN infrastructure to better adapt to dynamic user needs.

Currently, there are many services which are distributed across multiple virtual machines (VMs). VMs should be able to migrate to locations which are closer to users in order to maintain Quality of Service (QoS) provided to users. The architecture of SDN and its support to network virtualization

enable it to support automated VM migration which is useful for enhancing service continuity and optimizing server workloads. Since a VM can have an IP address, service context, content, storage, and function, the SDN's support to VM migration means that SDN can support IP address continuity, session continuity, content continuity, storage continuity, and function continuity.

2.2.8 Mobile Content Distribution Networks

A mobile content delivery network (Mobile CDN) is a system of distributed servers that cooperate transparently to deliver content to end users based on geographic location of the users on any type of mobile network. This service is very useful to optimize and speed up delivery of content of websites with high traffic since it can deliver content from the closest location to the users with high availability and high performance. The nearest server will be responsible to respond a user's request. The CDN will copy popular content to a network of servers dispersed at geographically different locations. When there is a request from a user for content of a webpage which is part of a content delivery network, the CDN will redirect the request to a server in the CDN which is close to the user, and deliver the cached content.

The CDN serving point can terminate TCP connection between a mobile user and an original content provider and redirect the traffic to its cache. The CDN serving point may also maintain connection to the original content provider in order to retrieve non-cached content while delivering content to the user from the cache engine. In current deployment, cache engine is usually placed outside of an operator's core network (e.g. close to PDN-GW or GGSN). Furthermore there is also another solution such as Content Aware Edge (CAE) [51], which leverages in offloading technique in order to enable short delivery paths and low delivery costs.

Traffic offload technology, such as distributed mobility gateways (L-GW) can be used to offload traffic from the centralized mobility anchors. Figure 28 shows how L-GW and CDN serving point are used to deliver content from the edge of mobile operator network. An L-GW, that can provide direct routes to access local services, is selected by a mobile device based on its location (I). CDN caches and L-GW need to be localized in order to provide short paths in delivering content to a mobile user. When a mobile user is moving to a new location (2), the delivery path could become sub-optimal, depending on mobile device's mobility pattern and the mobile operator's transport network topology.

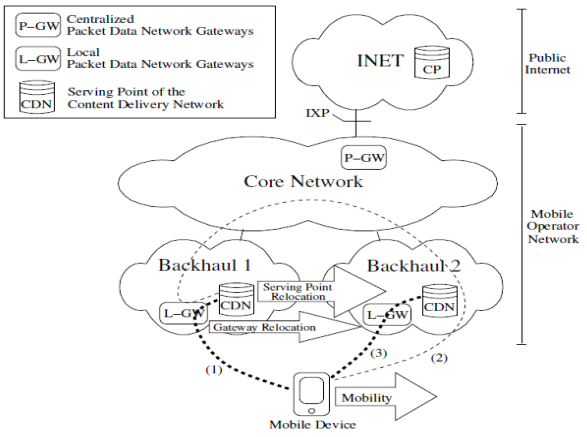

Figure 28: L-GW and CDN Serving Point relocations, copied from [52]

There are some requirements that need to be satisfied in order to relocate CDN serving point as depicted in the Figure 29. To enable runtime relocation CDN serving point, IP address continuity and session continuity during L-GW relocation, as well as transport endpoint migration, such as socket migration, and migration of a user's application context (e.g. state between previous and a new serving point) have to be supported.

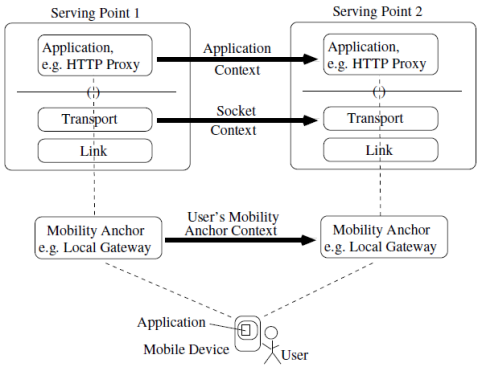

Figure 29: Requirements for runtime relocation of CDN serving point, copied from [52]

2.2.9 Conclusion

Cloud based LTE systems require a service continuity solution that can support the seamless migration of content and services by providing the support of IP address continuity, session continuity, content continuity, storage continuity, and function continuity.

According to their ability in supporting IP address continuity, session continuity, content continuity, storage continuity, and function continuity, the solutions have been described above can be summarized as shown on the Table 1.

Table 1: List of solutions and its ability in supporting service continuity

Solutions / Approaches	IP Address Continuity	Session Continuity	Content Continuity	Storage Continuity	Function Continuity
ICN	supported	supported	supported	supported	supported
ILNP	supported	not supported	not supported	not supported	not supported
HIP	supported	not supported	not supported	not supported	not supported
SCTP	supported	not supported	not supported	not supported	not supported
SIP	supported	not supported	not supported	not supported	not supported
PMIPv6	supported	not supported	not supported	not supported	not supported
SDN	supported	supported	supported	supported	supported
Mobile CDN	supported	supported	supported	supported	not supported

From the table above, we can see that only the ICN and SDN solutions can satisfy all requirements of service continuity. Therefore in this assignment we consider ICN/CCN as one of service continuity solutions in cloud based LTE systems. It is important to be noticed that the function continuity support proposed by SCN will also be considered. In addition to that, also the mobile CDN solution will be considered, since it supports IP address continuity, session continuity, content continuity, and storage continuity.

The combination of CCN, mobile CDN, and SDN approaches can be a good solution for cloud based LTE systems in order to enhance service continuity. The CCN solution offers reliable, scalable and cost-efficient content distribution since it leverages in-network caching. CCN routers can cache data resulted from the service call and then use these data to satisfy subsequent requests sent by mobile users. Similar to CCN approach, mobile CDN solution also offers reliable, scalable, and efficient content distribution since it can also cache content frequently requested by users. On the other hand, the SDN architecture based on OpenFlow provides a dynamic network architecture that fosters network virtualization and can improve network manageability, scalability, and agility. The centralized network state in the control layer provides simplicity and flexibility for network designers and network operators to configure and improve their network using automated SDN programs. However, significant changes on the network infrastructure need to be supported, i.e., use of Openflow switches instead of typical IP routers, when applying a SDN technology such as OpenFlow.

2.3 Architecture of Cloud Components to Support Service Continuity

Mobile Cloud Networking (MCN) project [53] has designed the MCN service architecture that can support end-to-end provision of mobile networks to enterprise end users, from the access part, core network to the service platforms.

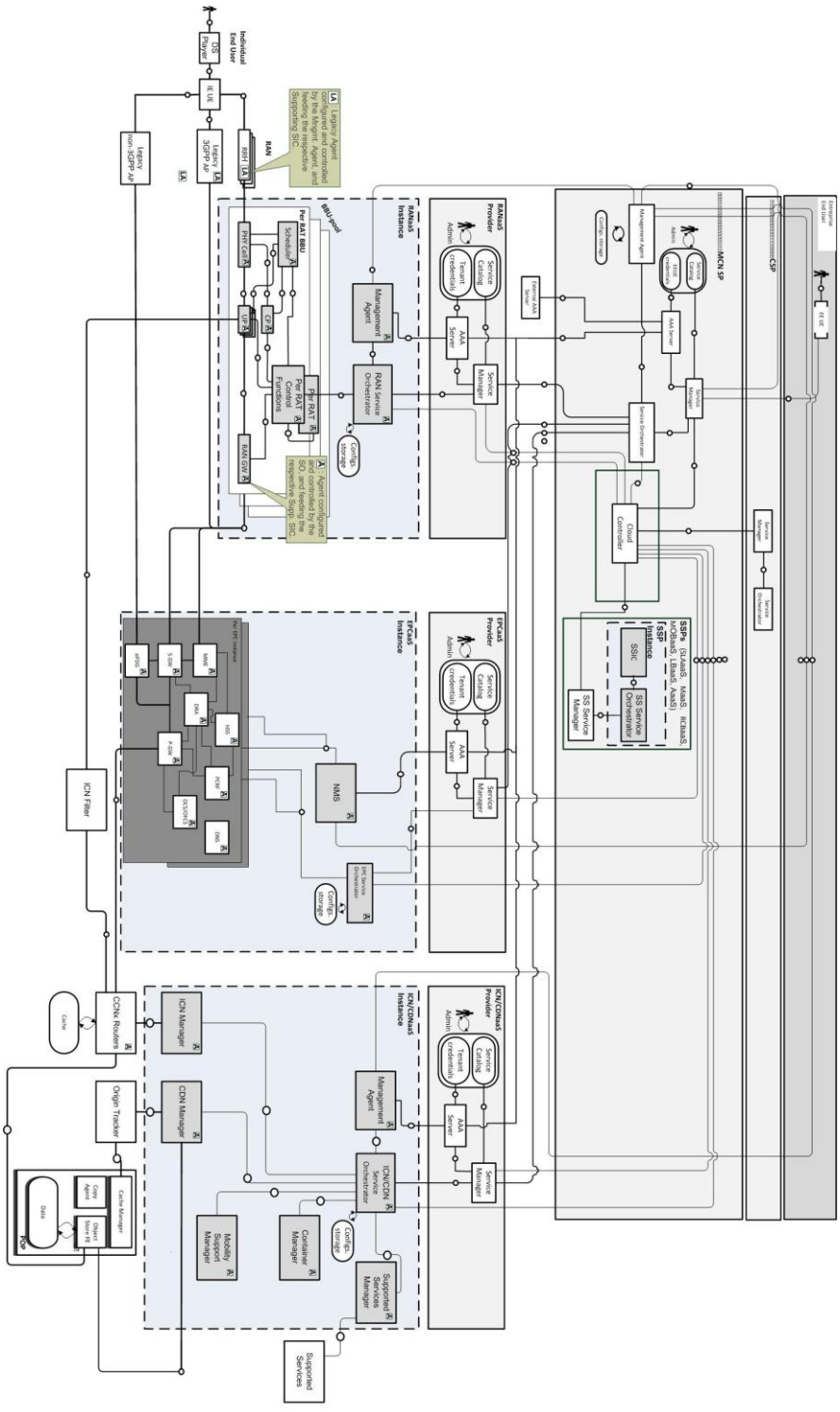

Figure 30: Architecture of the combined MCN service composed of RANaaS, EPCaaS and ICN/CDNaaS, copied from [54]

Figure 30 depicts a combined MCN service that is created using the composition of three MCN services. The description of this combined MCN service is provided in [54]. The access part of MCN services is provided through the Radio Access Network as a Service (RANaaS), see [55], that can

provide a service (based on demand) of Radio Access Network (RAN), both for 3GPP and non-3GPP access based, to Enterprise End Users (EEUs). On the core network, the Evolved Packet Core as a Service (EPCaaS), see [56], is used to provide a service (based on demand) of Evolved Packet Core (EPC) to EEUs. Furthermore, the MCN service which is used to support service continuity in cloud based LTE systems is Information Centric Networking/Content Delivery Network as a Service (ICN/CDNaaS), see [57]. ICN/CDNaaS is a service that integrates the features provided by Content Delivery Networks (CDNs) and Information Centric Networking (ICN) in a single service. Figure 30 shows how ICN/CDNaaS is used by RANaaS, EPCaaS and ICN/CDNaaS to support service continuity in cloud based LTE systems.

The detail of ICN/CDNaaS is depicted in Figure 31. There are seven components within the ICN/CDN instance, namely ICN Manager, CDN Manager, Service Orchestrator, Container Manager, Mobility Support Manager, Supported Services Manager and Management Agent. The main functions of those components are as the following ones.

- ICN Manager
  - creates and manages ICN layer based on current requirements
  - manages and allocates resources using the interface with the Service Orchestrator
  - ensures a good ICN topology to achieve better results in terms of costs and QoS/QoE
  - uses prices, energy and availability of physical resources to better optimize resources allocation
- CDN Manager
  - responsible for the management of CDN related components, including capabilities allowing the EEU to manage policies such as replication strategy, Point of Presence (POP) locations, etc
- Service Orchestrator
  - takes decisions (based on monitoring/policies) about the usage of virtual resources and acts towards the Cloud Controller to execute them
  - acts as a gateway for the connection with most Supporting Services
- Container Manager
  - deals with creation, instantiation and configuration of VMs
  - asks for VMs relocation and optimization according to the current needs
  - triggers VMs resources cleanup
- Mobility Support Manager
  - deal with the mobility of the user in terms of services and resources allocation/relocation
  - use mobility prediction to avoid reactive mechanisms and use proactive strategies

- Supported Service Manager
  - responsible for the management of services supported by ICN/CDN (when ICN/CDN is used as Support Service)
- Management Agent
  - provides an interface to the Enterprise End User (EEU) for managing the deployed instance

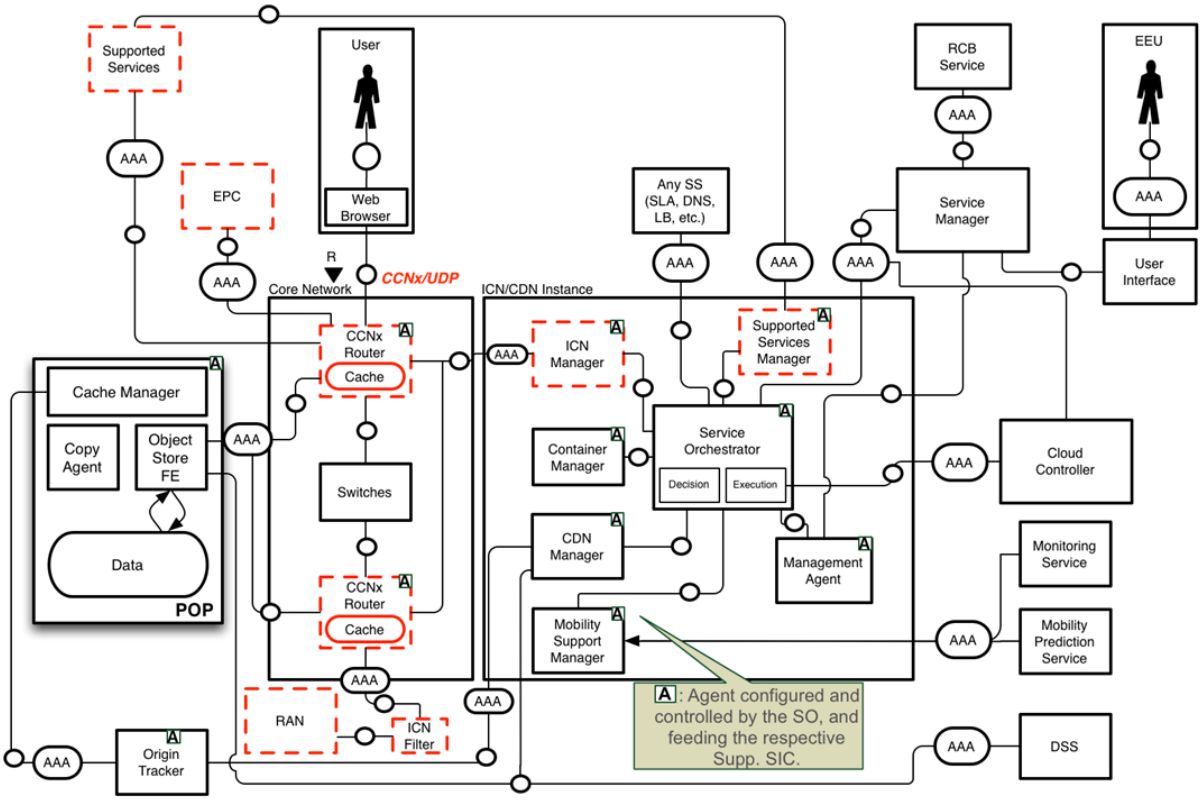

Figure 31: ICN/CDNaaS Architecture Diagram, copied from [54]

There are several other components outside the ICN/CDN instance that can also support service continuity in cloud based LTE systems, such as Cloud Controller and Mobility and Bandwidth Availability prediction as a Service (MOBaaS), CCNx Router and ICN Filter. The main functions of those components are as follows:

- Cloud Controller
  - resource provisioning, coordinating with Service Orchestrator
  - resource deployment
  - scaling on demand
  - management of virtual resources

- MOBaaS
  - gets mobility prediction information (predicts the location of user in future moments in time and the bandwidth that will be generated by these users in future moments in time)
- CCNx Router
  - router that implements all the ICN functionalities using the CCNx protocol
- ICN Filter
  - translates HTTP(S) based traffic to CCNx traffic and vice versa
  - implements any 3GPP related features that are usually implemented by the P-GW and used to assist the operator with capabilities like lawful interception, charging, etc
  - filters CCNx related traffic from other types of traffic

Chapter 3

CCN Integration in Cloud Based LTE Systems

In this chapter, the basic concept of CCN is introduced. Moreover, several options on how to integrate the CCN concept into LTE systems and the proposed solutions used to support content and VM migrations are discussed.

3.1 CCN Concept

The transport protocol used in CCN is called CCNx. There are two types of CCNx messages, namely the Interest message, which contains a request for NDO, and the Data message, which contains a response for an Interest.

The CCNx protocol works based on three main data structures as follows:

- *Content Store (CS):* CS is a buffer memory used for data retrieval by prefix match lookup on names. A replacement policy such as Least Recently Used (LRU) or Least Frequently Used (LFU) can be implemented by the CS in order to maintain the storage.
- *Forwarding Information Base (FIB):* FIB contains list of information to where the Interest messages should be forwarded. Each entry in the FIB may points to multiple interfaces, so that interest messages can be forwarded not only to a single interface.
- *Pending Interest Table (PIT):* PIT is used to keep track of interest messages forwarded upstream. It contains information of sources of unsatisfied interests. Each entry in the PIT may points to multiple sources. Entries in the PIT will not be held indefinitely since there is timeout for each entry.

3.1.1 Interest Packet

The format of Interest packet defined in [16] is depicted in the Figure 32. The Interest packet consists of three main elements, namely *Content Name*, *Selector*, and *Nonce*. *Content Name* contains information about the name of requested content. *Selector* contains information about selection of Data (Content Object) which is used to select a Data packet that will be delivered to a requester. While *Nonce* is used to detect and prevent duplicates received over different paths or interfaces.

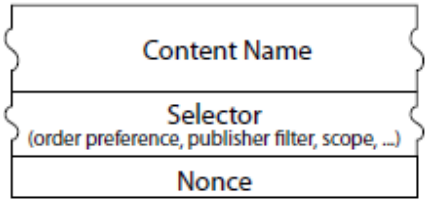

Figure 32: Interest packet, copied from [16]

3.1.2 Data Packet

The format of Data packet defined in [16] is depicted in the Figure 33**.** The Data packet consists of four main elements, namely *Content Name*, *Signature*, *Signed Info*, and *Data*. *Content Name* contains information about the name of content (Data). *Signature* is used for authentication. *Signed Info* contains some information such as *publisher ID, key locator* (used to find the key to verify this content), and *stale time* (used to specify how long the content is considered valid).

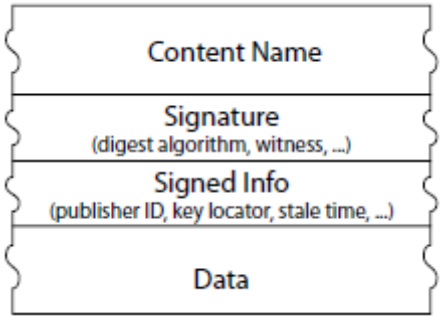

Figure 33: Data packet, copied from [16]

3.1.3 Processing of Interest and Data Messages

An Interest message received by CCN node will be processed by checking CS, PIT, and FIB tables sequentially:

- Lookup process on CS. If a match is found, this Data will be sent to the requester through the arrival interface of the Interest message, and the Interest message will be discarded. If no matching Data is found, the PIT table will be checked.
- Lookup process on PIT. If a match is found (it means that the same Interest has already been sent upstream), the arrival face of this Interest will be added into the list of requesting interfaces in the PIT entry, and then the Interest will be discarded. If no match is found, the FIB table will be checked.
- Lookup process on FIB. If a match is found, a new PIT entry for this Interest will be created by identifying the arrival interface of the Interest. Afterwards the Interest will be sent

upstream based on the strategy rules have been specified (Interest might be sent to one or more of the outbound interfaces). If no match is found, the Interest might be held for a short time before being discarded.

A data message received by CCN node will be processed by checking CS and PIT tables sequentially:

- Lookup process on CS. If a match is found, the Data will be discarded since it will be recognized as a duplicate Data. If no match is found, the PIT table will be checked.
- Lookup process on PIT. If a match is found, the Data will be sent downstream to all arrival interfaces of the Interests represented in the PIT table. A CCN node can apply Data verification and some policy restrictions or cache the Data before sending it downstream. If no match is found in the PIT table, the Data will be recognized as unsolicited Data and will be discarded. However, the CCN node might store this Data in case it is subsequently requested.

3.1.4 Routing Scheme

Figure 34 shows a basic routing scheme in CCN. All of clients, CCN routers, and source implement CCNx protocol and maintain the CS, FIB, and PIT. The CCNx protocol operates using the following steps:

- Step 1: Source advertises the data that can be provided to clients by doing a Register operation to the local CCN core.
- Step 2: Afterwards the FIB entry for the new registered prefix pointing to the repository application's interface is created. The announcement agent reads the registered prefix table on the local node (source) and advertises (via a CCN broadcast) the registered prefix
- Step 3: CCN router 1 that receives the prefix advertisement (step 2) from the source will update its local FIB by installing a new entry for this prefix pointing to the interface where it heard the advertisement. Afterwards CCN router 1 forwards this advertisement to CCN router 2. After receiving the advertisement, CCN router 2 will update its local FIB by installing a new entry for this prefix that points to the arrival interface of advertisement.
- Step 4: Client 1 establishes a connection to its access router (CCN router 2). After the connection is established, client 1 can send Interests via CCN router 2 (the local FIB of client 1 is set so that the interface to CCN router 2 will be used to send all Interest messages).
- Step 5: Client 1 sends an Interest to CCN router 2. A new entry for this Interest pointing to the application's interface is installed in its local PIT. When CCN router 2 receives the Interest from client 1, a look-up will be performed on its CS, PIT, and FIB sequentially. In this case, the matching information is only found on its FIB that informs that the Interest message has to be forwarded to the interface pointing to CCN router 1. After the Interest message is

forwarded to CCN router 1, CCN router 2 will install a new entry for this Interest that points to the arrival interface of Interest in its PIT.

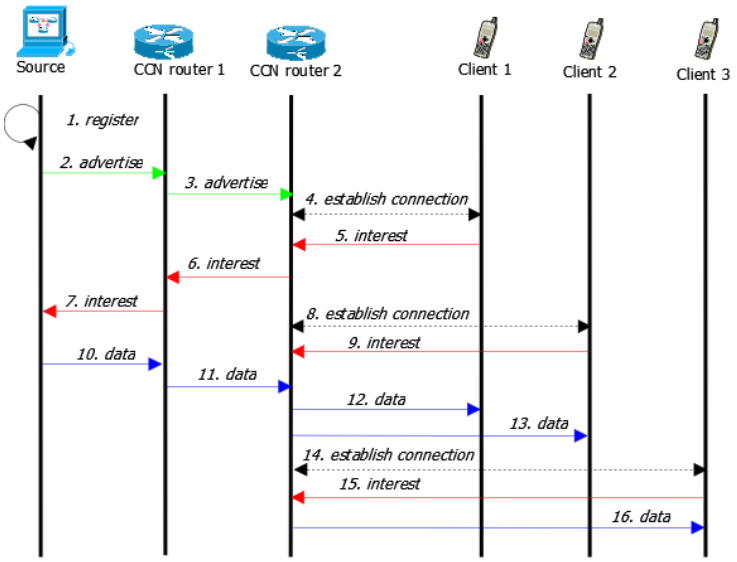

Figure 34: Routing scheme in CCN

- Step 6: When CCN router 1 receives the Interest message from CCN router 2, it will follow the same procedure performed by CCN router 2. After doing a look-up on its CS and PIT, and there is no matching information found, it checks its FIB. The FIB informs that the Interest has to be forwarded to the interface pointing to the source.
- Step 7: After forwarding the Interest to the source, the CCN router 1 will update its PIT by installing a new entry for this Interest that points to the arrival interface of Interest.
- Step 8: Client 2 establishes a connection to CCN router 2
- Step 9: Client 2 sends an Interest message to request the same content/data requested by client 1 to CCN router 2. After the connection is established, client 2 can send Interest messages via CCN router 2 (the local FIB of client 2 is set so that the interface to CCN router 2 will be used to send all Interest messages). When client 2 sends an Interest message to CCN router 2, a new entry for this Interest message that points to the application's interface is installed on its local PIT. When CCN router 2 receives this Interest message, it will check its CS first. However in this case it does not find the matching content/data since when the Interest message arrives on CCN router 2, it has not received the requested Data message yet sent by the source. Afterwards, CCN router 2 looks-up on its PIT and finds that the same Interest message has been sent previously. Since there is an exact-match PIT entry, the interface where the Interest arrived will be added to the PIT entry's interface list and the Interest will be discarded.

- Step 10: When the source receives the Interest sent by CCN router 1, it will look-up on its CS. When the matching content is found, it sends the content/Data as the response for the Interest to the arrival interface of Interest.
- Step 11 & 12: Afterwards, the Data message will be sent to the client 1 following the reverse path that the corresponding Interest message followed. After receiving the Data message from the source in step 10, CCN router 1 checks its CS to make sure that this content/data has not been received before. If the matching content/data is found on its CS, that Data message will be discarded. However, if there is no matching content/data, it will check its PIT to know where the Data message needs to be forwarded. In this case, the matching information is found on its PIT and then the Data message is forwarded to CCN router 2 (step 11). Before the Data message is forwarded, its content/data will be cached by CCN router 1. After the Data message is forwarded to CCN router 2, the PIT entry that corresponds to this Data message is deleted. CCN router 2 will follow the same procedure when it received the Data message from CCN router 1 (step 11). In particular, CCN router 2 checks its CS to know whether the Data message has been received before. Since there is no matching content/data in the CS, then it checks its PIT to know where the Data message has to be forwarded. The PIT of CCN router 2 informs that the CCN router 2 that the Data message needs to be forwarded to the interface pointing to client 1. Before the Data message is forwarded, the content/data is cached by CCN router 2. After receiving the requested data, client 1 stores the content/data to be used by the client 1's application into its CS. Afterwards, the entry on its local PIT that corresponds to this Data is deleted.
- Step 13: When the requested content/data is received by CCN router 2 , the Data message that will contain this content/data will be sent not only to client 1 (step 12) but also to client 2 (step 13), since the interface that points to client 2 has been added to the PIT entry that corresponds to this Data message. After receiving the requested content/data, client 2 will store the data to be used by the client 2's application into its CS. Afterwards, the entry on its local PIT that corresponds to this content/data will be deleted.
- Step 14: Client 3 establishes a connection to CCN router 2. After the connection is established, client 3 can send Interest messages via CCN router 2 (the local FIB of client 3 is set so that the interface to CCN router 2 will be used to send all Interests).
- Step 15: Client 3 sends an Interest to CCN router 2 and a new entry for this interest that points to application's interface is installed on its local PIT.
- Step 16: When CCN router 2 receives the Interest sent by client 3, it checks its CS and finds that there is a Data message that matches the Interest message. Afterwards, the Data message is sent out through the interface that the Interest message arrived on (the interface to client 3) and the corresponding Interest message is discarded. After receiving the requested

message/data, client 3 will store the data to be used by the client 3's application into its CS, and the entry on its local PIT that corresponds to this content/data is deleted.

3.1.5 Seamless Content Delivery Using FMA

R.Haw and C.S. Hong in "A Seamless Content Delivery Scheme for Flow Mobility in Content Centrix Network" [58] proposed a solution to solve problems caused by CCN routing scheme and client mobility by using the Flow Mapping Agent (FMA). In this solution, the client is assumed to have two interfaces that can be used to connect to different CCN routers. As it is illustrated in the Figure 35, FMA supports client mobility and seamless content delivery as follows:

- Step 1: Client establishes a connection to CCN router 2
- Step 2: CCN router 2 sends client information containing node ID and interface ID to FMA . FMA will register that information into the Flow Mapping Table (FMT). Figure 36 shows the FMT.
- Step 3: Now the client can send Interest messages via CCN router 2 (the local FIB of client is set so that the interface to CCN router 2 will be used to send all Interest messages). When the client sends an Interest to CCN router 2, a new entry for this Interest that points to the application's interface is installed in its local PIT. When CCN router 2 receives the Interest message from the client, a look-up is performed on its CS, PIT, and FIB sequentially. In this case, the matching information is only found in its FIB that informs that the Interest message has to be forwarded to the interface pointing to CCN router 1. After the Interest is forwarded to CCN router 1, CCN router 2 installs a new entry for this Interest that points to the arrival interface of Interest in its PIT.
- Step 4: CCN router 1 receives the Interest message from CCN router 2 (step 4) and follows the same procedure performed by CCN router 2. After doing look-up on its CS and PIT, and no matching information is found, it checks its local FIB. The FIB informs that the Interest has to be forwarded to the interface pointing to the source.
- Step 5: After forwarding the Interest to the source, CCN router 1 updates its PIT by installing a new entry for this Interest that points to the arrival interface of Interest.
- Step 6: When the source receives the Interest sent by CCN router 1 (in step 5), it looks-up on its CS. When the matching content is found, it sends the Data message as the response for the Interest through the arrival interface of Interest. The Data is sent to the client following the reverse path of the corresponding Interest. When the CCN router 1 receives the Data message, it checks its CS to make sure that this Data has not been received before. If a matching of content/data is found on its CS, that Data message will be discarded.

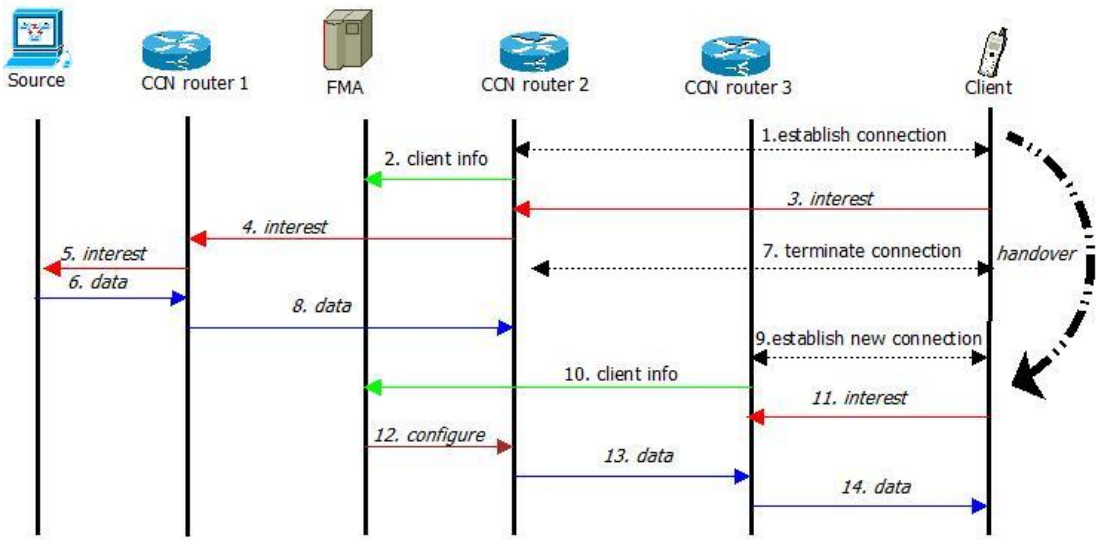

Figure 35: Seamless content delivery in CCN

Content	Router Address	Node Information
Router ID		
. . .		

Figure 36: Flow Mapping Table

- Step 7: Before the requested Data is received by CCN router 2, the client moves to another location and start doing handover mechanism by establishing a new connection to the CCN router 3 using another interface.
- Step 8: CCN router 2 receives the Interest.
- Step 9: Client establishes a connection to CCN router 3
- Step 10: CCN router 3 sends client information containing node ID and interface ID to FMA, and then FMA registers that information into the Flow Mapping Table (FMT). After the registration of client, FMA checks its FMT using Node ID.
- Step 11: Interest messages will be sent by client to the interface connected to CCN router 3.
- Step 12: Since the Node ID of the client exists in the FMT, FMA considers that the client has moved to CCN router 3. Afterwards, FMA sends an Interest message to CCN router 2 to deliver Data message to CCN router 3.
- Step 13: When the CCN router 2 receives this Interest, it will update its PIT so that the requested Data message will be forwarded to CCN router 3. After the handover process is completed, the client can receive the Data message from CCN router 3 via the interface connected to CCN router 3. Since the client is not connected to CCN router 2 anymore, the client will update its local FIB by changing the interface used to send Interest messages.

- Step14: When the CCN router 3 receives this Data message and after that the client sends an Interest message for the content/data has not been received in the previous location, the CCN router 3 can satisfy the Interest message by delivering the requested content/data to the client. CCN router 3 will cache the content/data before forwarding it to the client. After receiving the requested Data message, the client stores the Data to be used by the client's application into its CS, and the entry on its local PIT that corresponds to this Data message is deleted.

3.2 Integration options of CCN in LTE

In this section we propose five different possible options to integrate the CCN concept in EPC components. In the provided solutions, we assume that UE is not aware of CCN, and a proxy functionality is needed to be used to intercept and translate the request sent by a UE to a CCNx message. In the 1, 2, 3 and 5 proposed options the proxy functionality could be placed in eNodeB and in the option 4 the proxy functionality may be placed in CDN engine. Another assumption is that, servers that deliver the content are CCN capable. Figure 37 shows the LTE user plane.

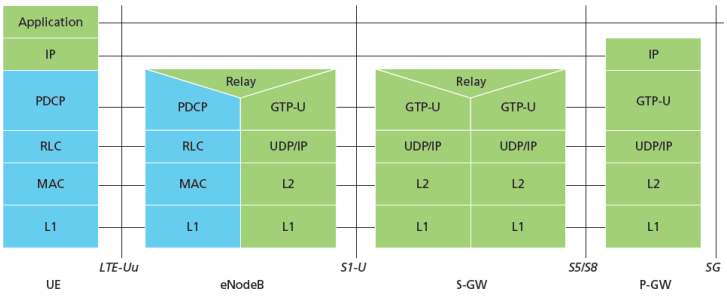

Figure 37: LTE user plane

3.2.1 Option 1 : Integrating CCN in eNodeB, S-GW and P-GW

In this architecture, it is considered that eNodeB, S-GW, and P-GW are CCN capable and are being able to maintain the Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS). It is considered that the routers deployed in the router infrastructure used to interconnect the EPS components are IP routers that are not CCN capable. The following subsections provide an overview of the user plane uplink and downlink operation.

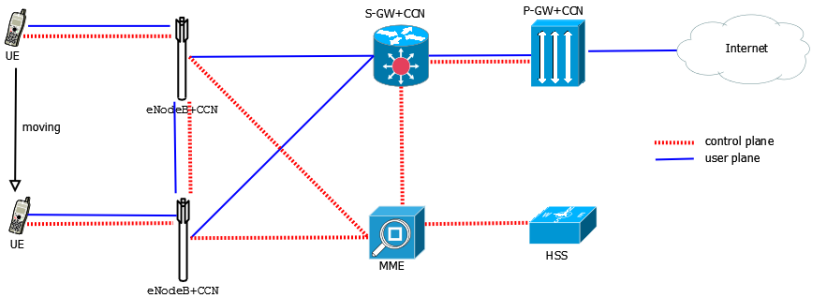

Figure 38: Deployment of CCN concept in the EPS Components

3.2.1.1 CCNx Uplink Communication Path

The message from the UE is sent to eNodeB over the LTE Radio Bearer. After receiving that message, the message will be translated by the "proxy" into CCNx message (composition: *IP+UDP+APP (Interest)*), the eNodeB notices the RBID (Radio Bearer Identifier) used by the UE to send the message. By using that RBID, the eNodeB can know what S1-TEID (Tunnelling End point Identifier) uplink that should be used to send this message to the S-GW since the eNodeB stores a one-to-one mapping between a radio bearer and an S1 Bearer to create the mapping between a radio bearer and an S1 bearer in both the uplink and downlink.

The eNodeB will check its CS to know whether there is the requested content in its local storage. If the requested content is found, this content will be sent to the UE directly over the same Radio Bearer used by the UE before. If no match is found in its CS, eNodeB will check its PIT to know whether there is the same Interest has been sent before. If a match is found, the Interest message will be discarded and a new interface pointing to the UE will be added to the interface list of PIT entry for that Interest message. If no match is found in the PIT, a new PIT entry that points to the UE's IP address will be installed. PIT will store the UE's IP address, RBID used by the UE, and the UE's UDP source port. Afterwards, the eNodeB will add GTP header, UDP (S1-U), IP (S1-U), and L1/L2 to the message (composition: L1/L2+IP(S1-U)+UDP(S1-U)+GTP+IP+UDP+ APP (Interest)) and send it to the S-GW.

In the S-GW, the GTP header of the received message will be removed. The S1-TEID is extracted from the GTP header to find the matching S5/S8-TEID uplink that should be used to send this message to the P-GW. Now, the composition of the message is IP+UDP+APP (Interest). The S-GW will check in its CS to know whether there is the requested content in its local storage. If the requested content is found, this content will be sent to the eNodeB serving the UE. This S-GW will be able to generate GTP header for this message since it knows what S1-TEID should be used to send the message to the eNodeB. If no match is found in its CS, S-GW will check its PIT to know whether there is the same Interest has been sent before. If a match is found, the Interest message will be discarded and a new interface pointing to the UE will be added to the interface list of PIT entry for

that Interest message. If no match is found in the PIT, a new PIT entry that points to the UE's IP address will be installed. By leveraging the mapping between an S1 bearer and an S5/S8 bearer in both the uplink and downlink stored by S-GW, PIT will store the S5/S8-TEID downlink that will be used by the P-GW to send the response message, i.e., Data message, to the S-GW. Besides, S-GW will also store the UE's UDP source port. Afterwards, the S-GW will add GTP header, UDP (S5/S8-U), IP (S5/S8-U), and L1/L2 to the Interest message (composition: L1/L2+IP(S5/S8-U)+UDP(S5/S8-U)+GTP+IP+UDP+ APP (Interest)) and send it to the P-GW.

In the P-GW, the GTP header of the received message will be removed. Now, the composition of the message is IP+TCP/UDP+APP (Interest). Then the P-GW will check in its CS to know whether the requested content/data is in its local storage. If the requested content/data is found, this content/data, using a Data message, will be sent directly to the S-GW serving the UE. The P-GW knows what S5/S8-TEID downlink should be used to send the message to the S-GW since it stores a mapping between a downlink message filter and an S5/S8 bearer to create the mapping between a traffic flow aggregate and an S5/S8 bearer in the downlink. If no match is found in its CS, P-GW will check its PIT to know whether there is the same Interest has been sent before. If a match is found, the message will be discarded and a new interface pointing to the UE will be added to the interface list of PIT entry for that Interest message. If no match is found in the PIT, a new PIT entry that points to the UE's IP address will be installed. PIT will also store the UE's UDP source port. Afterwards, the P-GW will send the Interest message to the destination address (composition: L1/L2+IP+UDP+Interest).

3.2.1.2 CCNx Downlink Communication Path

The composition of the Data message received by the P-GW is L1/L2+IP+UDP+APP (Data). When the P-GW receives the Data message from the internet, it will check its CS to know whether there is the same Data stored in its CS or not. If a match is found, the Data message will be discarded. If no match is found, the P-GW will cache this content/data and check its PIT to know where the Data message should be sent. For instance there are 3 IP addresses of UEs with their UDP ports in the PIT, the content/data (inside the Data message) will be encapsulated by using those IP addresses. The UDP destination ports of those Data messages will be changed according to the information obtained from the PIT. In this step, the P-GW has 3 CCN/IP messages to be sent. From the UE's IP address, the P-GW will be able to determine the S-GWs to which the messages should be sent. Then the P-GW will add the GTP headers to them. The P-GW determines S5/S8-TEIDs downlink will be used by those 3 CCN/IP messages by using Downlink Traffic Flow Template (DL-TFT). After the GTP headers are added to those messages, they will be forwarded to the S-GWs serving those UEs (composition: L1/L2(S5/S8-U)+IP(S5/S8-U)+UDP(S5/S8-U)+GTP+IP+UDP+ APP (Data)), and then the PIT entry for these Data messages will be deleted.

In the S-GW, the GTP header of the received message will be removed. The S5/S8-TEID is extracted from the GTP header to find the matching S1-TEID downlink should be used to send this Data message to the eNodeB. Now, the composition of the message is IP+UDP+APP (Data). The S-GW will check in its CS to know whether there is the same content/data stored in its CS or not. If a match is found, the Data message will be discarded. However, if no match is found, S-GW will cache the content/data and check its PIT to know where the Data message should be sent. For instance there are 2 IP addresses of UEs with their UDP ports and S5/S8-TEIDs in the PIT, the content/data (inside the Data message) will be encapsulated by using those IP addresses. The UDP destination ports of those Data messages will be changed according to the information obtained from the PIT. In this step, the S-GW has 2 CCN/IP messages to be sent. From the UE's IP address, the S-GW is able to determine the eNodeBs to which the messages should be sent. Then the S-GW will add the GTP headers to them. The S-GW determines that the S1-TEIDs downlink will be used by those 2 messages by leveraging the mapping between an S1 bearer and an S5/S8 bearer in downlink stored by the S-GW. After the GTP headers are added to those messages, they will be forwarded to the eNodeBs serving those UEs (composition: L1/L2(S1-U)+IP(S1-U)+UDP(1-U)+GTP+IP+UDP+ APP (Data)), and then the PIT entry for these Data messages will be deleted.

In the eNodeB, the GTP header of the received Data message will be removed. The S1-TEID is extracted from the GTP header to find the matching RBID that should be used to send this Data message to the UE. Now, the composition of the Data message is IP+UDP+APP (Data). The eNodeB will check its CS to know whether there is the same content/data stored in its CS or not. If a match is found, the message will be discarded. However, if no match is found, eNodeB will cache the Data and check its PIT to know where the Data should be sent. For instance there are 5 IP addresses of UEs with their UDP ports and RBIDs in the PIT, the Data (inside the message) will be encapsulated by using those IP addresses. The UDP destination ports of those messages will be changed according to the information obtained from the PIT. Afterwards, before those 5 Data messages are sent to the 5 UEs over their Radio Bearer (specified by their RBIDs), those 5 CCNx messages will be translated by the "proxy" into standard IP packets (so that the payload can be read by the UEs). After the packets are sent to the UEs, the PIT entry for these messages will be deleted.

3.2.1.3 Advantages

- Enhancing service continuity by accelerating content delivery to users
- User's traffics can be localized on the EPS core network by leveraging in-network caching
- Efficient content delivery; content can be delivered from the location close to users
- Reducing bandwidth consumption on the EPS core network and internet
- Saving network resource on the EPS core network and the internet
- Protocols used in EPS core network are not changed

3.2.1.4 Drawbacks

- Modifications of eNodeB, S-GW, and P-GW implementations are needed
- If UE is not aware of CCN, a proxy (that can intercept a request sent by a UE, translate that request into CCNx message, and maintain content retrieval) is required to be implemented in eNodeB
- Functions that currently run in P-GW, such as lawful interception, might need to be replicated on S-GW and eNodeB.

3.2.2 Option 2: Integrating CCN in Routers Deployed in EPS Core Router Infrastructure

In this architecture (see Figure 39), CCN+IP routers that are CCN and IP capable are deployed in the router infrastructure used to interconnect the EPS components. It is considered in this option that the EPS components (eNodeB, S-GW, P-GW) are not CCN capable.

The request message sent by the UE will go to the internet through eNodeB, S-GW, and P-GW. Since the routers deployed in the EPS core router infrastructure are CCN and IP capable, will intercept the Interest message hop-by-hop. For instance, when the eNodeB forwards the Interest message to the S-GW, the CCN+IP routers through which the Interest message is forwarded will intercept that Interest message and perform CCN's procedures (checking CS, PIT, FIB, etc.) to determine a decision that should be applied for this Interest message. If there is no CCN+IP router that can satisfy that Interest message, the Interest message will be sent to the Internet towards the server that advertised the content/data, through S-GW and P-GW. When the server that stores the content/data receives the Interest message, it will respond by sending a Data message to the UE through the P-GW, S-GW, and eNodeB. The CCN+IP routers which are traversed by the Data message should be able to cache that content/data before sending it to the next hop (another CCN+IP router, S-GW, or eNodeB).

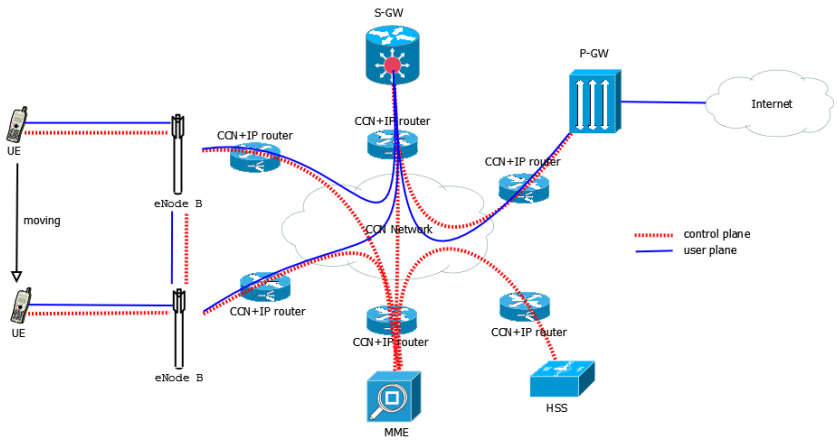

Figure 39: Deployment of CCN+IP routers in the middle of EPS core network

3.2.2.1 Advantages

- Does not change protocols used in EPS core network
- Modifications of S-GW and P-GW implementations are not needed

3.2.2.2 Drawbacks

- A proxy (that can intercept a request sent by a UE, translate that request into CCNx message, and maintain content retrieval) is required to be implemented in the eNodeB
- Functions that currently run in P-GW, such as lawful interception , might need to be replicated on S-GW and eNodeB
- Other drawbacks are as follows: This architecture generates some issues. From Figure 37 we can see that GTP-U is used among eNodeB, Serving GW, and PDN GW. Since the CCN+IP routers are deployed in the EPS core router infrastructure to enhance content delivery service, the CCN+IP routers need to be able to forward request (Interest message) sent by the UE to the node (CCN+IP router) that has a copy of requested content (content/data). If there is a CCN+IP router that has the requested content/data, the Data message has to be sent to the UE through the reverse path that was used by the Interest message.

3.2.2.3 Problem Description

- The Interest sent by UE is an "application" layer message (see the protocol stack at UE side in the Figure 37). After the Interest message is processed by the eNodeB, CCN+IP routers need to forward the Interest to the destination. The IP address recognized by the CCN routers is only the first IP layer, starting from the bottom of the protocol stack shown in the Figure 37 (see UDP/IP layer). The CCN+IP routers will recognize the GTP-U layer, the second IP layer (starting from the bottom of the protocol stack shown in the Figure 37), and the application layer as a payload. They will not be able to check the Interest requested by the UE since they cannot read and use the GTP protocol. This problem also happens when the Interest/traffic flows from serving GW to PDN GW.
- When an Interest message is sent to the internet via PDN GW, and the server in the internet can satisfy the Interest, the requested Data/content will be sent by the server to the UE via the PDN GW, serving GW, and eNodeB using a Data message. The CCN+IP routers which are traversed by the Data message need to be able to cache that content/data before sending the Data message to the next hop (another CCN+IP router, S-GW, or eNodeB). However, the question is how are CCN+IP routers going to be able to cache the content if they cannot read GTP.

- Since the size of GTP header is fixed, we can easily get the content; but we have to modify the functionality of CCN+IP router in reading a message.
- Another case is, when there is a CCN+IP router that has a copy of the requested content/data (the Interest does not need to be forwarded to PDN-GW). How does the CCN+IP router encapsulates the content so that the encapsulated content can be sent to the right serving GW and eNodeB (S-GW and eNodeB communicate each other by using GTP, but the CCN+IP router does not understand GTP).
- The router needs some information (e.g. TEID) to generate the GTP-U header. When the Data message is going to be forwarded back through the reverse path, the tunnel endpoint identifier (TEID) of specific EPS component (e.g. serving GW or eNodeB) is needed so that the message can be sent to the right EPS component (e.g. S-GW or eNodeB).

3.2.3 Option 3: Integrating CCN in eNodeB, S-GW and P-GW and in Routers Deployed in EPS Core Router Infrastructure

This architecture (see Figure 40) is the combination of option 1 and option 3. In this architecture, eNodeB, S-GW, and P-GW implement the basic concept of CCN in which Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS) are maintained. The existing FIBs of those EPS components are not change; two functionalities added to them are PIT and CS. Furthermore it is considered that the routers that are deployed in the EPS core router infrastructure are CCN and IP capable.

3.2.3.1 Drawbacks

- Similar to the architecture used in option 2, the main drawback of this architecture rely on the CCN+IP routers deployed in the middle of EPS components. The details of this drawback are the same as explained in option 2 (see section 3.2.2).
- Modifications of eNodeB, S-GW, and P-GW implementations are needed
- If UE is not aware of CCN, a proxy (that can intercept a request sent by a UE, translate that request into CCNx message, and maintain content retrieval) is required to be implemented in eNodeB
- Functions that currently run in P-GW, such as lawful interception , might need to be replicated on S-GW and eNodeB

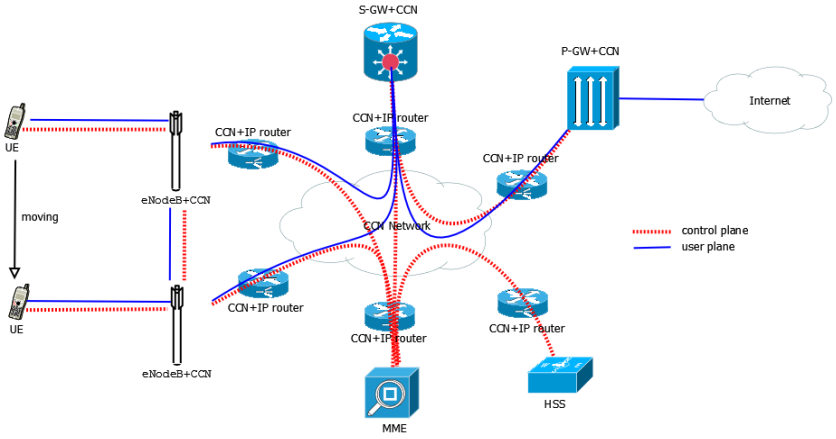

Figure 40: Combination of option 1 and option 3

3.2.4 Option 4: Mobile CDN Solution

Figure 41, shows the architecture of mobile CDN solution. In this architecture, content is stored and served at the edge of EPS core network. CDN engines/repositories are used to maintain and support content retrieval and are considered to be CCN capable. The CDN engine implements a proxy that can intercept a request sent by a UE, translate that request into a CCNx message, and maintain content retrieval. Since the CDN engine is aware of CCN, it will inspect all CCNx messages (by checking its CS, PIT, and FIB) going to and coming from the Internet via the router connected directly to the P-GW.

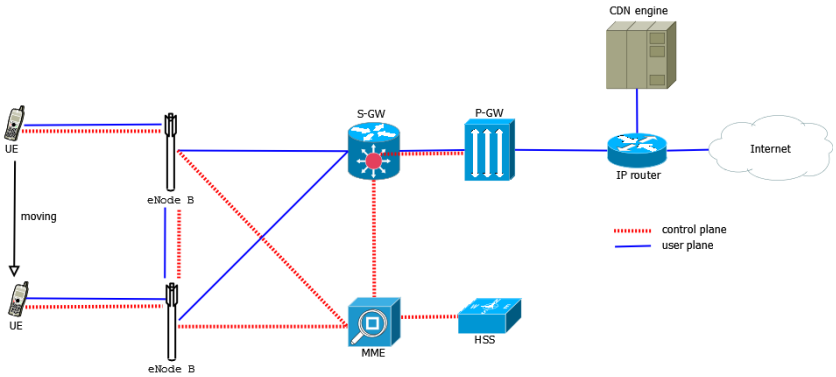

Figure 41: Mobile CDN solution

3.2.4.1 Advantages

- Enhancing service continuity by accelerating content delivery to the UE
- Accelerates the content delivery to users and reduce bandwidth consumption on network, due to the use of the CDN engines/repositories
- No modifications on EPS components and protocols

- Functions that currently run in P-GW, such as lawful interception, are not required to be replicated on S-GW and eNodeB

3.2.4.2 Drawbacks

- User's traffics cannot be localized on the EPS core network; session establishment between a user (UE) and the content source is needed
- The closest location which is possible to deliver content to a user is only from the location of CDN engine/repository which is placed on the edge of the EPS core network
- Content delivery is not as efficient as the content delivery on the option 1
- If UE is not aware of CCN, a proxy (that can intercept a request sent by a UE, translate that request into CCNx message, and maintain content retrieval) is required to be implemented in CDN engine

3.2.5 Option 5: Integrating CCN in eNodeB, S-GW and P-GW and CDN Repositories

Figure 42, illustrates the architecture of option 5. This architecture is an integration of the architectures proposed in Options 1 and 4. In this architecture, the same as option 1, it is considered that eNodeB, S-GW, and P-GW are CCN capable. Furthermore, it is also considered that the CDN engines/repositories are CCN capable. In this option also, similar to option 1, the routers deployed in the EPS core router infrastructure are IP routers and are not CCN capable.

The data flow procedures in the uplink and the downlink between the UE and the internet are similar to option 1. However in this scenario, the CDN engine/repository are also CCN aware and are able to provide the requested content/data to the CCN infrastructure faster than the main server that stores the content. The advantages and shortcomings of option 5 are the same as the ones identified for option 1 and 4. However utilizing the CDN repository could effectively accelerate the content delivery to users and reduce bandwidth consumption on network.

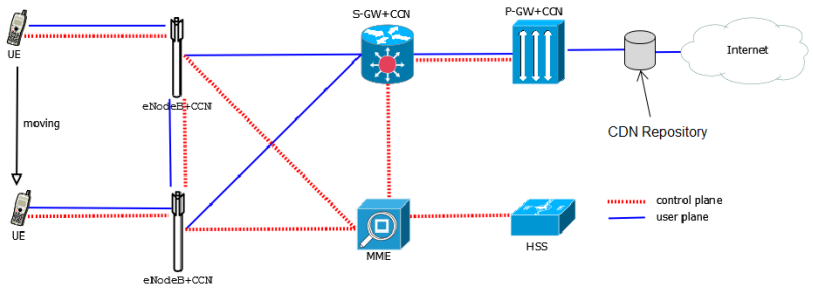

Figure 42: Combination of option 1 and option 4

3.2.6 Selected Option

By studying the advantages and drawbacks of each of the 5 proposed options we concluded that option 1 and option 5 are the most suitable to integrate the CCN concept into LTE.

The reasons of selecting option 1 are:

- It enhances service continuity by accelerating content delivery to users
- It reduces bandwidth consumption on the EPS core network and internet
- By leveraging in-network caching, the user's traffics will be localized on the EPS core network
- It optimizes network resources on the EPS core network and the internet
- It does not affect the protocols used in EPS core network
- It could improve the content delivery by delivering content from the location close to users

The reasons of selecting option 5 are the same ones of selecting option 1 with the following additional one:

- Accelerates the content delivery to users and reduce bandwidth consumption on network, due to the use of the CDN engines/repositories

In the next section, we provide more details in order to plan, design and implement option 5 to support service continuity for the mobile users in the virtualized LTE.

3.3 Design of the CCN Integration in eNodeB, S-GW, P-GW and CDN Engines/Repositories

Figure 43 shows the logic architecture of option 5, described in Section 3.2.5. In this architecture, it is considered that eNodeB, S-GW, and P-GW are CCN capable. Furthermore, it is also considered that the CDN engines/repositories are CCN capable. Moreover, the routers deployed in the EPS core router infrastructure are IP routers and are not CCN capable.

Note that Option 1 is using the same logic architecture and flow diagrams, with the difference that the CDN repository component is not used and it is not involved in the provided flow diagrams.

Here the assumption is that UE is not aware of CCNx protocols, meaning that a proxy functionality is needed to be placed in eNodeB to intercept and translate the request sent by UE to a CCNx message, and vice versa.

Utilizing a CDN engine/repository placed between the internet and PDN gateway, improves the speed of the content delivery to users and enhances the continuity of services to the mobile users. It could also optimize bandwidth consumption on the EPS core network and internet.

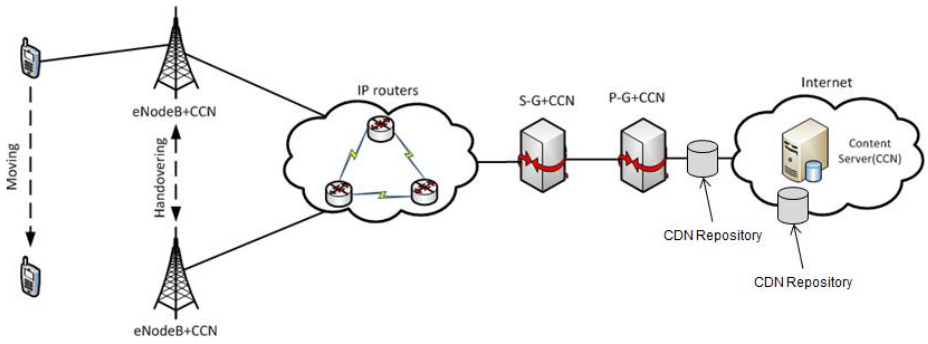

Figure 43: Deployment of option 5

Figure 44, gives a brief overview of the CCNx uplink communication path procedure. In this scenario the CCNx uplink communication path procedure can be accomplished using the following steps:

- A request is sent by UE to eNodeB over the LTE Radio Bearer
- The request will be translated by the "proxy" into a CCN Interest message.
- eNodeB notices the RBID used by the UE and finds the S1-TEID uplink to send the Interest message to the S-GW.
- The eNodeB performs the CCN procedures as follows:
  - Checks its CS to find out whether there is the requested content/data in its local storage. If this content/data is found, it will be sent to the UE.
  - If the content/data is not found, then the eNodeB will check its PIT to know whether the received Interest message has been previously received. If a match is found, the message will be discarded and a new interface pointing to the UE will be added to the interface list of PIT entry for that Interest message.
  - If not, a new PIT entry that points to the UE's IP address will be installed. PIT will store the UE's IP address, RBID used by the UE, and the UE's UDP source port. Afterwards, the eNodeB will add a GTP header, and send the Interest message to the S-GW.
- After receiving the Interest message, the S-GW removes the GTP header and finds the matching S5/S8-TEID uplink used to send this message to the P-GW. Now, the composition of the message is IP+UDP+APP (Interest). The S-GW performs the same CCN procedures as the ones accomplished by the eNodeB.
- After receiving the Interest message, the P-GW removes the GTP header of the received message. Now, the composition of the message is IP+TCP/UDP+APP (Interest). Then the P-GW performs the same CCN procedures as the ones accomplished by the S-GW. Afterwards, the P-GW sends the Interest message (L1/L2+IP+UDP+Interest) to the CDN engine/repository.
- The CDN engine/repository performs the CCN procedures and if needed sends the Interest message towards the server that stores the requested content/data.

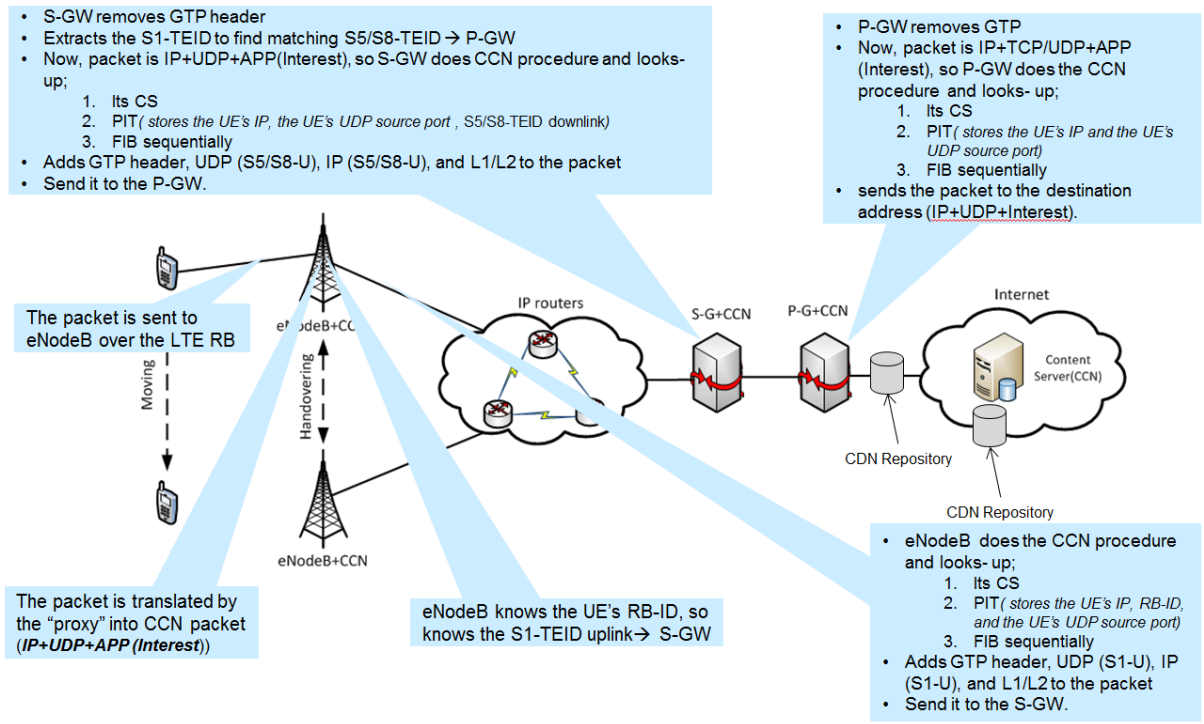

Figure 44: CCNx uplink communication path procedure

Figure 45 gives a brief overview of the CCNx downlink communication path procedure. The CCNx downlink communication path procedure can be accomplished using the following steps:

- The composition of the Data message received by the CDN engine/repository is L1/L2+IP+UDP+APP (Data). When the CDN engine/repository receives the Data message from the server via Internet, will perform the CCN procedures as follows:
  - It check its CS to find out whether content/data carried by the Data message is the same as the one stored in its CS or not. If no match is found, then the CDN repository will cache this content/data.
  - It will then check its PIT to know where the Data message should be sent.
  - The Data message will then be sent to the P-GW based on its PIT
- After receiving the Data message the P-GW accomplishes the same CCN procedures as the ones performed by the CDN engine/repository. P-GW determines the S5/S8-TEIDs downlink by using Downlink Traffic Flow Template (DL-TFT) and uses it to determine the corresponding S-GW. P-GW adds the GTP headers to the Data message and forwards it to the S-GW. The PIT entry for this Data message is deleted.
- In the S-GW, the GTP header of the received message is removed and the matching S1-TEID downlink is determined using the S5/S8-TEID. Now, the composition of the Data message is IP+UDP+APP (Data). S-GW performs the same CCN procedures as the ones accomplished by the P-GW. Then the S-GW adds the GTP headers to Data message and forwards it to the

corresponding eNodeB based on S1-TEID downlink. Then the PIT entry for the message is deleted.

- After receiving the Data message, the eNodeB removes the GTP header to find the matching RBID in order to know where the Data message should sent. Now, the composition of the message is IP+UDP+APP (Data) and eNodeB does the same CCN procedures as done by S-GW. Afterwards, before the Data messages is sent to the UE over their Radio Bearer based on its RBIDs, the Data message is translated by the "proxy" into standard IP messages. Then the message is sent to the UEs, and the PIT entry for the Data message is deleted.

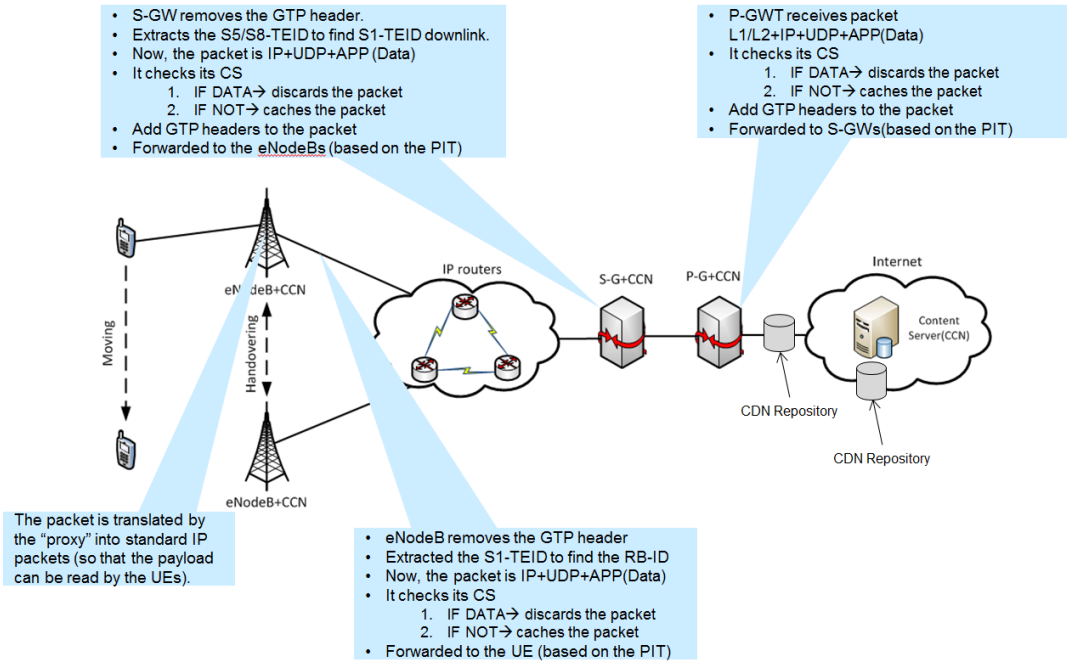

Figure 45: CCNx downlink communication path procedure

3.3.1 Content Migration Support

In this section we propose two different possible solutions to support content migration due to user mobility when the CCN functionalities are integrated in virtualized LTE components. In the figures used in this section, the CCN routers represent the eNodeB that are capable of using the CCN function and protocols capabilities. Moreover, the SO represents the service orchestrator, the MSM represents the Mobility Support Manager and the MOBaaS represents the Mobility Prediction System, which is realised by the Mobility and Bandwidth Availability Prediction as a Service (MOBaaS).

This proposal is based on the CCNx protocol and is extended in order to support mobility and content migration when users are moving from one data centre to another one. In this scenario it is considered that the client is CCN capable.

3.3.1.1 Without Mobility Prediction

Figure 46, shows step-by-step of mobility and content migration support procedure for the client interested to download content/data provided by a source node. In this scenario the mobility prediction provided by the MOBaaS is not applied.

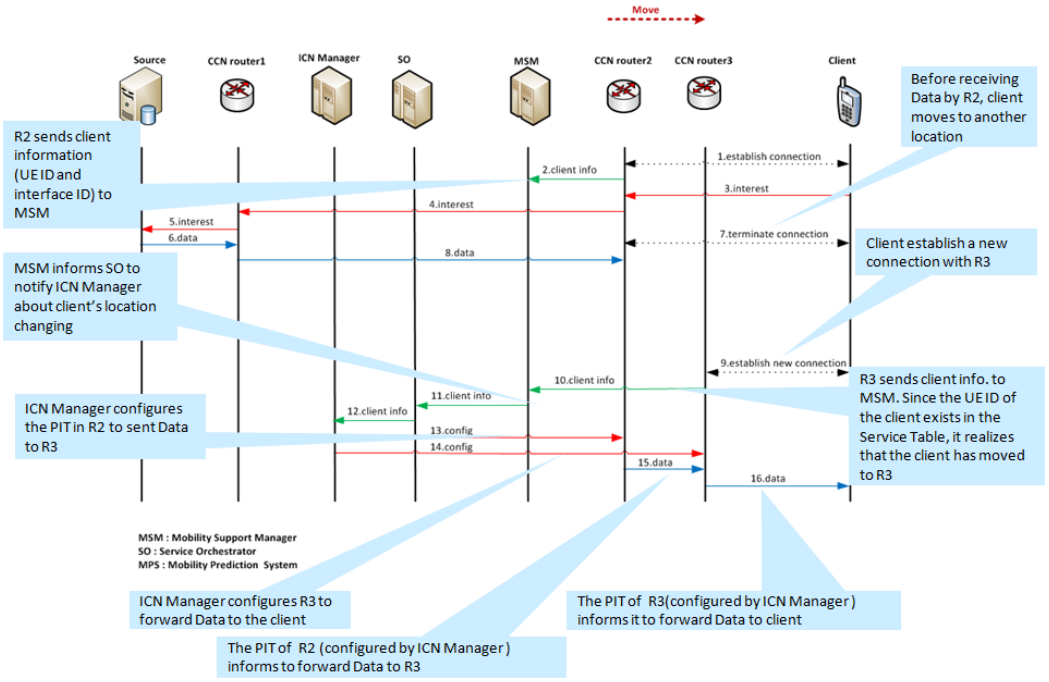

Figure 46: Content Migration support procedure without mobility prediction

The content migration procedure can be described using the following steps:

- Step 1: Client establishes a connection to CCN router 2
- Step 2: CCN router 2 sends the client information containing node ID and interface ID to MSM.
- Step 3: Client sends the Interest message via the CCN router 2. When the CCN router 2 receives the Interest message from the client, a look-up is performed on its CS, PIT, and FIB sequentially. In this case, the matching information is only found in its FIB that informs the Interest has to be forwarded to the interface pointing to CCN router 1.
- Step 4: CCN router 1 receives the Interest message from CCN router 2 and does the same procedure performed by CCN router 2. After doing look-up on its CS and PIT, and no matching information is found, it checks its local FIB.
- Step 5: The CCN router 1's FIB informs it that the Interest has to be forwarded to the interface pointing to the source.

- Step 6: When the source receives the Interest sent by CCN router 1, it looks-up on its CS. When the matching content is found, it sends the content/data using the Data message as the response for the Interest through the arrival interface of Interest (to the CCN router 1) .
- Step 7: Before the requested Data is received by CCN router 2, the client moves to another location and terminates its connection with CCN router 2.
- Step 8: When the CCN router 1 receives the Data message forwards it to the CCN router 2, based on its PIT, and stores it in its cache.
- Step 9: After the handover procedure is completed, the client establishes a new connection with CCN router 3.
- Step 10: CCN router 3 sends the client information containing node ID and interface ID to MSM.
- Step 11: MSM determines that client has been moved, because it receives client information from two different routes.
- Step 11, 12: MSM informs SO to notify ICN Manager about the client's location changing.
- Step 13: ICN Manager starts to configure PIT in the CCN router 2 to be able to forward the Data message to the CCN router 3.
- Step 14: ICN Manager also starts to configure PIT in the CCN router 3 to be able to forward Data message to the client.
- Step 15: CCN router 2 sends the Data message to the CCN router 3 based on its PIT and stores it in its cache.
- Step 16: CCN router 3 sends the Data message to the client based on its PIT and stores it in its cache.

3.3.1.2 With Mobility Prediction

Figure 47, shows a similar procedure as shown in Figure 46. However in this case the prediction service MOBaaS is used, which predicts the future location of the mobile client and informs the MSM. The procedure steps used to support content migration in this proposal are similar to the one described in section 3.3.1.1. However, in this scenario the MSM uses this prediction in order to trigger the migration of the content/data from CCN router 2 to CCN router 3 before the client starts the connection establishment procedure.

In this case the ICN Manager could configure the PIT in CCN router 2 and 3 in advance and the Data message could be moved before the client starts the connection establishment procedure.

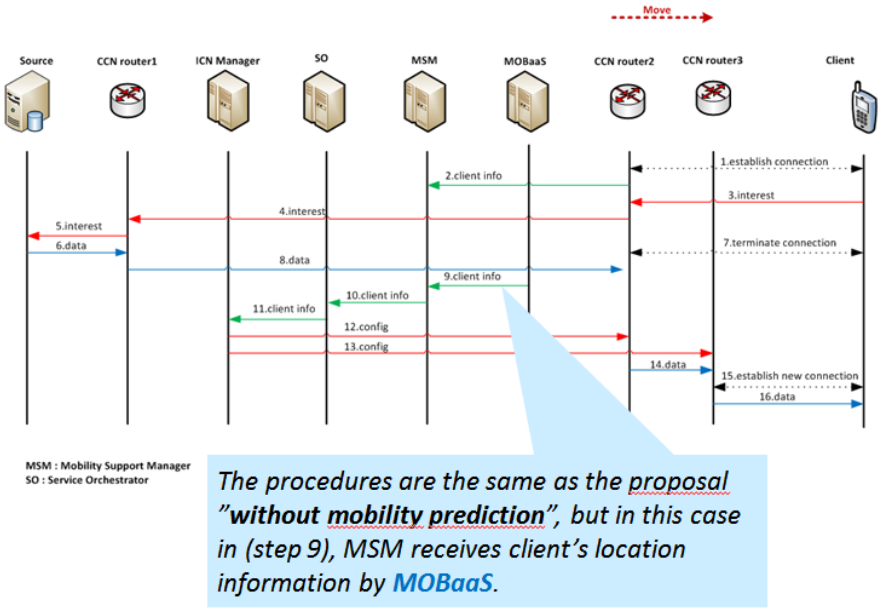

Figure 47: Content Migration support procedure with mobility prediction

In Figure 48, the two proposed scenarios are compared. It can be seen that MOBaaS could effectively decrease the time between establishing connection by the client and forwarding the Data message from CCN router 2, via CCN router 3, to the client.

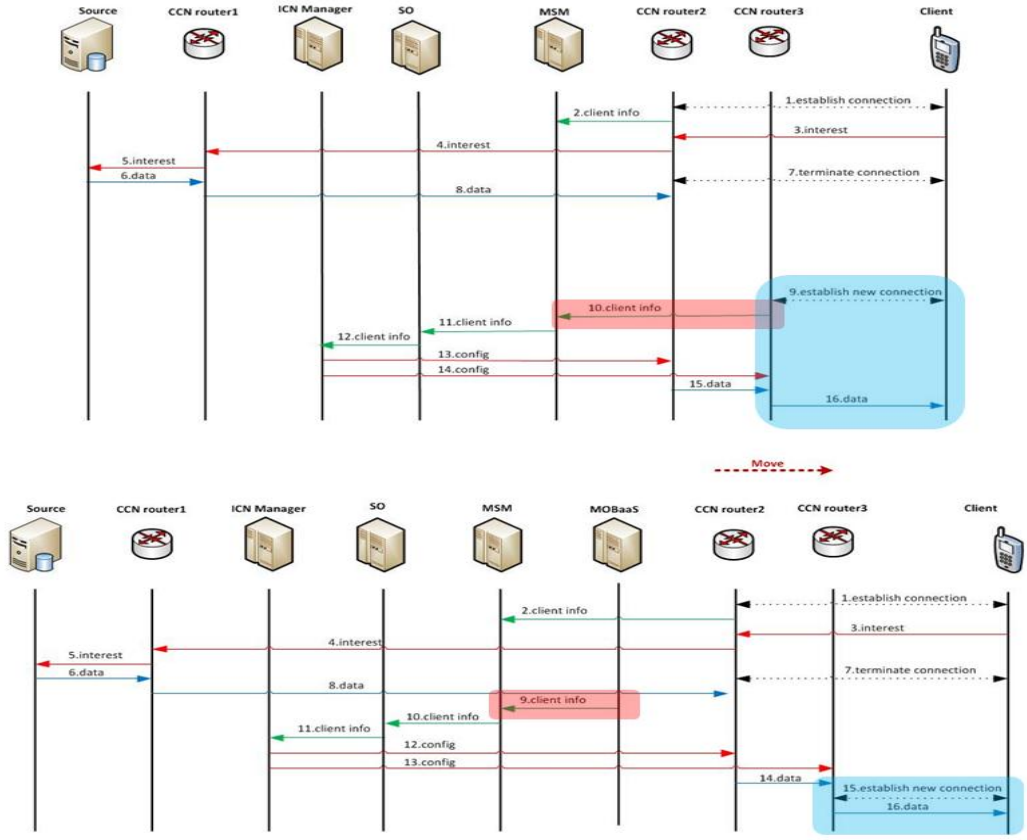

Figure 48: Comparison of content migration support with/without mobility prediction

3.3.2 VM (Container) and Content Migration Support

In this section we propose two other possible solutions that could support content and VM and/or container migrations. The main difference between this section and section 3.3.1 is that in this case the migration solution is used to migrate not only content, but also VMs and/or the container running on a VM, while the one described in section 3.3.1 is used to migrate only content of a relatively small size, which can be carried by one IP packet,

In the figures used in this section the yellow and violate coloured CCN router, CCN Router 2 and 3 represent data centres that are able to host VMs and containers that can run instances like eNodeB and CCN routers. In particular, in this case the CCN router 2 represents the source data centre, where a VM and or a container need to be migrated to the target data centre, which in this case is represented by CCN router 3. It is important to be noticed that in this example it is assumed that the source data centre is running an eNodeB and a CCN router (CCN router 2) and the target data centre is running only an eNodeB at the moment that the subscriber moves at the target data centre. This procedure describes how the VM (container) together with content can be moved from the source data centre to the target data centre using the extended CCNx protocol.

This proposal is based on the CCNx protocol and is extended in order to support VM (container) and content migrations when users are moving from one data centre to another one.

3.3.2.1 Without Mobility Prediction

Figure 49 shows the step-by-step procedure of VM and content migration support between two data centres for the client who is moving between two data centres. In this scenario the mobility prediction, i.e., MOBaaS, is not applied. Here we assume that at the source data centre (i.e., CCN router 2) a VM is running the functionality of eNodeB and CCN router. Due to e.g., the fact that many users are moving from the area of the source data centre towards the target data centre, the VM that is running on the source data centre needs to be copied and migrated to the target data centre, e.g., CCN router 3.This means that the CS, PIT and FIB used at the source data centre will also be copied at the target data centre.

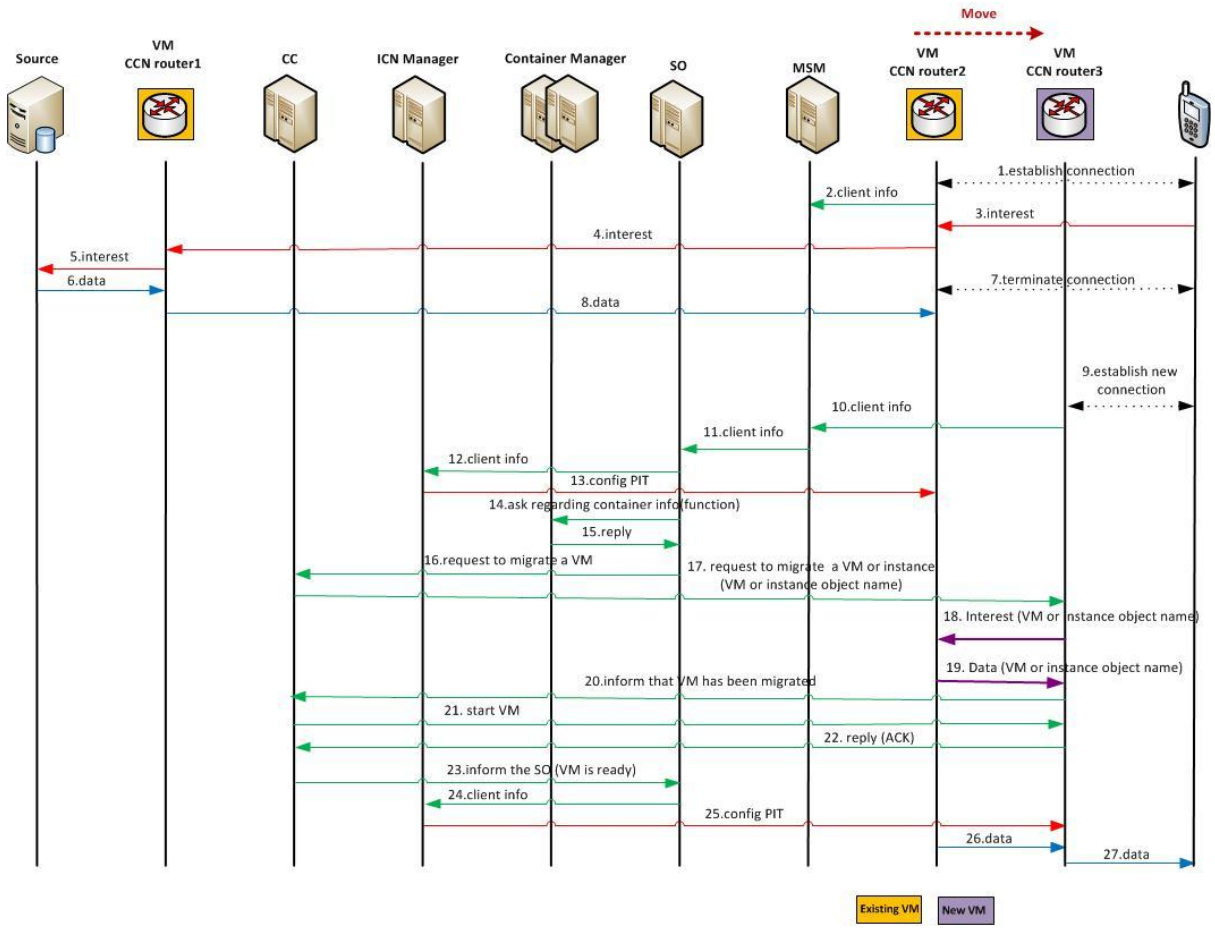

Figure 49: Content and VM Migration support procedure without mobility prediction

The content and VM migration procedure that is accomplished by this proposal can be provided using the following steps:

- Step 1: Client establishes a connection to CCN router 2
- Step 2: CCN router 2 sends the client information containing node ID and interface ID to MSM.
- Step 3: Client sends the Interest message via CCN router 2. When the CCN router 2 receives the Interest message from the client, a look-up is performed on its CS, PIT, and FIB sequentially. In this case, the matching information is only found in its FIB that provides information that the Interest message has to be forwarded to the interface pointing to CCN router 1.
- Step 4: CCN router 1 receives the Interest message from CCN router 2 and performs the same procedure as the one accomplished by CCN router 2. After completing the look-up on its CS and PIT, and no matching information is found, the CCN router 1 checks its local FIB.
- Step 5: The CCN router 1's FIB provides information that the Interest message has to be forwarded to the interface pointing to the source.

- Step 6: When the source receives the Interest sent by CCN router 1, it looks-up on its CS. When the matching content is found, it sends the content/data, using a Data message as the response for the Interest message through the arrival interface of the corresponding Interest (to the CCN router 1) .
- Step 7: Before the requested Data message is received by CCN router 2, the client moves to another location and terminates its connection with CCN router 2.
- Step 8: When the CCN router 1 receives the Data message, forwards it to the CCN router 2, based on its PIT, and stores it in its cache.
- Step 9: After the handover procedure is completed, the client establishes a new connection with the target data centre.
- Step 10: The target data centre forwards the client information containing node ID and interface ID to MSM.
- Step 11: MSM determines that client has been moved, since it receives client information from two different routes.
- Step 11, 12: MSM informs the SO to notify ICN Manager about the client's location changing.
- Step 13: ICN Manager starts to configure PIT in the CCN router 2 to be able to forward data to CCN router 3.
- Step 14: SO decides to copy and the VM that runs the CCN router functionality in the source data centre and migrate it from the source data centre (location of CCN router 2) to the target data centre (location of CCN router 3). The SO requests the required information from CM(Container Manager).
- Step 15: CM provide the needed information to migrate the VM to SO.
- Step 16: SO provides this information to CC and requests to migrate the VM to the target data center (location of CCN router 3).
- Step 17: CC ask from Target data center (Target eNodeB, location of CCN router 3) to send the request message to migrate a VM (or instance object name).
- Step 18: Target data center (Target eNodeB) send an Interest message (VM or instance object name) to the source data center (Source eNodeB)
- Step 19: Source eNodeB migrate the VM (instance object name) to the Target eNodeB.
- Step 20: Target eNodeB. sends a message and informs CC that the VM has been migrated.
- Step 21: CC sends a command to Target eNodeB to start up the CCN functionality.
- Step 22: Target eNodeB sends ACK and informs CC that the VM has been migrated and the new CCN router functionality that uses CS,PIT and FIB of CCN router is running.
- Step 23: CC informs the SO that the new VM is available and that the functionality of CCN router 3 is running.

- Step 24: SO notifies ICN Manager about the client's information containing node ID and interface ID and about the fact that the CS, PIT and FIB running on the CCN router 3 needs to be configured accordingly.
- Step 25: ICN Manager starts to configure PIT in the CCN router 3, such that is able to be able to forward the Data message to the client.
- Step 26: CCN router 2 sends the Data message to the CCN router 3 based on its PIT and stores it in its cache. Note that this step can be performed also right after step 13.
- Step 27: CCN router 3 sends the Data message to the client based on its PIT and stores it in its cache.

3.3.2.2 With Mobility Prediction

Figure 50 shows a similar procedure as the one shown in the Figure 49. However in this case a mobility prediction system is used. In the case of migration the MOBaaS informs the MSM about the future movement of the client.

The procedure steps to support content, VM and container migration in this proposal are similar to the previous ones, described in section 3.3.2.1, however in this scenario the mobility prediction service, MOBaaS, is used in order to trigger the migration of the VM, container and content/data from CCN router 2 to CCN router 3 before the client starts the connection establishment procedure.

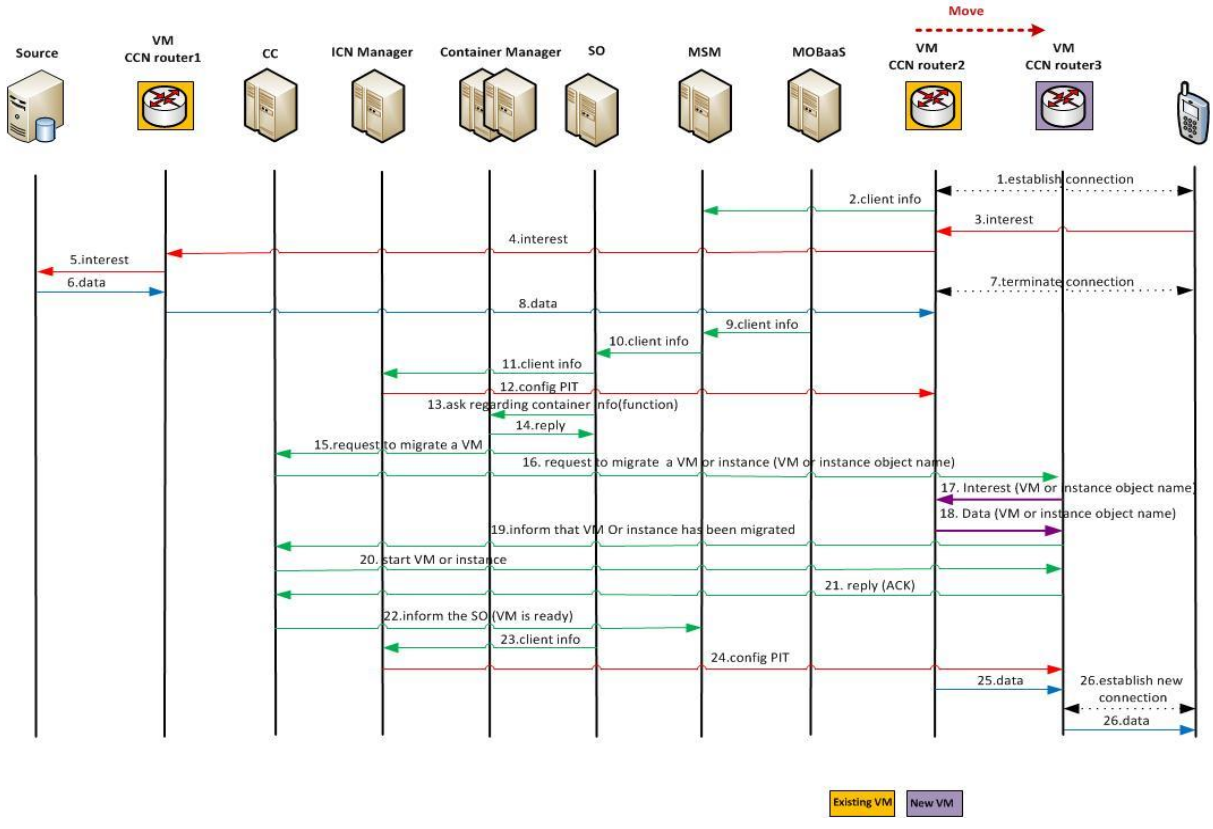

Figure 50: Content and VM Migration support procedure with mobility prediction

In Figure 51, the two proposed scenarios to support mobility, content and VM (container) migration are compared. From this figure it can be deduced that MOBaaS could also effectively decrease the time between establishing connection by the client and forwarding the Data message from CCN router 2, via CCN router 3, to the client.

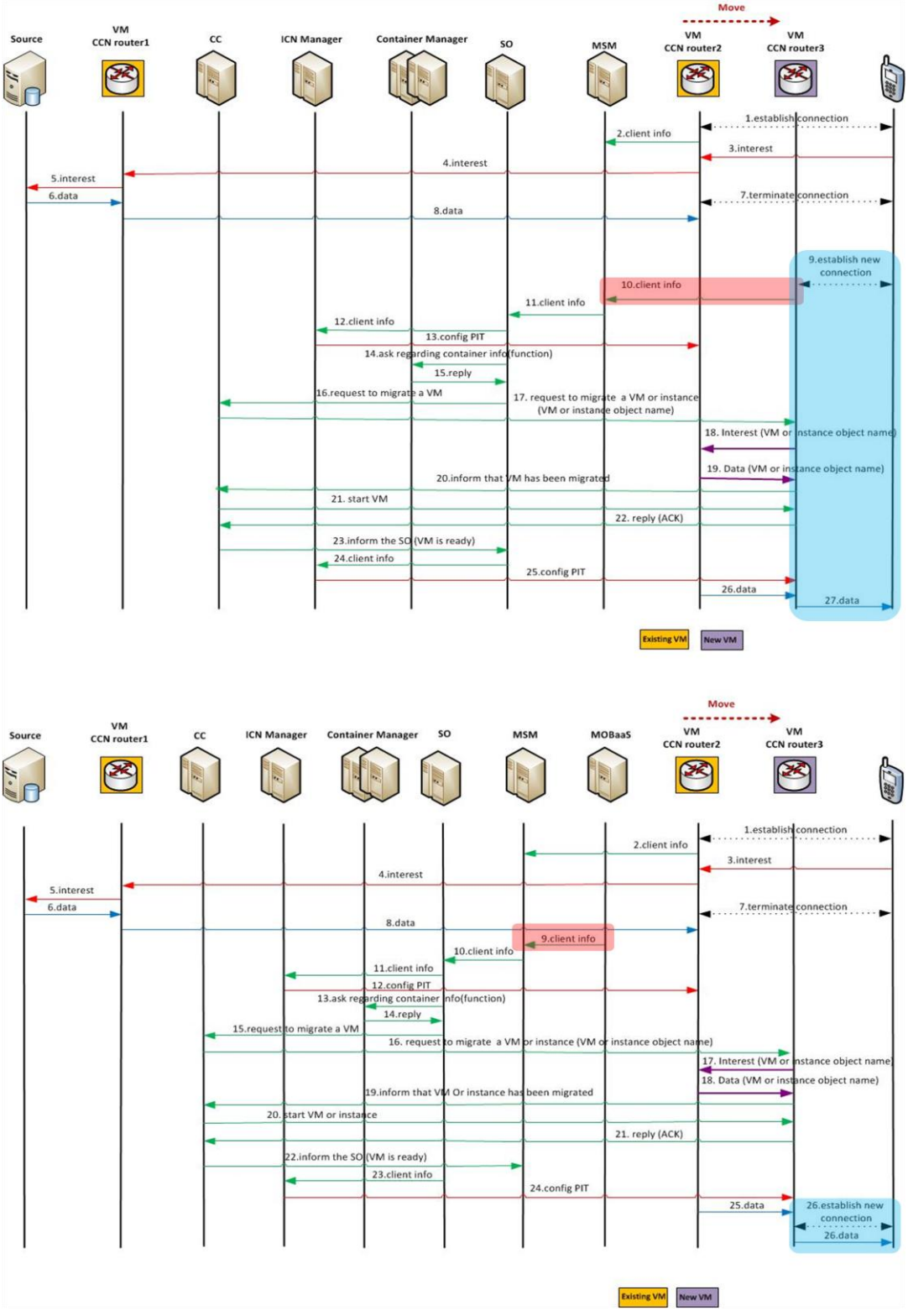

Figure 51: Comparison of content and VM migration support with/without mobility prediction

Chapter 4

Simulation Experiments

This chapter consists of four sub-chapters that discuss the simulation environment, network topology, performance metrics, and experiment scenarios. The goal of the simulations is to evaluate the service continuity solution (CCN) in cloud based LTE systems. In order to achieve that, we performed several simulations to study the migration of content and VM between two (source and target) virtualized eNodeBs served by two different data centres. These eNodeBs support CCNx functionality, which is only used by CCNx traffic. The type of handover that we simulated is the X2 handover, and the type of traffic/service that we simulated is video streaming. The details of simulation experiments that we have performed are explained in the following sections.

4.1 Simulation Environment and Assumptions

The simulator that we used in this assignment is ns-3 (ns-3.17 release), a discrete event network simulator for internet systems. The main module in ns-3 that we used is LTE module (LENA) [59].

The ns-3 LTE model (LENA) is a software library that allows the simulation of LTE networks, optionally including the Evolved Packet Core (EPC) [60]. LENA has two main components as follows:

- *LTE model*: consists of LTE Radio Protocol Stack (RRC, PDCP, RLC, MAC, PHY) which reside within UE and eNodeB nodes.
- *EPC model*: consists of core network interfaces, protocols, and entities which reside within SGW, PGW, and MME nodes, and partially within the eNodeB nodes. In LENA, functionalities of SGW and PGW are implemented within a single node.

The figure below is the overview of LTE-EPC simulation model:

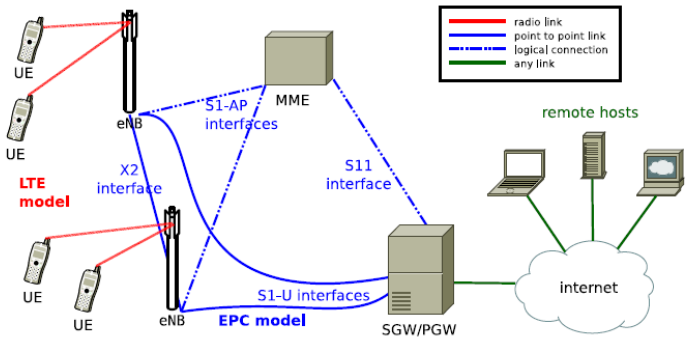

Figure 52: Overview of LTE-EPC simulation model, copied from [60]

In LENA, S1-AP and S11 interfaces are modelled in a simplified way since MME node is not modelled/created in LENA. Interaction between MME and SGW/PGW, and between MME and eNodeB are modelled only by using a function call. On the other hand, X2-AP interface is modelled in a more realistic way by using protocol data units sent over an X2 link modelled as a point-to-point link. S1-U interface connecting eNodeB and SGW/PGW is also modelled in a more realistic way by using a point to point link.

In order to simulate a cloud based LTE system, we used several assumptions in the experiments as the following ones:

- eNodeB and SGW/PGW are virtualized. It means that they are hosted in data centres. In this simulation experiments it is considered that the SGW/PGW and eNodeB are hosted in different data centres.
- The source eNodeB and target eNodeB that we used for simulating handover are hosted in two different data centres.
- eNodeB and SGW/PW support CCNx protocol functionality. However, this CCNx functionality is only used by the CCN traffic. Other type of traffic is bypassing this functionality.
- Cloud components used to support service continuity (e.g. CC, MSM, SO, CM and ICN Manager) are modelled in our simulation experiments within a single entity which is named as Virtualization Controlling Platform (VCP).
- The VCP node can be hosted in any data centre platform.
- UE is aware of CCNx, since there is no implementation of a proxy in eNodeBs.
- The CCN application implemented in UEs is used for video streaming.
- Not all CCNx features are implemented (only routing, forwarding, and caching mechanisms are implemented). CCNx retransmission mechanism is not implemented on the UE side since the CCN application that we implemented in the experiments is for video streaming. In case of live streaming, retransmission of lost segments can increase delay since the video streaming application has to wait for the retransmission of those lost segments before it continue processing newer data.
- The remote host (server) is aware of CCNx

LENA [59] provides some tracers, such as RLC, PDCP, MAC, and PHY tracers, that we can use to measure performance of LTE model. In the simulations, we used the RLC tracer in order to measure traffic load in the radio access network. Furthermore in order to measure throughput, packet loss, and Round Trip Time (RTT) delay, we implemented our own tracer that we installed in the application layer of UE. This tracer will count the number of packet sent and received by a UE and record transmitting and receiving time of packets.

4.2 Simulation Topology and Parameters

This section explains the simulation topology (simulation models) and parameters that we used in the experiments.

4.2.1 Simulation Topology

This section describes the simulation topology that has been used in the accomplished simulation experiments.

Figure 53 shows the simulation topology that we designed including the locations of enhancements that we made. We implemented several enhancements in LENA (LTE module) by implementing several main CCN concepts (PIT, FIB, and CS), which are used for routing, forwarding, and caching, into eNodeB and SGW/PGW nodes. Another enhancement that we made is the deployment of an IP router network (see Figure 54) in the middle of EPS (Evolved Packet System) components as depicted in Figure 53. In the current implementation of LENA [59] the S1-U link (the link connecting eNodeB and SGW/PGW) and X2 link (the link between two eNodeBs) are implemented only by using direct point-to-point connection. Since we would like to simulate communications between two or more data centres, the existence of IP router network connecting those data centres has an important role in our experiments. By implementing IP routers in the middle of LTE core network elements, the X2 tunnel is not established using direct point-to-point connection anymore. Packets sent through X2 tunnel will be forwarded to the destination through IP routers connecting the source data centre (source eNodeB) to the target data centre (target eNodeB).

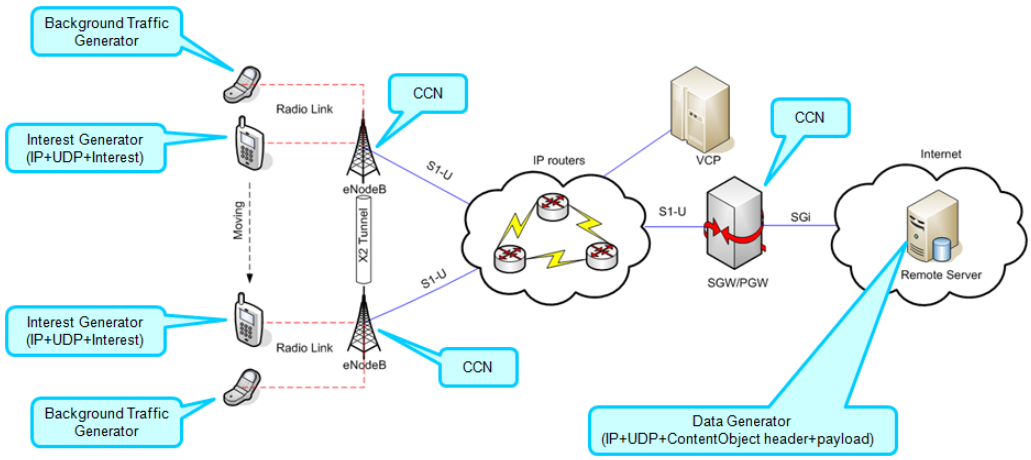

Figure 53: Simulation topology

In our experiments, we used some parts of network topology of Ebone (an Internet Service Provider (ISP) in Europe) [62]. This topology (see Figure 54) was inferred using Rocketfuel (an ISP topology

mapping engine) by University of Washington in 2002. The parts of Ebone's network that we used in the simulations are located in Amsterdam, Rotterdam, Antwerp, and Brussels [63].

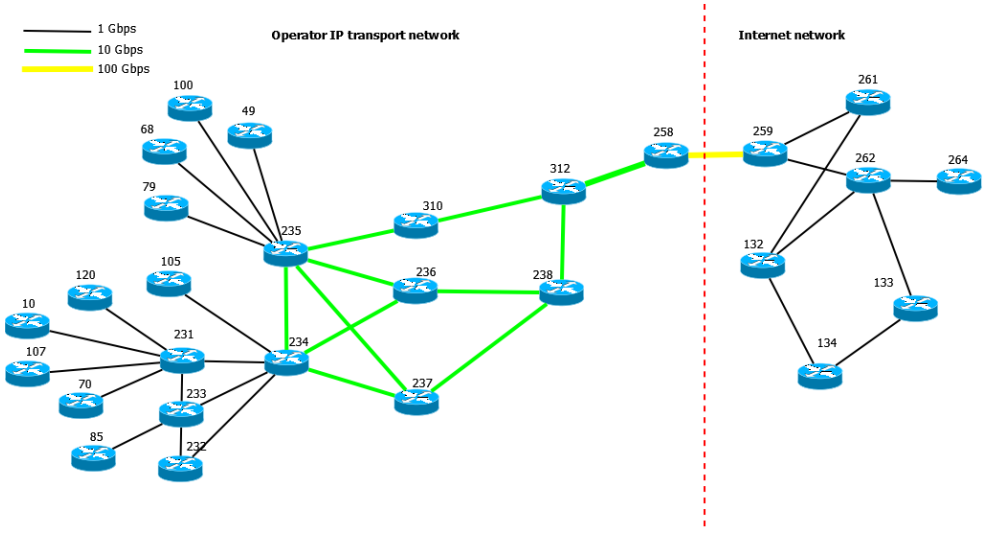

Figure 54: Network Topology

As it can be seen in Figure 54, there are two parts of network that we have modelled, namely Operator IP transport network and Internet network. Those two networks are connected each other through router 258 (gateway on the operator side) and router 259 (on the internet side), with the link speed between them equals to 100 Gbps. The operator IP transport network is the IP network which is deployed in the middle of EPS components as depicted in the Figure 53. In the experiments, the operator IP transport network is divided into two parts, namely IP backbone network, that has 10 Gbps link speed, and IP access network (where eNodeB is connected), that has 1 Gbps link speed. The details of parameters used to deploy the network topology can be seen on the Table 4.

4.2.1.1 Enhancements in LTE Module

In this assignment we made several enhancements in LENA in order to enable it to support the CCNx protocol and to model the interaction between eNodeBs (both source and target eNodeBs) and Virtualization Controlling Platform (VCP). The source codes within the LTE module that we enhanced can be found in [71], and the guideline of how to use the source codes for running our experiments is explained in Appendix B.

The codes that we enhanced in the LTE module are the following ones:

- **epc-enb-application.cc**

  In this file, we enhanced the functions used for processing packets receiving from the air interface (*RecvFromLteSocket()*) and S1-U interface (*RecvFromS1uSocket()*). We

implemented the CCNx protocol in those two functions in order to process Interest and Data packets (processing of Interest and Data packets is based on subsection 3.1.3). *RecvFromLteSocket()* is the function which is used to process Interest packet received from the air interface, while *RecvFromS1uSocket()* is the function which is used to process Data packet received from SGW/PGW. Moreover, in order to enhance seamless service continuity supported by LENA for downlink, we implemented a buffer (placed in the *RecvFromS1uSocket()* function) which is used to buffer a packet received by eNodeB during handover preparation state of a UE. In addition, we also created a new function which is used to send a message to the VCP node, namely *SendToCloudComponent()*.

- **epc-sgw-pgw-application.cc**

  In this file, we enhanced the functions used for processing packets receiving from S1-U interface (*RecvFromS1uSocket()*) and internet (*RecvFromTuneDevice()*). We implemented the CCNx protocol in those two functions in order to process Interest and Data packets (processing of Interest and Data packets is based on subsection 3.1.3). *RecvFromS1uSocket ()* is the function which is used to process Interest packet received from S1-U interface, while *RecvFromTuneDevice ()* is the function which is used to process Data packet received from the internet.

- **epc-x2.cc**

  In this file we made several modifications. We changed the UDP port numbers for both X2-U and X2-C links. This modification is needed in order to establish an X2 link via IP router network that we deployed in the middle of EPS components. Moreover we also enhanced *RecvFromX2cSocket(),* the function which is used to process signalling packets received from X2 link. Within that function, we implemented a piece of code used for sending a message to the VCP when there is a handover request.

  The other functions within epc-x2.cc that we enhanced are *DoSendHandoverRequestAck(), DoSendUeData(),* and *RecvFromX2cSocket()*. Those three functions are organized together in order to simulate a communication among a target data centre, a source data centre, and VCP.

  Furthermore, within this file we also implemented a new function, called *SendVm(),* which is used for migrating VM from the source data centre (source eNodeB with CCNx functionality) to the target data centre (target eNodeB with CCNx functionality).

- **lte-enb-rrc.cc**

  In this file we enhanced a function, namely *DoRecvUeData(),* which is called when a target eNodeB receives Data packet forwarded by a source eNodeB through an X2 link. Within that function we implemented the CCNx protocol used to process Data packet received through an

X2 link. Besides, we also enhanced *RecvHandoverRequestAck()* function (which is called when the source eNodeB receives a handover request acknowledgement) in order to enhance a seamless service continuity supported in LENA for downlink.

- **epc-helper.cc**

  In this file, we implemented overloading to the constructor of EpcHelper class (*EpcHelper (Ptr <Node> pgw)*) in order to deploy IP router network in the middle of EPS components. Moreover we enhanced *AddEnB()* function for the same purpose. *AddEnb()* is a function that will be called when we want to install an eNodeB device. In addition we also enhanced an *AddX2Interface()* function in order to make an X2 link be established through an IP router network.

Moreover, there is also a new code that we included in the LTE module, namely lte-ccn-common.cc. This file is used to create a data structure of PIT and CS tables implemented in epc-enb-application.cc, epc-sgw-pgw-application, and lte-enb-rrc.cc (see [71]).

4.2.1.2 Packet Formats

In the experiments, we generated several packets (see Figure 68 and Figure 69) such as Interest, Data, and some messages exchanged between VCP and eNodeBs (both source and target eNodeB). In this subsection we describe the format of those packets.

The format of CCN packets (Interest and Data packets) are shown in the Figure 55 and Figure 56.

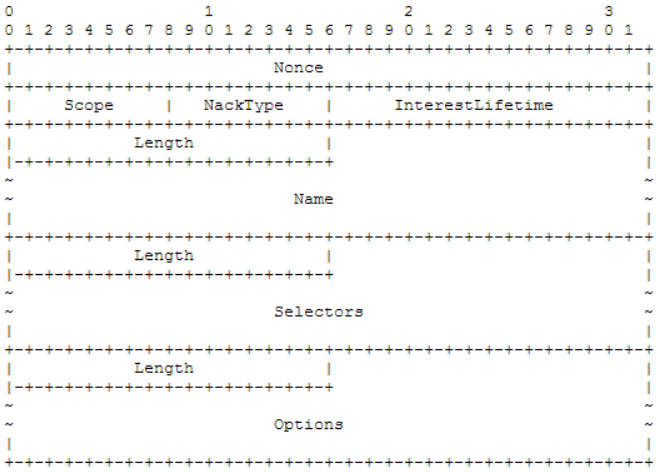

Figure 55: CCN Interest packet format in the simulations, copied from [61]

As it can be seen in the Figure 55, there several information fields in the Interest packet. *Nonce* is used to detect and prevent duplicates received over different paths or interfaces. *Scope* is used to limit propagation of Interest packet. *NackType* contains information about a type of NACK.

InterestLifeTime contains information of Interest's lifetime. *Name* contains information of content name. *Selector* contains information about selection of Data (Content Object) which is used to select a Data packet that will be delivered to a requester. The last field is *Options* which is an optional field.

Figure 56 shows the format of Data packet used in the simulations. As we can see, there are three main fields in the Data packet, namelu *Name*, *Content*, and *Signature*. *Signature* is used for authentication. *Name* is used to indicate the name of content (Data). *Content* consists of two main information, namely *Content Info* and *Content Data*. *Content Info* contains *Freshness* (used to specify how long the content is considered valid) and *Timestamp* (used to specify content generation time).

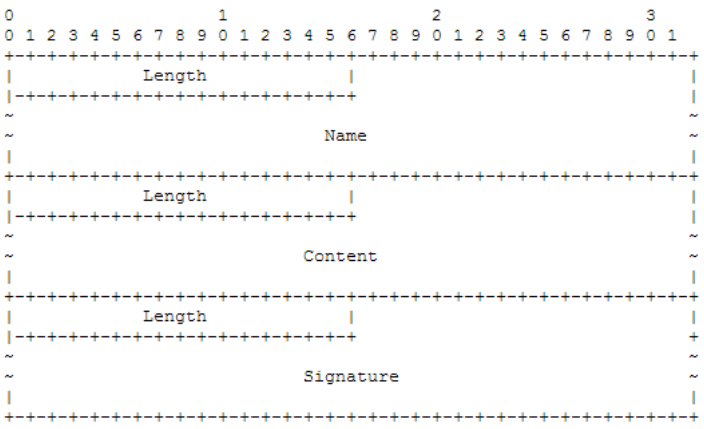

Figure 56: CCN Data packet format in the simulations, copied from [61]

The LTE protocol stack which is considered in the simulations is the LTE user plane protocol stack shown in Figure 37. Since we assumed that UE is aware of CCN, the CCNx protocol layer will be a part of the Application layer. Therefore before being transmitted, the Interest and Data packet will be encapsulated by the layers below the Application layer.

Moreover, there are several signalling messages that we implemented in the experiments. As we have explained in previous chapter (3.3.1 and 3.3.2), when there is handover, the target data centre (eNodeB with CCNx functionality) will send a message to MSM (in the simulations, the node that receives this message is the VCP node (see Figure 68 or Figure 69)). Subsequently, the ICN Manager will send a message used to configure the PIT table in the CCN router functionality (both in the source and target data centres) in order to forward the requested content to the mobile user. In the simulations, we modelled the format of these messages as shown in the Figure 57, Figure 58. In addition, in the simulations, the used transport protocol to transmit these messages is UDP.

In the rest part of this section we show the messages that we implemented in the simulation experiments and are used in the information flow diagrams depicted in Figure 68 and Figure 69.

Figure 57 shows the structure of *Client Info* message used by the target eNodeB for sending client information to VCP when handover occurs (see step 2 in the Figure 68 or Figure 69). The *Packet Type*

field is used by the VCP to recognize the type of message. The *IMSI* field is the IMSI (International Mobile Subscriber Identity) of the UE doing handover, and the *IP address* field is the IP address of the UE.

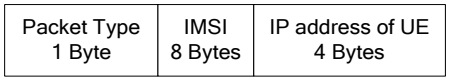

Figure 57: Client Info message

Figure 58 shows the structure of *Config PIT* message sent by VCP used for configuring a PIT table of eNodeB (see step 11 and 12 in the Figure 68 or step 11 and 18 in the Figure 69). The *Packet Type* field is used to identify the message type. The *IP address* field is the IP address of UE doing handover. The *UDP port* field is the UDP port used by the UE for receiving packets. The *RBID* field is the Radio Bearer ID used by the UE. The *RNTI* is the Radio Network Temporary Identifier used by the UE in a new location (when it is connected to the target eNodeB). The *length* field contains information of the length of Interest name.

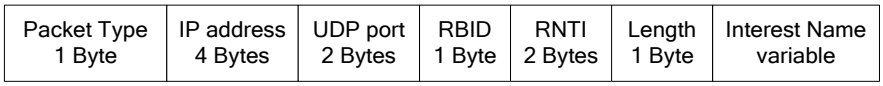

Figure 58: Config PIT message

When a VM needs to be migrated, VCP will send a VM migration request message (see step 12 in Figure 69) to the target data centre (target eNodeB with CCNx functionality). Figure 59 shows the structure of this message. *Packet Type* field is used to identify the type of message. *VM ID* is the identity of the VM which is need to be migrated, while *IP address* field contains information about the IP address of source data centre that has the requested VM.

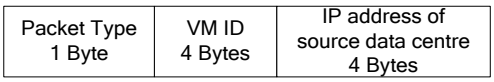

Figure 59: VM migration request message

Whenever VM migration has been completed, the target data centre will send this information to the VCP (see step 15 in the Figure 69). Figure 60 shows the structure of that message. *Packet Type* field is used to identify the packet. *VM ID* is the identity of the VM that has been migrated. *Info Code* is used to show the result of VM migration (successful or failed).

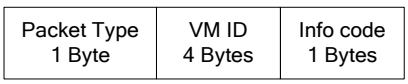

Figure 60: Completion of VM migration info message

After receiving the information that VM has been migrated successfully, the VCP sends a *Start VM* command used to start the VM in the target data centre (see step 16 in Figure 69). Besides the *Packet Type*, this message contains information about *VM ID* of the VM that will be started, *Time* and *Date* when the VM has to be started.

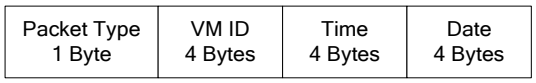

Figure 61: Start VM command message

The last message that we implemented in the experiments is the response message (ACK) for the *Start VM* (see step 17 in Figure 69). The format of this message is the same as the message format shown in Figure 60.

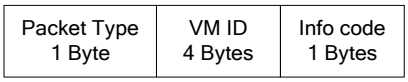

Figure 62: Response message for Start VM command

4.2.2 Traffic Generators

There are four types of traffic that are implemented in the simulation experiments, such as background traffic on the wireless link, background traffic on the wired link, CCN traffic, and signalling traffic. The following sections describe how those four types of traffic are implemented.

4.2.2.1 Background Traffic on the Wireless Link

On the wireless link (radio access network) we generated background traffic with composition 70% of the total traffic (the other 30% is CCN traffic). There are three types of traffic models that we used in the experiments, namely Voice over IP (VoIP), video streaming, and gamming. The composition of those three types of traffic models is shown on Table 2.

Table 2: Composition of traffic

Traffic Model	Percentage of Traffic
VoIP	30%
Video streaming	20%
Gamming	20%

The code that we used for generating those traffic models is the code implemented by T.G. Pham in [69] and A.D. Nguyen in [70] based on traffic models specified by NGNM (Next Generation Mobile Network) Alliance [65]. The code (general-udp-client.cc) is located in the gen-udp module, and can be found in [72].

4.2.2.2 Background Traffic on the Wired Link

Background traffic on the wired link was generated only on the operator IP transport network. It was generated using the Poisson Pareto Burst Process (PPBP) application developed in [73]. This application is able to generate an accurate network traffic generator that matches statistical properties of real IP networks. In the PPBP application, there are several parameters that need to be considered in order to generate background traffic that can simulate internet traffic in a realistic way, such as the Hurst parameter, *H*, the mean length of a burst, T_{on}, the bit rate of each individual burst, *r* and the arrival rate of burst, λ_p.

The PPBP application can generate multiple bursts that will arrive according to a Poisson process with rate λ_p. The length of bursts is varied based on Pareto distribution characterized by Hurst parameter, typically between 0.5 and 0.9, and the mean length of a burst. In the PPBP model, a burst is modelled as a flow with a constant bit rate.

The average number of active bursts, *E[n]*, can be calculated using the Little's law as follow:

$$E[n] = T_{on} x \lambda_p \qquad (1)$$

Each burst on the PPBP application generates a flow with a constant bit rate, therefore the formula used for calculating the overall rate of the traffic, λ:

$$\lambda = T_{on}\ x\ \lambda_p\ x\ r \qquad (2)$$

4.2.2.3 CCN Traffic

The CCN traffic that we modelled in the experiments is video streaming traffic. The bit rate that we used for generating video streaming traffic is 464 Kbps as recommended in [74]. The size of packet (without headers) that will be generated by a server when it receives an Interest is 1316 Bytes, see e.g., [75]. Since the CCN application that we used is a video streaming application, we consider that the traffic generated by such an application is CBR (Constant Bit Rate) traffic. In order to generate such type of traffic (for each UE) we used a CBR generator that generates the packets with a bit rate of 464Kbps. For the 1316 bytes size packets, in order to achieve this bit rate we used an inter-packet generation time of 23 ms. In particular, the Interest packet itself will be encapsulated by using UDP and IP headers, and sent periodically to the source eNodeB that supports the CCNx functionality.

In order to generate CCN traffic (Interest and Data packets), we enhanced the *UdpEchoClient* and *UdpEchoServer* applications which are located in the applications module of ns-3. By leveraging some classes provided in ndnSIM [61] module (*Name*, *Interest*, and *ContentObject*) we generated Interest and Data (Content Object) and inserted them as payloads of UDP packets generated by *UdpEchoClient* and *UdpEchoServer* applications. In our experiments, we installed the *UdpEchoClient* application in a UE in order to generate Interest packets; and installed the *UdpEchoServer* application

in a remote host (server) in order to generate Data packets. The codes that we enhanced in order to generate CCN traffic are the following ones, see [73]:

- **udp-echo-client.cc**

  This file is used to generate *UdpEchoClient* application. In this file, we enhanced the *Send()* function which is used to generate and send a packet to a server. Within this function, we generated an Interest packet by using *Name* and *Interest* classes provided by the ndnSIM module. Afterwards, we encapsulated the Interest packet using UDP and IP headers. By doing this, the format of packet sent to the server will be IP+UDP+Interest. Besides, we also enhances the *HandleRead()* function which is used to read a packet received by the application. Within this file we implemented a decoder used for reading a Data (content object) packet received by the application. We did it by leveraging *ContentObject* class provided in the ndnSIM module. The format of packet received by the application is IP+UDP+Content Object Header+Content.

- **udp-echo-server.cc**

  This file is used to generate *UdpEchoServer* application. In this file, we enhanced the *HandleRead()* function which is used to read a packet received from a client. Within this function, the received Interest packet will be decoded by using the *Interest* class provided by the ndnSIM module in order to know the content name requested by a client. Afterwards, the requested Data (Content Object Header+Content) is generated by leveraging *ContentObject* and *Name* classes provided by ndnSIM module. This Data packet will be encapsulated using IP and UDP headers before it is sent to the client (IP+UDP+Content Object Header+Content).

The enhancement of the codes can be found in [76].

4.2.2.4 Signalling Traffic

As explained in section 3.3.1 and 3.3.2, there are several messages used for signalling purposes when handover between two eNodeBs hosted in different data centres is triggered. See Figure 57 - Figure 62 for the messages that we implemented. In order to simulate a communication between VCP and data centres (both the source and target eNodeBs), we created a new server application, namely *UdpEchoServerApplication2*, which is installed at the VCP node. When handover occurs, the VCP node will communicate with both source and target data centres (eNodeBs with CCNx functionality) by sending signalling messages for completing the handover procedures. The source code of this implementation can be found in [76].

4.2.3 X2 Handover Simulation

There are two types of X2 handover simulations that we performed, which focuses on:

- Mobility and content migration support
- Mobility, VM (container) and content migration support

In the simulations, we implemented option 1 (see subsection 3.2.1) as the solution for integrating CCN concept into LTE system. Basically option 1 and option 5 are the same, except the fact that in option 1, the CDN repository is not used.

In the experiments, we simulated the handover performed by several users moving from one location (source data centre, where the source eNodeB with CCN functionality is hosted) to another location (target data centre, where the target eNodeB with CCN functionality is hosted) served by different data centres without using MOBaaS.

We modelled the user mobility by using three types of vehicular speeds (e.g. car, bus, and truck) in urban area. Since X2 handover in the current implementation of LENA can be triggered only by time, not by a UE's position, we leveraged a distribution of residence time of those vehicles in order to trigger handover of UEs attached to the source-eNodeB. The distribution of time that we generated is based on the calculation of residence time in one cell [66] using the following formula:

$$E[T] = \frac{8R.E[\frac{1}{V}]}{3\pi} \qquad (3)$$

E[T] or residence time is defined as the length of time a mobile terminal resides in the cell where the call originated before crossing the cell boundary. R is the cell radius and V is the speed of the mobile user in the cell.

As we can see on the formula above, one of the parameters required for calculating the residence time is the speed of mobile user (V). In the simulation, we modelled the speed of mobile users (who are using different types of vehicles), see Table 3, provided in [67].

Table 3: Traffic in urban area, based on [64]

Vehicle Class	Average Speed (Km/h)	Estimated Standard Deviation
Cars	49.9	9.7
Buses and coaches	45.1	8.91
2 axle trucks	49.9	9.25

4.2.4 Simulation Parameters

The following subsections explain the parameters that we set up in the simulations.

4.2.4.1 Parameters in IP Transport Networks

On the operator side (see Figure 54), there are two types of IP transport networks, namely IP backbone network (which consists of backbone routers, marked by the green line) and the IP wired

access nework (which consists of access routers, marked by the black line). For the link speed of the IP backbone network, we set it as 10 Gbps, and for the link speed of the IP access network, we set it as 1 Gbps. On the internet network side, we set the link speed between two routers as 1 Gbps. The parameters implemented in the IP core network are summarized in Table 4.

Table 4: Parameters in IP transport network

Parameter	Value
IP backbone link speed (operator side)	10 Gbps
IP access network link speed (operator side)	1 Gbps
Internet network link speed	1 Gbps
Gateway router link speed (to internet)	100 Gbps
Maximum Transmission Unit (both on the operator and internet sides)	1500 Bytes
Queue scheme	Drop tail
Buffer size of router	1 Gbps : 3125000 bit 10 Gbps : 31250000 bit 100 Gbps : 312500000 bit
Background traffic	80%

One of parameters that we set up in the experiments is the buffer size of router. As it can be seen in Table 2 we have three different values of buffer size since in the IP core network (as depicted in Figure 54), we implemented three types of link speed on the links connecting the routers. For calculating a router buffer size, we used the following formula that we obtained from [64]:

$$Buffer\ size\ (bit) = \frac{CxRTT}{10} \qquad (4)$$

C is the link speed, and RTT is the round trip time of the flow. In our calculation, we specified RTT as 250 ms (the minimum round trip time worth's of buffering that routers should provide [64]).

4.2.4.2 Parameters in LTE Systems

The parameters in LTE systems that we implemented in the simulations are summarized in Table 5.

Table 5: Parameters in LTE systems

Parameter	Value
Uplink bandwidth	5 MHz (25 Resource Blocks)
Downlink bandwidth	5 MHz (25 Resource Blocks)
Uplink EARFCN	21100 band 7 (eNodeB 1) 21150 band 7 (eNodeB 2)
Downlink EARFCN	3100 band 7 (eNodeB 1) 3150 band 7 (eNodeB 2)
CQI generation period	10 ms
Transmission mode	MIMO 2x2
UE transmission power	26 dBm
UE noise figure	5 dB
eNodeB transmission power	49 dB
eNodeB noise figure	5 dB
MAC scheduler	Proportional Fair (PF)
Cell radius	1 Km

4.2.5 Confidence Interval

In order to guarantee the reliability of the collected performance results, the experiments that we performed are repeated several times using different random seed. In the experiments, we used a confidence interval of 95% and the confidence interval should be less than 5% of the sample mean value.

Since the number of samples in the experiments is less than 30 samples, the following formula is used to calculate confidence interval:

$$Confidence\ Interval = x \pm t_{\frac{\alpha}{2}}\left(\frac{\sigma}{\sqrt{n}}\right) \qquad (5)$$

Where, x is the sample mean, n is the number of sample, σ is the standard deviation, and $t_{\frac{\alpha}{2}}$ is the value corresponding to $\frac{\alpha}{2}$ in the t table.

4.2.6 Utilization of the Wired and Wireless Links

In the experiments, we tried to simulate traffic in rush hour, where the network resources are utilized around 80% (both in radio access network and IP core network). In order to reach 80% cell utilization for the uplink direction on the radio access network, we performed several measurements (using a traffic mix of 70% background traffic and 30% CCN traffic) to measure the maximum uplink traffic load that could be served by a single eNodeB (using configuration parameters shown on Table 3), and we got the result as shown in Figure 63.

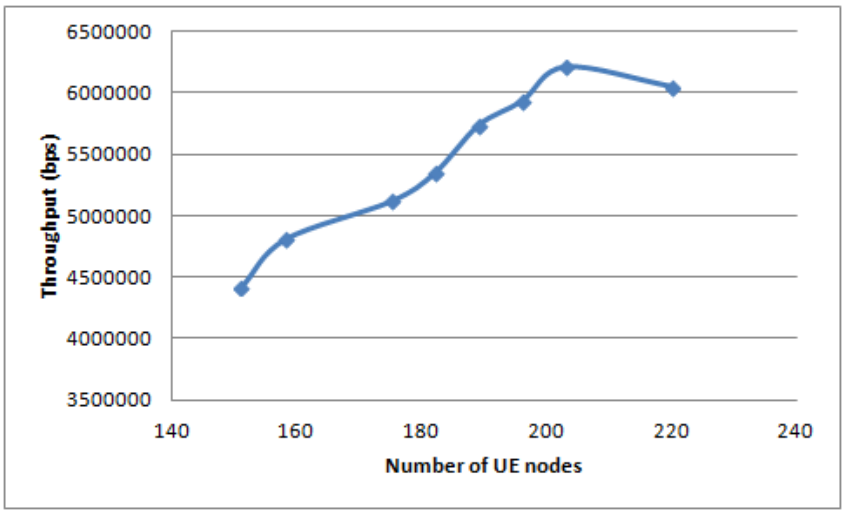

Figure 63: Throughput vs number of UE nodes

The maximum throughput that can be reached using the traffic mix is 6.207 Mbps with the traffic load 6.225 Mbps (total number of UE nodes is 203). As it can be seen in Figure 63, the total throughput will decrease when the traffic load is further increased (by increasing the number of UE nodes to be

more than 203 UE nodes). In order to reach 80% cell utilization on the radio access network (wireless link), the uplink traffic load that we generated in the simulations is around 4.979 Mbps (80% of max. traffic load). The total traffic load that we generated is a traffic mix (70% background traffic and 30% CCN traffic). The total number of UE nodes that we used in order to generate traffic that can utilize 80% of the resource in the air interface link is 158 UE nodes (60 CCN nodes and 98 background nodes).

Moreover, in order to reach 80% utilization in the Operator's IP fixed transport network, we leveraged on the PPBP application that can generate background traffic and occupy 80% of the link capacity between two network entities (e.g. between router and router, between eNodeB and router, between SGW/PGW and router, and between VCP and router). As we have explained in the subsection 4.2.2.1, there are several parameters that need to be set in order to generate traffic, such as Hurst parameter, *H*, the mean length of a burst, T_{on}, the bit rate of each individual burst, *r* and the arrival rate of burst, λ_p.

The average number of active bursts that we want to generate can be calculated using formula (1). Moreover, the overall rate of the traffic, λ, can be calculated using formula (2).

For 1 Gbps and 10 Gbps links, the overall bit rate (λ) that we expected respectively is 0.8 Gbps and 8 Gbps; therefore by using that formula, we set the parameters of PPBP application (for both 1 Gbps and 10 Gbps links) with the values as shown on the Table 6. The value of the Hurst parameter, *H*, that we set for both 1 Gbps and 10 Gbps links is 0.7 as recommended in [73].

Table 6: PPBP Parameters

Transmission Speed	H	T_{on}	r	λ_p
1 Gbps	0.7	0.02	20 Mbps	2000
10 Gbps	0.7	0.02	200 Mbps	2000

As it can be seen in Table 6, for both 1 Gbps and 10 Gbps links, we generated 40 bursts. Each burst represents a traffic aggregator that generates 20 Mbps and 200 Mbps traffic respectively for 1 Gbps and 10 Gbps links.

Figure 64 and Figure 65 show respectively the fluctuation of traffic load during 70 seconds of simulation for 1 Gbps link and 10 Gbps link. The traffic load always fluctuates during the simulation time, but it never reaches the maximum capacity of the link (both for 1 Gbps and 10 Gbps link).

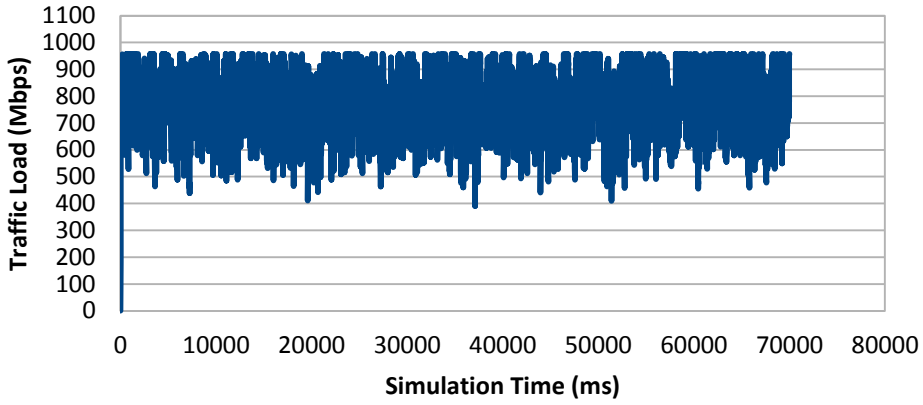

Figure 64: The fluctuation of traffic load on 1 Gbps link

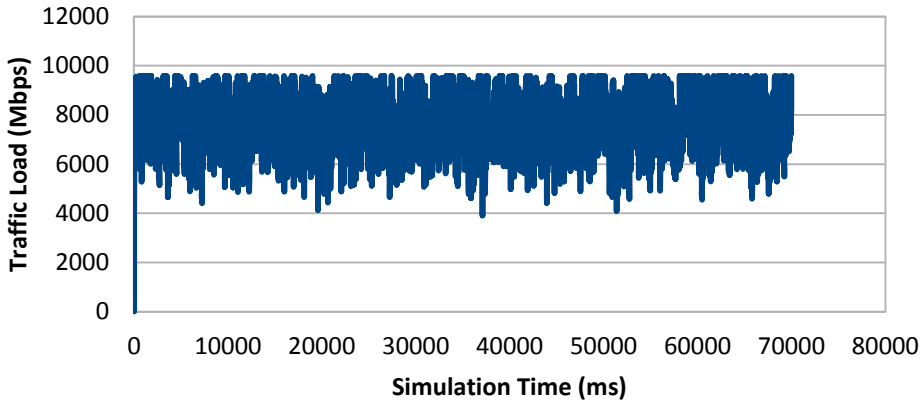

Figure 65: The fluctuation of traffic load on 10 Gbps link

Figure 66 and Figure 67 show respectively the graphs of overall bit rate (traffic load) during 70 seconds of simulation for 1 Gbps link and 10 Gbps link. From the graphs, it can be seen that the total bit rate (traffic load) for both 1 Gbps and 10 Gbps links after 10 seconds simulation remains stable above 75% of the total capacity of the link. Therefore, the simulation results will be collected after this initiation time of 10s.

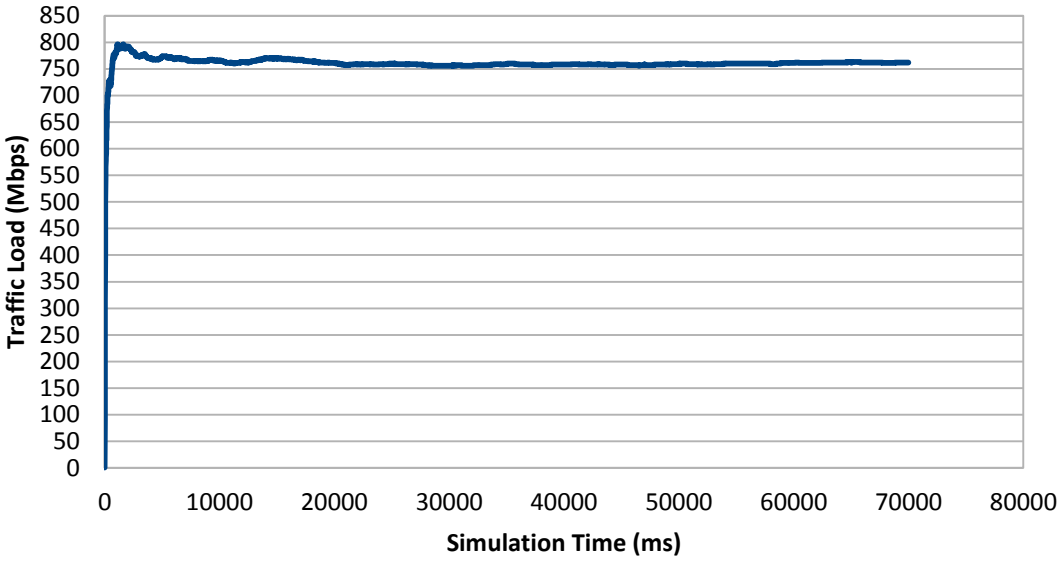

Figure 66: The total traffic load on 1 Gbps link

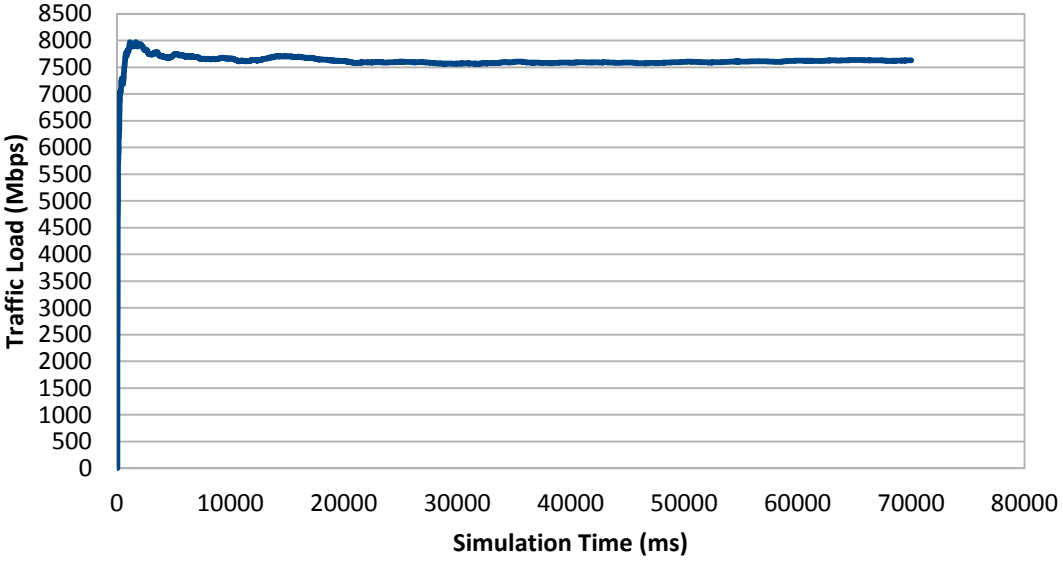

Figure 67: The total traffic load on 10 Gbps link

4.3 Performance Metrics

This section describes the performance metrics applied in the simulation experiments. These are:

- Average RTT for CCN Interest/Data packets when content and/or VM migration occurs

  In CCN, each request (Interest) will be satisfied by one response (Data/Content Object). The measured latency is the RTT (Round Trip Time) delay of the CCN request-response packets

when CCN content and/or VMs need to be migrated. When a user is moving to a new location, content or data requested by the user which is not received in the previous location can be obtained in the new location through the LTE X2 link established between the two eNodeBs. We define latency as the round trip time (RTT) needed to receive the CCN data (response) at the new location of user, after sending the CCN Interest (request) from the previous location.

- Cumulative Distribution Function (CDF) for RTT for Interest/Data packets when content and/or VM migration occurs

  This metric is the CDF of RTT of CCN Interest/Data packets and is used to observe the maximum RTT value that can be observed during the simulation experiments, when content and/or VM migration occurs.

- Average RTT in receiving CCN Data packets when content and/or VM migration does not occur

  This metric is defined in the same way as the previously described average RTT metric, with the difference that now it is considered the content and/or VM is not migrated from one data centre to another.

- Cumulative Distribution Function (CDF) for RTT for Interest/Data packets when content and/or VM migration does not occur occurs

  This metric is defined in the same way as the previously described CDF for RTT metric, with the difference that now it is considered the content and/or VM is not migrated from one data centre to another.

- Throughput of CCN Data packets when content and/or VM migration occurs

  In the experiment, we defined throughput as the number of CCN data (response) packets correctly received by CCN mobile users divided by simulation time, when content and/or VM is migrated from one data centre to another. For calculating throughput of the CCN Data packets, we implemented a tracer that we installed in the application layer on the UE side (inside *UdpEchoClient* application). This tracer will record the number of packet sent and received by a UE and record the transmitting and receiving time of packets. By looking to this tracer, we can calculate the throughput of CCN Data packets. By calculating the throughput, we can see the comparison between throughput and traffic load in each experiment.

- Throughput of CCN Data packets when content and/or VM migration does not occur

This metric is defined in the same way as the previously described Throughput of CCN Data packets metric, with the difference that now it is considered the content and/or VM is not migrated from one data centre to another.

- Maximum RTT of Interest/Data packets when VM and Content Migration Occur

  This metric is used only for the set of experiments that focuses on VM (container) and Content Migration Support". The maximum RTT is representing the maximum RTT values obtained during the performed simulation experiments.

4.4 Experiment Scenarios

As explained in section 4.2.3, there are two sets of experiments that we performed. The first set of experiments focuses on "Content Migration Support", while the second one focuses on "VM (container) and Content Migration Support". From both sets of experiments we investigated the impact of varying different parameters to (1) average RTT of CCN Interest/Data packets when content and/or VM migration occurs, (2) the CDF of the RTT of CCN Interest/Data packets when content and/or VM migration occurs, (3) average RTT of CCN Interest/Data packets when content and/or VM migration does not occur, (4) the CDF of the RTT of CCN Interest/Data packets when content and/or VM migration does not occur, (5) throughput of CCN Data packets, when content and/or VM migration occurs, (6) throughput of CCN Data packets, when content and/or VM migration does not occur and (7) maximum RTT of Interest/Data packets when VM and content migration occur.

4.4.1 Definition of the Parameters to be Varied

The following sections include the details that show how the parameters are varied in order to investigate the impact on the selected performance metrics.

4.4.1.1 Distance between source and target data centers running eNodeBs

In cloud based LTE systems, a mobile UE can move from one eNodeB, that runs on the source data center, to another eNodeB, that runs on the target data center. When a UE moves, then a X2 handover occurs. The CCN response (Data) for UE‘s CCN request that has not yet been received in the previous location (previous eNodeB; source data centre) can be forwarded to the UE in the new location (target eNodeB; target data center) through the X2 link. The time needed to forward the CCN response (Data) through X2 link can vary since there are some routers in between two (source and target) data centers (running the eNodeBs). The number of hops can provide some impact to latency.

In the experiments, the distance (number of hops) between source and target data centres, which are represented by source and target eNodeBs, is varied. As explained in section 4.2.1, we implemented a

topology of router in the middle of EPS components (eNodeB and SGW/PGW) and VCP. By using that topology, we varied the distance between source and target eNodeBs by increasing the number of hop, started from 1 hop, 2 hops, 4 hops, 6 hops, 8 hops to 10 hops. In order to increase the number of hops, we specified the different route used by the source eNodeB to communicate with the target eNodeB. The detail figure of the route (for each number of hops between two eNodeBs) that we have specified can be found in Appendix A. Moreover, the codes that we used to model the distances between source and target data centres can be found in [77].

4.4.1.2 Location of Virtualization Controlling Platform

The Distance (number of hops) between the Virtualization Controlling Platform (VCP) and both source and target data centres running eNodeBs can affect the time needed to forward user's data from the source data centre to the target data centre, due to the fact that when a handover occurs, both source and target data centers (running source and target eNodeBs respectively) will communicate with the VCP. The target eNodeB will send handover information to the VCP, and VCP will respond to this information by sending messages, to both source and target eNodeBs, that contain information on how to configure their PIT. The source eNodeB will not be able to forward a user's Data packet to the target eNodeB before it receives this information (used to configure PIT) from the VCP. Similarly, the target eNodeB will not be able to forward the user's Data packet received from the source eNodeB to the user if it has not received the information (used to configure its PIT) sent by VCP.

In the experiments, we changed the location of VCP node by hosting it into three different data centres as follows:

- the VCP is located in the same data centre as where the S-GW/P-GW is located
- the VCP is located in the same data centre as where the target eNodeB is located
- the VCP is located in the same data centre as where the source eNodeB is located

The detail configuration of the used topology can be found in Appendix A. Moreover, the codes that we used to model this topology can be found in [77].

4.4.1.3 Size of Virtual Machine

One of parameters that can affect how long a VM migration will take place is the size of VM. When a CCN mobile user is moving from one location to another, and in the new location there is no VM that can support the CCN functionality, a VM migration need to be performed in order to maintain service continuity.

In the experiment, we modelled a VM migration, by transferring a copy of VM (which is represented by a number of chunks) from the source data centre (where the source eNodeB is hosted) to the target data centre (where the target eNodeB is hosted). The VM is fragmented into several chunks (the

number of chunks depends on the size of VM) before it is sent to the target data centre. The source data centre will transfer all chunks of VM to the target data centre after receiving an Interest (request) for VM sent by the target data centre. The target eNodeB (hosted in the target data centre) will not be able to forward the user's Data packet received from the source eNodeB and serve the mobile CCN user before the migration process of VM has been completed (the migrated VM in the target data centre has to be started/resumed).

There are two sizes of VM that we implemented in the experiment, 128 MB and 256 MB. As shown in Table 7, the chunk size that we used is 1280 Bytes; the transmission speed used to send the chunks of VM is 50 Mbps; while the number of chunks depends on the size of VM. The selected transmission speed represents 5% of the capacity of the slowest wired link used in the simulation experiments, which can be considered to be a realistic value. However, the operator can use any value for this generation/transmission speed, as long as the communication links do not become congested.

Table 7: Sizes of VM used in the experiment

VM Size	Chunk Size	Number of chunks	Transmission speed
128 MB	1280 Bytes	100000	50 Mbps
256 MB	1280 Bytes	200000	50 Mbps

Moreover the source codes that we have implemented to model VM migration using different VM sizes can be found in [71].

4.4.2 Content migration support

The experiment that we performed for content migration support is based on the information flow diagram in section 3.3.1.1 (see Figure 46). Figure 68 shows the flow diagram that we implemented in order to simulate content migration support. This information flow diagram is the simplified information flow diagram depicted in Figure 46.

In this experiment, the target eNodeB is already aware of CCN, so that there is no need to migrate VM (there is only content migration). Moreover, the cloud components required to support content migration (MSM, SO, ICN Manager) are assumed to be placed in the same data centre and the processing time consumed by those components are considered to be zero. Therefore in this experiment, those cloud components are modelled within a single entity, named as Virtualization Controlling Platform (VCP).

In this set of experiments, we deployed two eNodeBs (source and target eNodeB), one SGW/PGW node, one VCP node, and one server. All of CCN users connected to the LTE network are accessing a video streaming service (provided by the server) and watching the same video. There are 60 CCN users who are connected to the source eNodeB, while there are 9 CCN users who are connected to the

target eNodeB. The 51 CCN users connected to the source eNodeB will do handover (based on the residence time of each CCN user in one cell) and move to the new location served by the target eNodeB, while the other 9 CCN users stay connected to the source eNodeB. At the end, there are 60 CCN users connected to the target eNodeB, while there are 9 CCN users connected to the source eNodeB.

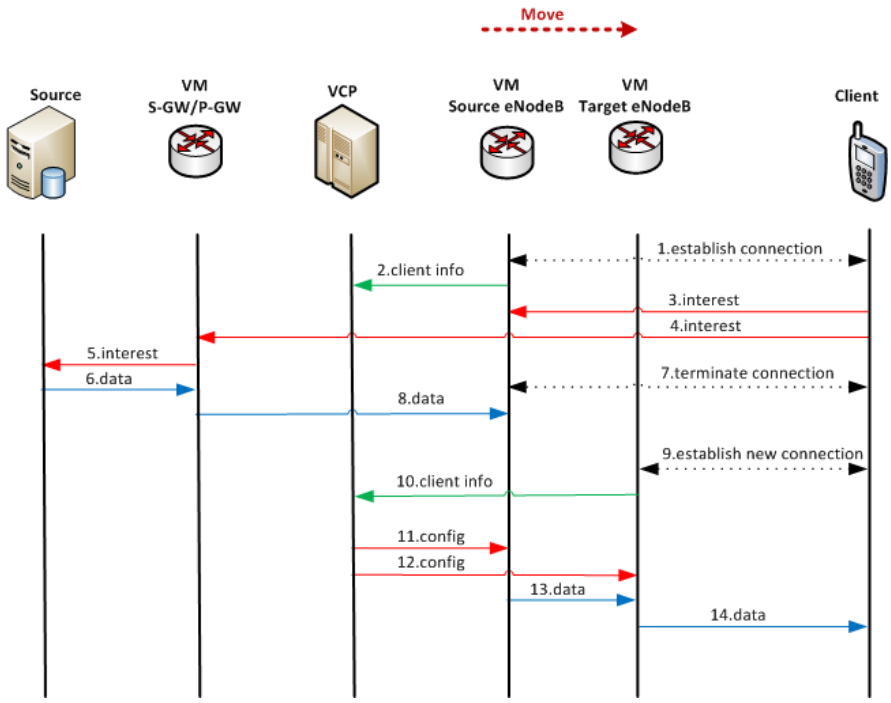

Figure 68: Information flow for the simulation of content migration support

We performed several experiments by varying the distance between source and target data centres running eNodeBs (see section 4.4.1.1) and location of VCP (see section 4.4.1.2). Those experiments are divided into three sets of experiments based on the position of VCP (see Figure 81 to Figure 86), such as follows:

- position 1: VCP and SGW/PGW are hosted in the same data centre.
- position 2: VCP and target eNodeB (T-eNodeB) are hosted in the same data centre
- position 3: VCP and source eNodeB (S-eNodeB) are hosted in the same data centre

In each set of experiment, the distance (number of hops) between source and target data centres running eNodeBs is varied. The numbers of hops that we simulated are 1 hop, 2 hops, 4 hops, 6 hops, 8 hops and 10 hops. Moreover, in performing these sets of experiments, we set the total utilization for both wireless and wired links to be ~80% (see section 4.2.6). The detail of topologies (including the location of network elements and the routing scenarios) used in these sets of experiments can be seen in Appendix A.

The following subsections list the performace metrics that we used in the set of experiments that focuses on content migration support.

4.4.2.1 Average RTT of Interest/Data Packets when Content Migration Occurs

As explained in section 4.3, the measured latency is the RTT (Round Trip Time) delay of the CCN request-response packets when CCN content needs to be migrated from the source data centre to the target data centre. In each set of experiments (based on the location of VCP), we measured and compared the average latency in receiving CCN Data packets for each variation of distance (1 hop, 2 hops, 4 hops, 6 hops, 8 hops and 10 hops) between the source and target data centre running eNodeBs.

4.4.2.2 Average RTT of Interest/Data Packets when Content Migration Does Not Occur

This set of experiments is similar to the one mentioned in section 4.4.2.1, with the difference that now content is not migrated from one data centre to another. The measurements of this performance metric are accomplished on the CCN users connected to the target eNodeB (hosted in the target data centre) that are requesting and receiving content.

4.4.2.3 CDF of RTT of Interest/Data Packets when Content Migration Occurs

The CDF of the RTT of the Interest/Data packets, when content migration does occur, see section 4.3, is used to observe the maximum value of RTT obtained during the accomplished simulation experiments. In each set of experiments (based on the location of VCP), we calculated and compared the CDF of RTT of the Interst/Data packets when content migration occurs for each variation of distance (1 hop, 2 hops, 4 hops, 6 hops, 8 hops and 10 hops) between the source and target data centre running eNodeBs.

4.4.2.4 CDF of RTT of Interest/Data Packets when Content Migration Does Not Occur

This set of experiments is similar to the one mentioned in section 4.4.2.3, with the difference that now content is not migrated from one data centre to another. The measurements of this performance metric are accomplished on the CCN users connected to the target eNodeB (hosted in the target data centre) that are requesting and receiving content.

4.4.2.5 Throughput of CCN Data Packets when Content Migration Occurs

According to section 4.3, throughput is defined as the number of CCN data (response) packets correctly received by all CCN mobile users divided by simulation time. In each set of experiments (based on the location of VCP), we measured and compared the throughput of CCN Data packets, when content migration occurs, for each variation of distance (1 hop, 2 hops, 4 hops, 6 hops, 8 hops and 10 hops) between the source and target data centre running eNodeBs.

4.4.2.6 Throughput of CCN Data Packets when Content Migration Does Not Occur

This set of experiments is similar to the one mentioned in section 4.4.2.5, with the difference that now the content is not migrated from one data centre to another. The measurements of this performance metric are accomplished on the CCN users connected to the target eNodeB (hosted in the target data centre) that are requesting and receiving content.

4.4.3 VM (container) and Content Migration Support

One of advantages offered by the cloud concepts is elasticity. Provisioning of computing resources can be done rapidly and elastically. Telecommunication operators can scale up and scale down their resources whenever they want. For instance, when there is a big event such as football match or music concert, they can migrate their resources (e.g. instances/VMs of eNodeB) to the location where the event takes place and migrate or release them when the event has finished.

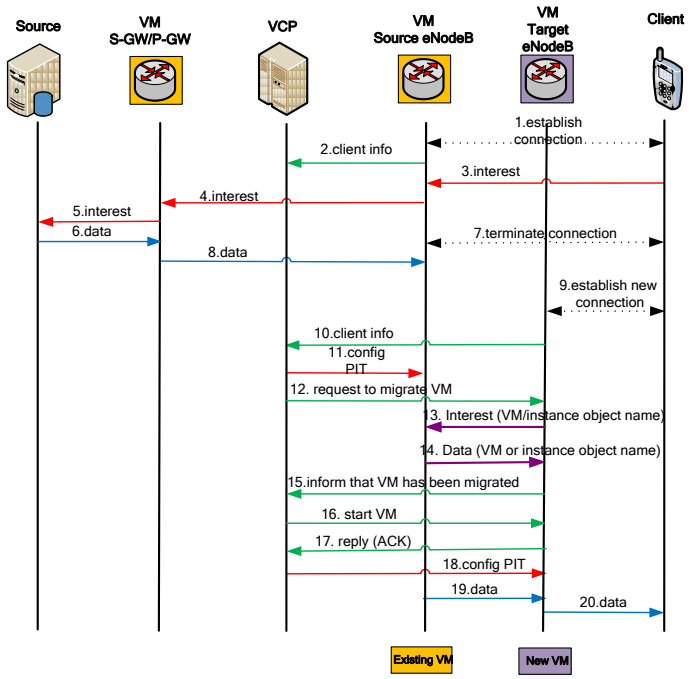

Figure 69: Information flow in the simulation of VM (container) and content migration support

The experiments that we performed for VM (container) and content migration support are based on the information flow diagram in section 3.3.2.1 (see Figure 49). Figure 69 shows the flow diagram that we implemented in order to simulate VM (container) and content migration support. This information flow diagram is the simplified information flow diagram depicted in the Figure 49. In this simulation, the cloud components used to support service continuity (e.g. MSM, SO, CM, ICN Manager) are also modelled within a single entity, named Virtualization Controlling Platform (VCP).

In the experiments, we simulated the handover of CCN mobile user from the source eNodeB to the target eNodeB hosted by different data centres. However, in the target cell served by the target data centre, there is no instance/VM that can serve the CCN mobile user's request; therefore an instance/VM migration to the target data centre is needed.

Basically there are three phases that need to be considered in the VM migration process, namely "suspend", "copy", and "resume" [68]. In the suspend phase, the VM is suspended on the origin host, and its memory is captured to the memory state file. In the copy phase, the VM's configuration, memory state and disk redo files are transferred to the destination host. In the resume phase, the VM memory state is restored from the snapshot and then resumes its execution.

In the experiments, we only considered the copy phase since based on [68], this phase is the one that provides the biggest impact to the time needed in VM migration. We simulated the migration of VM in order to show the impact of this migration on service continuity. The copy phase is simulated by sending the copy of VM (virtualized eNodeB), which is fragmented into several chunks from the source data centre (where the source eNodeB is hosted) to the target data centre (where the target eNodeB is hosted). As it can bee seen on the step 13 and 14 in the Figure 69, when the target eNodeB receives a request from the VCP to migrate VM, this target eNodeB will send a CCN Interest packet (that contains the name of requested instance/VM) to the source eNodeB. After receiving that Interest packet, the source eNodeB will send the requested instance/VM, that has been fragmented into several chunks, to the target eNodeB. These chunks of VM are sent from the source eNodeB to the target eNodeB using 50 Mbps generation/transmission speed. Note that this value is 5% of the capacity of the slowest wired link used in the simulation experiments, which can be considered to be a realistic value. Of course the operator can use any value for this generation/transmission speed, as long as the communication links do not become congested.

In this set of experiments, we deployed two eNodeBs (source and target eNodeB), one SGW/PGW node and one VCP node. All of CCN users connected to the LTE network are accessing a video streaming service (provided by the server) and watching the same video. There are 60 CCN users who are connected to the source eNodeB, while there is no user (both CCN and background users) connected to the target eNodeB, since we assumed that there is no VM in the target data centre that can provide a service to CCN users. In this simulation it is assumed that only one CCN user, that will move from the source data centre to the target data centre, will trigger the migration of the VM and the content. The other 59 CCN users will stay connected to the source eNodeB. This is done, since we want to investigate the impact of VM migration on the RTT delay of the CCN Interest/Data packets, when the Data packets will be received by the CCN user in the new location (served by the target data centre).

We performed several experiments by varying the size of VM (see 4.4.1.3) that need to be migrated and the distance between source and target data centres running eNodeBs (see section 4.4.1.1). Different from the set of experiments implemented in section 4.4.2 (where the location of VCP is varied), in this set of experiments, we placed the VCP only in one location, i.e., at the same data centre as where the target eNodeB is located, since this is considered as being the worst case situation (from the RTT point of view when the distance between source and target eNodeBs is increased). Those experiments were divided into two sets of experiments based on the size of VM that need to be migrated, such as follows:

- the size of migrated VM is 128 MB
- the size of migrated VM is 256 MB

In each set of experiment, the distance (number of hops) between source and target data centres running eNodeBs was varied. The number of hops that we simulated are 1 hop, 2 hops, 4 hops, 6 hops, 8 hops and 10 hops. Moreover, in performing these sets of experiments, we set the total utilization for both wireless and wired links to be ~80% (see section 4.2.6). The detail of topologies (including the location of network elements and the routing scenarios) used in these sets of experiments can be seen in Appendix A.

The following are the performace metrics that we measured in the set of experiments that focuses on VM (container) and content migration support.

4.4.3.1 Average RTT of Interest/Data Packets when VM and Content Migration Occur

As explained in section 4.3, the measured latency is the RTT (Round Trip Time) delay of the CCN request-response packets when CCN VM and content needs to be migrated from the source data centre to the target data centre. It is important to note that in this set of experiments the RTT is measured only between the Interest packet that triggers the VM migration and the Data packet associated with this Interest packet, which is received by the user after the VM and content are migrated from the source data centre to the target data centre. So this metric shows the delay of migrating both the VM and content from the source data centre to the target data centre. In the experiments, for each size of the migrated VM, we measured and compared the average latency in receiving CCN Data packets for each variation of distance (1 hop, 2 hops, 4 hops, 6 hops, 8 hops and 10 hops) between the source and target data centre running eNodeBs.

4.4.3.2 Maximum RTT of Interest/Data packets when VM and Content Migration Occur

Since the number of CCN users who will do handover is only one user, the maximum RTT of Interest/Data packet when VM and content migration occur is used to measure the maximum value of the RTT that is obtained when VM and content migration occurs. In particular, this metric shows the

maximum delay of migrating both the VM and content from the source data centre to the target data centre. In each set of experiments (based on the size of the migrated VM), we calculated and compared the maximum RTT for each variation of distance (1 hop, 2 hops, 4 hops, 6 hops, 8 hops and 10 hops) between the source and target data centre running eNodeBs.

4.4.3.3 Throughput of CCN Data packets when VM and content migration occurs

According to section 4.3, throughput is defined as the number of CCN data (response) packets correctly received by all CCN mobile users divided by simulation time. Since the number of CCN user doing handover in this set of experiments is only one, the throughput that we measured is only the throughput of one CCN mobile user. It is important to notice that if the VM and content migration procedure is not successful, then the user will not receive the Data packets that are associated with the Interest packet that initiated the VM migration procedure. In each set of experiments (based on the size of the migrated VM), we measured and compared the throughput of CCN Data packets for each variation of distance (1 hop, 2 hops, 4 hops, 6 hops, 8 hops and 10 hops) between the source and target data centre running eNodeBs.

Chapter 5

Simulation Results and Analysis

This chapter presents the results and analysis for both sets of experiments that focus on "Content Migration Support" and "VM (Container) and Content Migration Support".

5.1 Content Migration Results

The following sections show the results and analysis for the set of experiments that focuses on "Content Migration Support". The description of this set of experiments is provided in section 4.4.2. In this set of experiments, there are six performance metrics used to investigate the proposed solution, namely (1) average RTT of Interest/Data packets when content migration occurs, (2) average RTT of Interest/Data packets when content migration does not occur, (3) CDF of RTT of Interest/Data Packets when content migration occurs, (4) CDF of RTT of Interest/Data packets when content migration does not occur, (5) throughput of CCN Data packets when content migration occurs, and (6) throughput of CCN Data packets when content migration does not occur.

In this set of experiments the position of VCP is varied (see Figure 81 to Figure 86) as follows:

- position 1: VCP and SGW/PGW are hosted in the same data centre.
- position 2: VCP and target eNodeB (T-eNodeB) are hosted in the same data centre
- position 3: VCP and source eNodeB (S-eNodeB) are hosted in the same data centre

5.1.1 Average RTT of Interest/Data Packets when Content Migration Occurs

According to the information flow diagram that we implemented in this set of experiments (see Figure 68), the RTT of Interest/Data packets when content migration occurs can be affected by several factors, such as the position (distance) of source eNodeB, target eNodeB, and VCP; those three entities will communicate each other when content migration occurs. Moreover the location of SGW/PGW and remote host (server) will also affect the RTT. In addition, the utilization of resources both on the wireless link (Radio Access Network) and wired link will also affect the RTT of the Interest/Data packets. In addition to that, the way of implementing the CCN concept can also affect the RTT.

The average RTT of Interest/Data packets when content migration occurs is depicted in Figure 70. As it can be seen in Figure 70 (for all positions), the average RTT of Interest/Data packets, as expected, tends to increase when the number hops between source and target data centre is increased (from 1

hop to 10 hops). Furthermore, the average RTT values when VCP is placed in position 2 and 3 are almost equal.

Figure 70 also shows that when VCP and SGW/PGW are hosted in the same data centre (position 1), the RTT is higher than the RTT when the VCP is hosted in the same data centre as the target eNodeB (position 2), or in the same datacenter as the source eNodeB (position 3). This is because VCP is located in the middle (neither close to source or target data centres), in between source and target data centres (where eNodeBs are hosted). Moreover, the signaling delay associated with the communication between VCP and other entities is higher when the VCP is located in position 1. The slope of the RTT graph for the situation that the VCP is located in position 1 and when the number of hops is increased, is not as sharp as the one when VCP is placed in the other two positions. This can be explained as follows. The signaling delay associated with the communication between the VCP, target eNodeB and source eNodeB impacts the RTT value. When the distance, in hops, between the source data centre and target data centre is lower than a threshold, (in Figure 70 this is 10 hops, which is quite high) the signaling delay associated with the placement of the VCP, is higher when VCP is hosted at the same centre as the SGW/PGW. After this threshold of hops, the signaling delay is higher when the VCP is hosted at the same data centre as the source or target eNodeB.

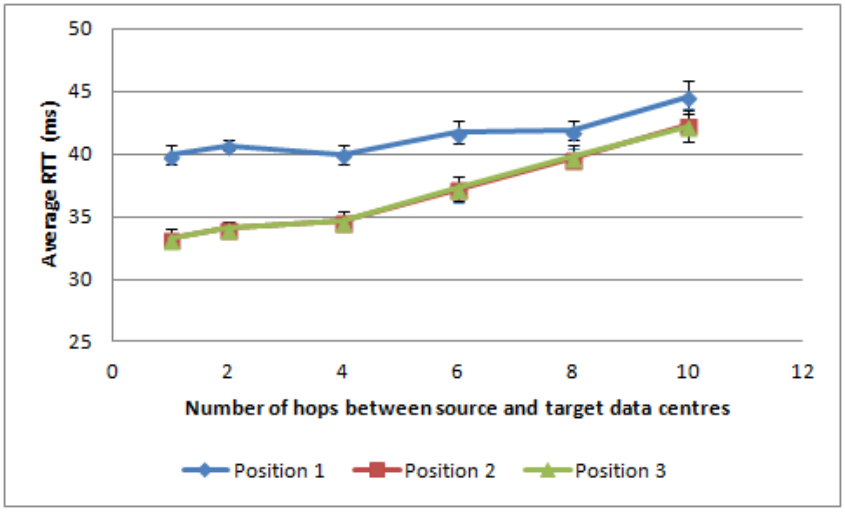

Figure 70: The average RTT of Interest/Data packets when content migration occurs

5.1.2 Average RTT of Interest/Data Packets when Content Migration Does Not Occur

Figure 71 shows the average RTT of Interest/Data packets when content migration does not occur for all distance (number of hops) between source data centre (where source eNodeB is hosted) and target data centre (where target eNodeB is hosted).

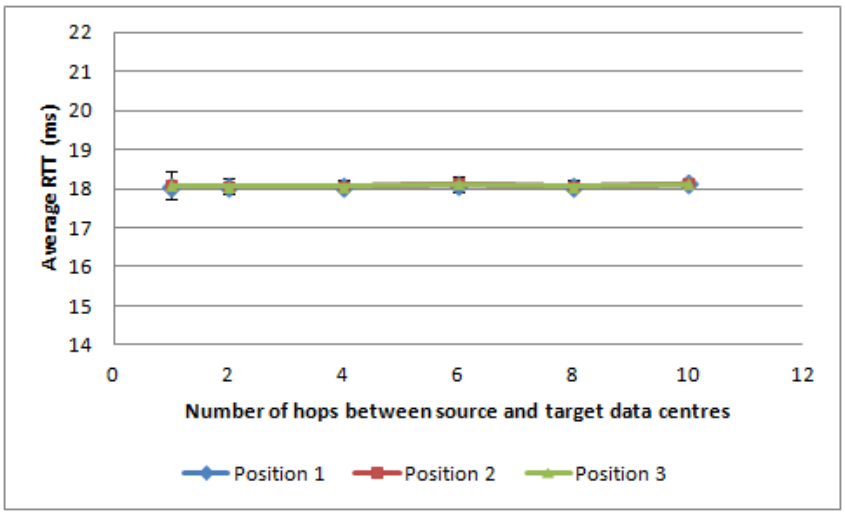

Figure 71: Average RTT of Interest/Data packets when content migration does not occur

As it can be seen on the graph, the average RTT values of Interest/Data packet for all positions of VCP and all distances (number of hops) between source and target data centres are almost equal, around 18 ms (lower than the RTT values when content migration occurs). These results show that the position of VCP and the distance between the source data centre (where the source eNodeB is hosted) and the target data centre (where the target eNodeB is hosted) do not affect the average RTT of Interest/Data packets when content migration does not occur. When there is no content migration, the location of VCP will not affect the RTT, since there is no signalling delay associated with the communication between VCP and data centres where the source or target eNodeB are hosted.

5.1.3 CDF of RTT of Interest/Data Packets when Content Migration Occurs

The following sections show the CDF of RTT of Interest/Data packets when content migration occurs and the position of VCP is varied.

5.1.3.1 VCP and SGW/PGW are Located in the Same Data Centre

Figure 72 shows the CDF of RTT of Interest/Data packets for all distances (number of hops) between the source data centre (where source eNodeB is hosted) and target data centre (where target eNodeB is hosted).

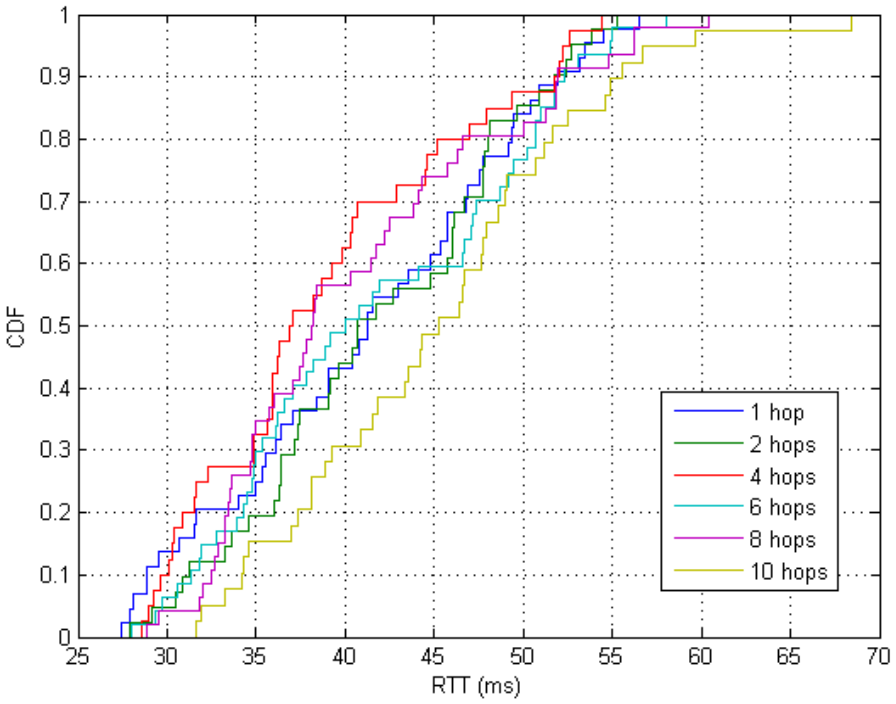

Figure 72: CDF of RTT of Interest/Data packets when content migration occurs, VCP and SGW/PGW are in the same data centre

As it can be seen in Figure 72, depending on the number of hops between the source data centre and target data centre, the minimum RTT varies between 28 ms and 32 ms, and the maximum RTT varies between 54 ms and 69 ms. The maximum RTT values of Interest/Data packets, when VCP is hosted at the same data centre as the SGW/PGW, are lower than 70 ms. Since the maximum delay for video streaming is 200 ms (see [78]), the RTT values of Interest/Data packets (for all distances) when content migration occurs are acceptable.

5.1.3.2 VCP and Target eNodeB are Located in the Same Data Centre

Figure 73 shows the CDF of RTT of Interest/Data packets for all distances (number of hops) between source data centre (where source eNodeB is hosted) and target data centre (where target eNodeB is hosted).

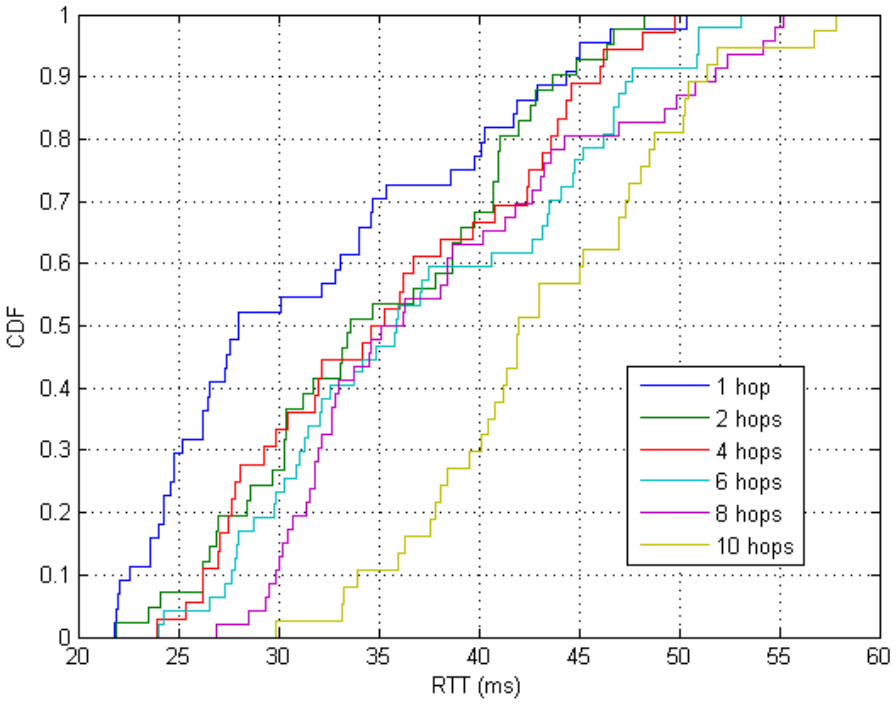

Figure 73: CDF of RTT of Interest/Data packets when content migration occurs, VCP and T-eNodeB are in the same data centre

As it can be seen in Figure 73, depending on the number of hops between the source data centre and target data centre, the minimum RTT varies between 22 ms and 30 ms, and the maximum RTT varies between 48 ms and 58 ms. The maximum RTT values of Interest/Data packets, when VCP is hosted at the same data centre as the T-eNodeB, are lower than 60 ms. Since the maximum delay for video streaming is 200 ms (see [78]), the RTT values of Interest/Data packets (for all distances) when content migration occurs are acceptable.

5.1.3.3 VCP and Source eNodeB are Located in the Same Data Centre

Figure 74 shows the CDF of RTT of Interest/Data packets for all distances (number of hops) between source data centre (where source eNodeB is hosted) and target data centre (where target eNodeB is hosted).

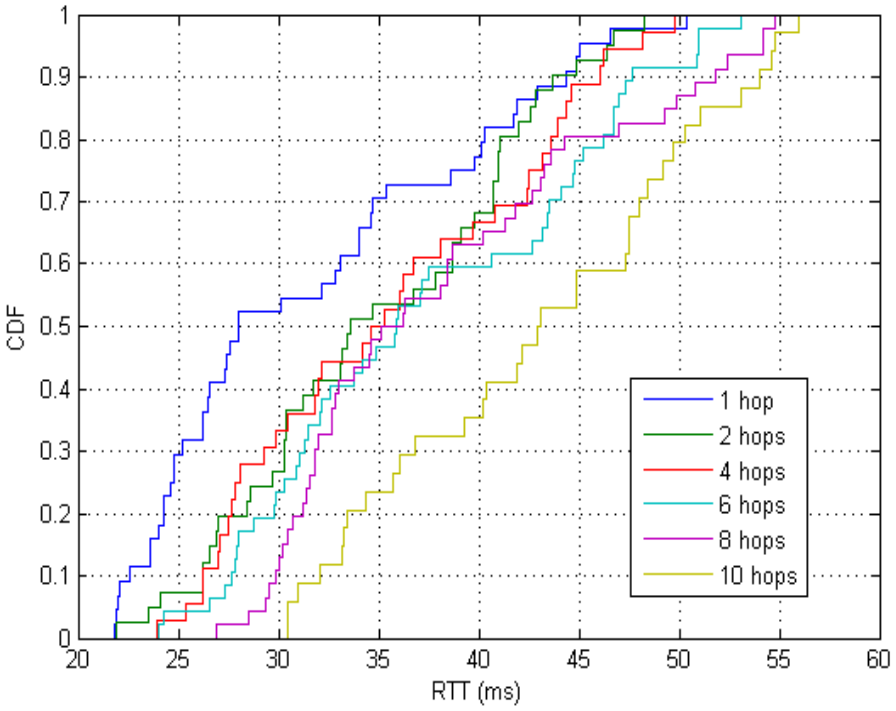

Figure 74: CDF of RTT of Interest/Data packets when content migration occurs, VCP and S-eNodeB are in the same data centre

As it can be seen in Figure 74, depending on the number of hops between the source data centre and target data centre, the minimum RTT varies between 22 ms and 31 ms, and the maximum RTT varies between 48 ms and 56 ms. The maximum RTT values of Interest/Data packets, when VCP is hosted at the same data centre as the S-eNodeB, are lower than 60 ms. Since the maximum delay for video streaming is 200 ms (see [78]), the RTT values of Interest/Data packets (for all distances) when content migration occurs are acceptable.

5.1.4 CDF of RTT of Interest/Data Packets when Content Migration Does Not Occur

The following sections show the CDF of RTT of Interest/Data packets when content migration does not occur, Since the RTT is not affected by the VCP position, only the scenario where the VCP is hosted at the same data centre as the SGW/PGW is discussed.

5.1.4.1 VCP and SGW/PGW are hosted in the Same Data Centre

Figure 75 shows the CDF of RTT of Interest/Data packets for all distances (number of hops) between the source data centre (where source eNodeB is hosted) and target data centre (where target eNodeB is hosted).

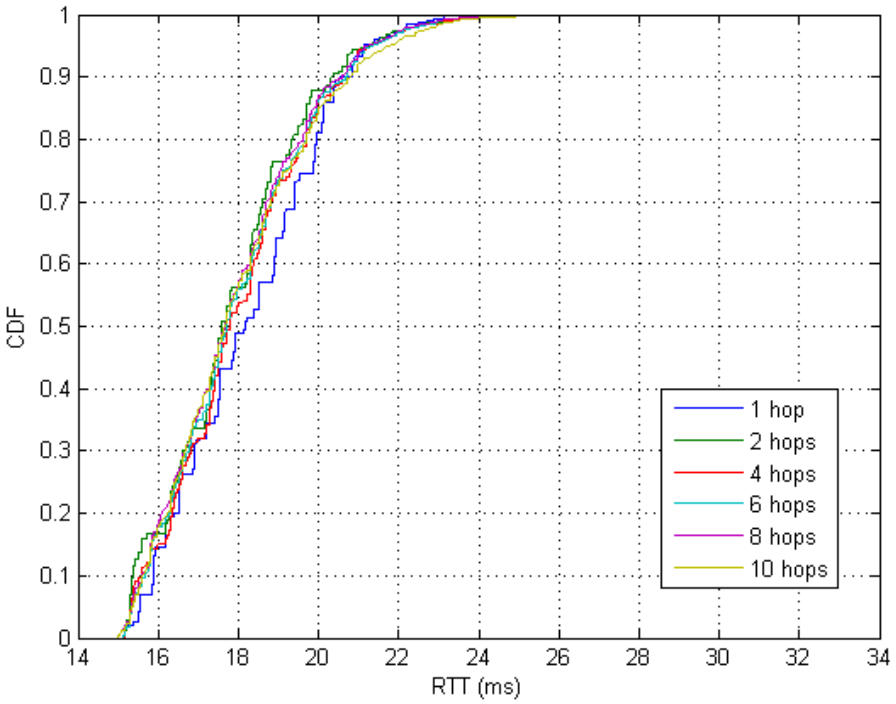

Figure 75: CDF of RTT of Interest/Data packets when content migration does not occur, VCP and SGW/PGW are in the same data centre

As it can be seen in Figure 75, the CDF of RTT for the situation that the VCP is hosted at the same data centre as the SGW/PGW for all distances are very similar; the minimum RTT is around 15 ms, and the maximum RTT varies between 24 ms and 33 ms. The maximum RTT values of Interest/Data packets are lower than 34 ms. Since the maximum delay for video streaming is 200 ms (see [78]), the RTT values of Interest/Data packets (for all distances) when content migration does not occur are acceptable.

5.1.5 Throughput of CCN Data Packets when Content Migration Occurs

As explained in section 4.3, throughput of CCN Data packets when content migration occurs is defined as the number of CCN Data (response) packets correctly received by all CCN mobile users divided by simulation time, when content is migrated from one data centre to another. The simulation time specified for the set of experiments of content migration support is 16.5 seconds, and as mentioned in section 4.4.2, the number of users simulated for doing handover is 51 users; it means that the handover rate during simulation time is 3.09 handovers/second.

Figure 76 shows the throughput of CCN Data packets when content migration occurs. The same figure shows the load that represents the number of Interest packets sent per second. As it can be seen in Figure 76, the throughput values for all positions of VCP and all distances (number of hops)

between source and target data centre are very similar to each other, around 2.5 Data packets/second. It means that the position of VCP and the distance between the source and target data centre does not significantly affect the throughput of the Data packets, when content migration occurs.

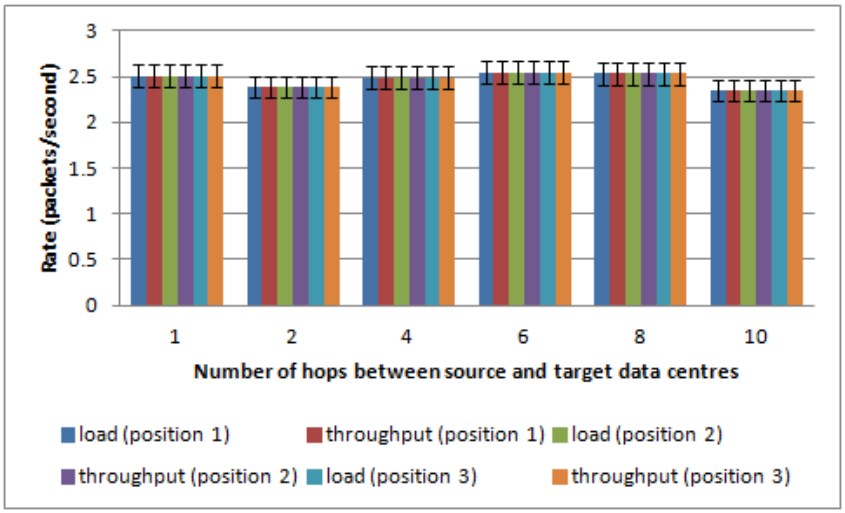

Figure 76: Throughput of CCN Data packets when content migration occurs

These throughput values are lower than the handover rate (3.09 handovers/second) because the content migration does not always occur when the mobile user moves from one location to another. Content migration will happen only if a user's request sent in the previous location (source eNodeB) has not been satisfied. If the user's request has been satisfied and the user moves to another location (served by different data centre), content migration will not occur and the user will send the next request to the target eNodeB in the new location. This is the reason why the throughput values are lower than the handover rate.

5.1.6 Throughput of CCN Data Packets when Content Migration Does Not Occur

As explained in section 4.3, the throughput of CCN Data packets when content migration does not occur is defined as the number of CCN Data (response) packets correctly received by all CCN mobile users divided by simulation time, when content is not migrated from one data centre to another.

Figure 77 shows the throughput of CCN Data packets when content migration does not occur. The same figure shows the load that represents the number of Interest packets sent per second. As it can be seen in the figure, the results show that the throughputs for all positions of VCP and all distances (number of hops) between source and data centres are almost equal, around 343Data packets/second. This means that the position of VCP and the distance between the source and target data centre does not affect the throughput of the Data packets, when content migration does not occur.

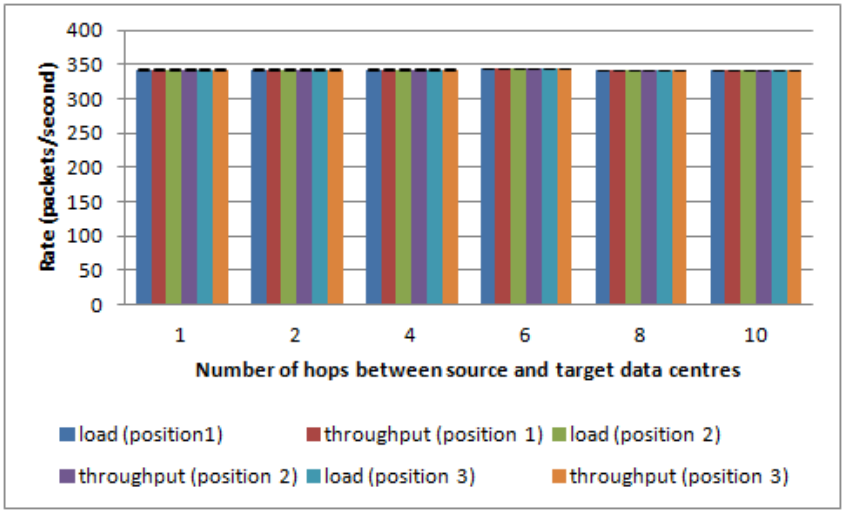

Figure 77: Throughput of CCN Data packets when content migration does not occur

5.1.7 Conclusion

The performance evaluation of the proposed solution used to investigate the content migration support in cloud based LTE systems has been carried out. Overall, the measurement results for all performance metrics show that the proposed solution can work well in supporting seamless service continuity in cloud based LTE systems when mobile users move from one location to another location served by different data centre.

When content migration occurs, the average RTT of Interest/Data packet will slightly increase when the number of hops (distance) between source and data centres increases. However, the maximum RTT can be tolerable (< 70 ms) since it is lower than the threshold specified in [78] (the maximum delay for video streaming is 200 ms). In addition, the position of the VCP will affect the RTT of Interest/Data packets when content migration occurs. In particular, when VCP and SGW/PGW are hosted in the same data centre (position 1), the RTT is higher than the RTT when the VCP is hosted in the same data centre as the target eNodeB (position 2), or in the same datacenter as the source eNodeB (position 3). This is because VCP is located in the middle (neither close to source nor target data centres), in between source and target data centres (where eNodeBs are hosted). Moreover, the signaling delay associated with the communication between VCP and other entities is higher when the VCP is located in position 1. The slope of the RTT graph for the situation that the VCP is located in position 1 and when the number of hops is increased, is not as sharp as the one when VCP is placed in the other two positions.

The measurements for the average RTT of Interest/Data packets, when content migration does not occur are also performed. The results show that the average RTT values for all positions of VCP and all distances (number of hops) between source and target data centres are almost equal, around 18 ms. These RTT values are acceptable since these values are lower than the threshold for video streaming,

which is 200 ms. Moreover, these results show that the position of VCP and the distance between the source data centre (where the source eNodeB is hosted) and the target data centre (where the target eNodeB is hosted) do not impact the average RTT of Interest/Data packets when there is no content migration.

The CDF of the RTT of Interest/Data packets when content migration occurs show that the maximum RTT is lower than 70 ms. In addition, the CDF of the RTT of Interest/Data packets when content migration does not occur shows that the maximum RTT values of Interest/Data packets are lower than 34 ms. Those RTT values are acceptable since the values are lower than the maximum tolerable delay for video streaming (200 ms).

The throughput values (when content migration occurs) for all positions of VCP and all distances (number of hops) between source and target data centre are very similar to each other, around 2.5 packets/second. It means that the position of VCP and the distance between source and target data centres do not have a significant impact on the throughput when content migration occurs. The total throughput is affected by how many number of Interest packets of UE (user) that has not been satisfied in the previous location (when handover is triggered).

Furthermore, for the throughput when content migration does not occur, the results show that the throughput values for all VCP positions and all distances (number of hops) between source and target data centres are almost equal. It means that the position of VCP and the distance between source and target data centre do not affect the throughput when content migration does not occur.

5.2 VM (Container) and Content Migration Results

The following sections show the results and analysis for the set of experiments that focuses on "VM (Container) and Content Migration Support". The simulation experiments are described in Section 4.4.3. In this set of experiments, there are three performance metrics used to investigate the proposed solution, namely (1) average RTT of Interest/Data packets when VM and content migration occur, (2) maximum RTT of Interest/Data packets when VM and content migration does not occur, and (3) throughput of CCN Data packets when content migration occur.

5.2.1 Average RTT of Interest/Data Packets when VM and Content Migration Occur

As explained in section 5.1.1, there are several factors that can affect the RTT of Interest/Data packets when content migration occurs, such as the position (distance) of source eNodeB, target eNodeB, VCP, SGW/PGW, remote host (server), etc. In case of VM migration, there are some other factors that can also affect the RTT of Interest/Data packets (see Figure 69) such as the delay of the signalling messages used to prepare the content and VM migration, and the size of the migrated VM.

The average RTT of Interest/Data packets when VM (128 MB and 256 MB) and content migration occur for all distances (number of hops) between source data centre (where source eNodeB is hosted) and target data centre (where target eNodeB is hosted) is depicted in Figure 78. As it can be seen in Figure 78, for 128 MB VM migration, the average RTT of Interest/Data is around 21seconds, while for 256 MB VM migration, the average RTT is around 42 seconds.

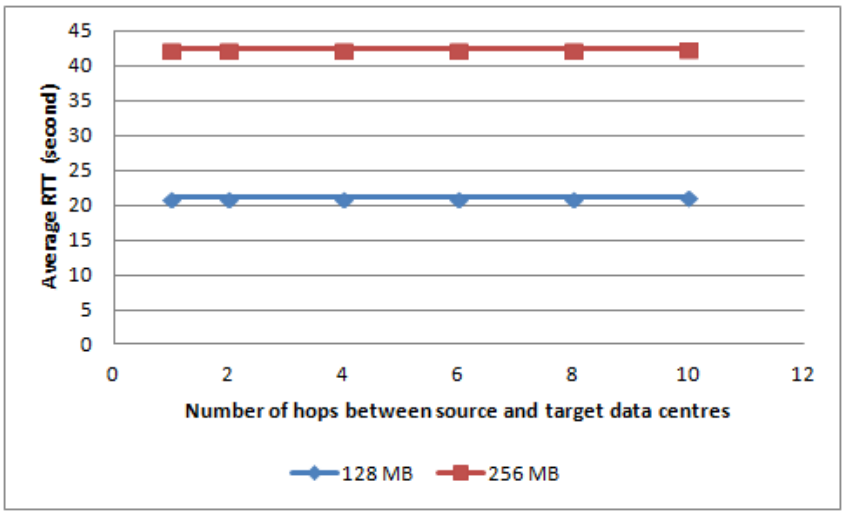

Figure 78: Average RTT of Interest/Data packets when VM and content migration occur

As explained in section 4.4.3, the VM is fragmented into several chunks before it is transferred from the source data centre to the target data centre. The delay for around 21 seconds (for 128 MB VM migration) and 42 seconds (for 256 MB VM migration) are mostly caused by the VM migration delay, that depends on the transmission speed used to migrate the VM from one data centre to another. Note that this transmission speed can be configured by the operator of the infrastructure that connects the two data centres. The signalling delays used to prepare the migration of the VM and content only contributes few milliseconds to the total RTT value.

The simulation results for 128 MB VM migration show that the minimum average RTT when VM and content migration occur is 21.260 seconds (when the distance is 1 hop), and the maximum average RTT is 21.273 seconds (when the distance is 10 hops). While the simulation results for 256 MB VM migration show that the minimum average RTT when VM and content migration occur is 42.477 seconds (when the distance is 1 hop), and the maximum average RTT is 42.488 seconds (when the distance is 10 hops). These RTT values are very high compared to the maximum delay of video streaming specified in [78]; therefore solutions, like mobility prediction, are needed in order to reduce the RTT of Interest/Data packet when VM migration occurs.

5.2.2 Maximum RTT of Interest/Data packets when VM and Content Migration Occur

The maximum RTT of Interest/Data packet when VM and content migration occur is used to measure the maximum value of the RTT that is obtained when VM and content migration occurs. As explained in section 4.2.5, the experiments that we performed are repeated several times using different random seeds. The RTT values shown in Figure 79 are the maximum RTT values obtained from the performed simulation experiments.

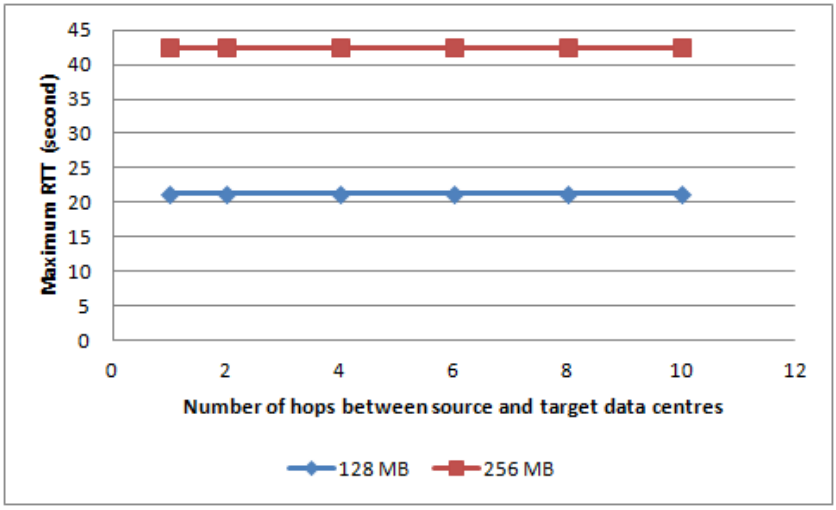

Figure 79: Maximum RTT of Interest/Data packets when VM and content migration occur

For 128 MB VM migration, the simulation results show that the minimum RTT value is 21.261seconds (when the distance is 2 hops) and the maximum RTT value is 21.277 seconds (when the distance is 10 hops). While for 256 VM migration, the simulation results show that the minimum RTT value is 42.477 seconds (when the distance is 2 hops) and the maximum RTT value is 42.491 seconds (when the distance is 6 hops).

5.2.3 Throughput of CCN Data packets when VM and Content Migration Occur

Throughput of CCN Data packets when VM and content migration occur is defined as the number of CCN Data (response) packets correctly received by all CCN mobile users divided by simulation time, when VM and content is migrated from one data centre to another. Figure 80 presents the throughput of CCN Data packets for each size of VM. In addition, Figure 80 also shows the load, which is the number of sent Interest packets per second.

For 128 VM migration, the simulation results show that the throughputs for all distances (number of hops) between source and target data centre are the same, 0.0372 packet/second. While for 256 VM migration, the throughputs for all distances (number of hops) between source and target data centre are also the same, 0.02 packet/second. It means that the distance between source and target data centre does not have any impact on the throughput of Data packets when content migration occurs.

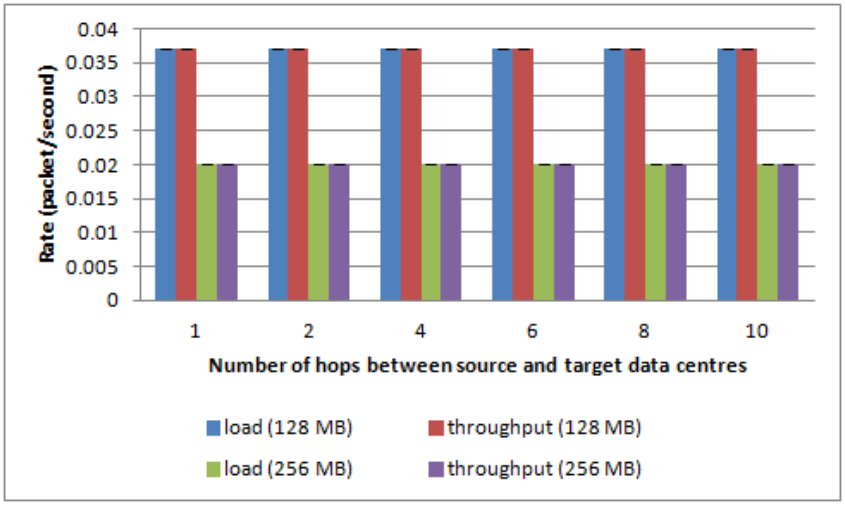

Figure 80: Throughput of CCN Data packets when VM and content migration occur

The throughput shown in Figure 80 is very low since the number of VMs that are migrated during one simulation run is one. In particular, the values of throughput (for both 128 MB and 256 MB VM migration) show that the number of Data packet which is migrated from one data centre to another is only one Data packet.

5.2.4 Conclusion

The performance evaluation of the proposed solution used to investigate the VM and content migration support in cloud based LTE systems has been carried out. The results show that, without a mobility prediction system (a system that can predict movements of users), service continuity will be difficult to be supported since the RTT of Interet/Data packet can take longer than the threshold (>200 ms) when VM and content migration occurs. For instance, if the size of migrated VM is 128 MB, seamless service continuity can be supported only if the VM can be migrated and ready in the target data centre at least 22 seconds before the handover is triggered. Therefore, solutions such as mobility prediction are needed to predict the movement of users and determine when the VM migration should be triggered in advance.

Similar to the results in the set of experiments that focuses on content migration support, the average RTT of Interest/Data packets tends to increase when the number hops between source and target data centre is increased (from 1 hop to 10 hops). However, the RTT values in this case are much higher. The highest delay is mostly caused by the VM migration delay, that depends on the transmission speed used to migrate the VM from one data centre to another. It is important to notice that this transmission speed can be configured by the operator of the infrastructure that connects the two data centres. The signalling delays used to prepare the migration of VM and content only contribute with some few milliseconds. If the size of migrated VM is increased, the RTT will become higher.

The throughputs (when VM and content migration occur) for all distances (number of hops) between source and target data centre are very similar to each other. It means that the distance between source and target data centre does not have any impact to the throughput when content migration occurs.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The cloud computing model offers better network resource utilization by pooling shared computing resources that can be rapidly provisioned and released. Applying this model into cellular systems, such as Long Term Evolution (LTE), could be a good solution in order 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.

Service continuity is one of the most important concepts used in mobile networks. A mobile user moving from one location to another should not lose service continuity. In cloud based LTE systems, migration of VMs and/or content should happen without losing service continuity.

In this report we have presented the evaluation of service continuity solution for cloud based LTE systems. Literature-based and simulation-based researches are used in order to answer the main research question in this project:

"How could seamless service continuity be implemented and evaluated in cloud based LTE systems?"

The main research question is divided into five sub-questions and the answers are as follows:

1. *What are the requirements that need to be satisfied by a service continuity solution when it is applied in cloud based LTE systems?*

Answer: The service continuity solution has to support migration of services that includes supports for IP address continuity, session continuity, content continuity, storage continuity and function continuity. The details of requirements are described in section 2.1.

2. *Which service continuity solutions could be implemented in cloud based LTE systems?*

Answer: Section 2.2 describes several service continuity solutions in mobile networks. Content Centric Networking (CCN) as an Information Centric Networking (ICN) approach could be implemented in cloud based LTE systems as the service continuity solution. Combination between CCN and Virtualization Controlling Platform (e.g. ICN Manager, SO, MSM, CC and CM) is able to support service continuity in cloud based LTE systems.

3. *Which architecture/framework could be used to support service continuity in cloud based LTE systems?*

Answer: The architecture/framework that could be used to support service continuity in cloud based LTE systems is shown in Figure 31 (ICN/CDNaaS) in section 2.3. In that architecture/framework, there are several components that are useful to support service continuity, such as ICN Manager, Service Orchestrator, Container Manager and Mobility Support Manager.

4. *How could the service continuity solution be applied in cloud based LTE systems?*

Answer: Section 3 describes several options to integrate CCN concept in cloud based LTE systems. CCN concept could be integrated in eNodeB, SGW and PGW. Moreover, designs of CCN integration that can support content and/or VM migration are proposed.

5. *How could the service continuity solution be evaluated and verified whether it is seamless?*

Answer: Section 4 and section 5 provide the answer for this sub-research question. In section 4, simulation experiments used to evaluate and verify the proposed solution are described in detail. In section 5, simulation results are analyzed to evaluate and verify whether the proposed solution can support seamless service continuity in cloud based LTE systems.

The simulation results show that the proposed solution can support seamless service continuity when content migration occurs. In particular, when content migration occurs, the average RTT of Interest/Data packet will slightly increase when the number of hops (distance) between source and data centres increases. However, the maximum RTT can be tolerable (< 70 ms) since it is lower than the threshold specified in [78] (the maximum delay for video streaming is 200 ms). In addition, the position of the VCP has an impact on the RTT of Interest/Data packets when content migration occurs. In particular, when VCP and SGW/PGW are hosted in the same data centre (position 1), the RTT is higher than the RTT when the VCP is hosted on the same data centre as the target eNodeB (position 2), or on the same datacenter as the source eNodeB (position 3). Furthermore, **t**he average RTT values when VCP is placed in position 2 and 3 are almost equal.

However, when VM migration occurs, the RTT of Interest/Data packets will be higher than the maximum tolerable delay (200 ms). The highest delay is mostly caused by the VM migration delay, that depends on the transmission speed used to migrate the VM from one data centre to another. It is important to notice that this transmission speed can be configured by the operator of the infrastructure that connects the two data centres. This can decrease the VM migration delay, however it is expected that this will not be lower than the threshold of maximum delay for video streaming, which is 200 ms. Therefore, in order to support seamless service continuity, solutions such as mobility prediction systems are needed to predict the movement of users and determine when the VM migration should be triggered in advance.

6.2 Future Work

There are many topics that can be explored to enhance service continuity solution in cloud based LTE systems. More experiments need to be accomplished in order to verify the performance of the CCN concept in supporting service continuity. Such experiments could focus on:

- using different LTE configuration parameters (e.g. channel bandwidth or type of scheduler in eNodeB)
- investigating VM and content migration when combining the CCN concept with a mobility prediction solution
- using another handover scenario such as the S1-based handover with and without SGW relocation
- integrating and investigating the implementation of proxy that can translate HTTP(S) based traffic to CCNx traffic and vice versa
- investigating the impact of cloud components on service continuity when they are implemented as separate entities. Currently, not all cloud components, such as Cloud Controller, ICN Manager, Service Orchestrator, Container Manager and Mobility Support Manager, are modelled as separate entities, but they are grouped in the VCP. Therefore, those components can be implemented as separate entities and investigate what is their impact on supporting service continuity
- investigating the integration and implementation of the CCN retransmission mechanism on the service continuity performance.

References

[1] Mell, P. and T. Grance, ”The NIST Definition of Cloud Computing”, Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-145, September 2011.

[2] Q. Zhang, L. Cheng, and R. Boutaba,” Cloud computing: state-of-the-art and research challenges”, Journal of Internet Services and Applications, 2010.

[3] P. Gupta and S. Gupta2,” Mobile Cloud Computing: The Future of Cloud”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 1, Issue 3, September 2012.

[4] A.J. Staring ,” Applying the Cloud Computing Model in LTE based Cellular Systems”, Proc. of the 17th Twente Student Conference on IT, 2012.

[5] N. Niebert et.al., “Network Virtualization: A Viable Path Towards the Future Internet”, Wireless Personal Communications, Volume 45, Issue 4, pp 511-520, June 2008.

[6] Y. Zaki et.al., “LTE mobile network virtualization”, Journal Mobile Networks and Applications, archive Volume 16, Issue 4, Pages 424-432, August 2011.

[7] N. M.M.K. Chowdhury and R. Boutaba, ” Network Virtualization: State of the Art and Research Challenges”, IEEE Communications Magazine, Vol.47, Issue 7, pages 20-26, July 2009.

[8] Z. Zhu, Q. Wang, and Y. Lin,” Virtual Base Station Pool: Towards A Wireless Network Cloud for Radio Access Networks”, Proc. of the 8th ACM International Conference on Computing Frontiers, 2011.

[9] A. Khan et.al., ”The Reconfigurable Mobile Network”, Proc. of IEEE International Conference on Communications Workshop, 2011.

[10] R. Kokku et.al., “NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks”, IEEE/ACM Transactions on Networking, Vol. 20, No. 5, October 2012.

[11] F. Firmin, “The Evolved Packet Core”, Retrieved January 5th, 2013, from `http://www.3gpp.org/The-Evolved-Packet-Core`.

[12] P. Bosch et al., ”Telco Clouds and Virtual Telco: Consolidation, Convergence, and Beyond”, Proc. of the 6th IFIP/IEEE International Workshop on Broadband Convergence Networks, 2011.

[13] K. Pentikousis and B. Ohlman, “ICN Baseline Scenarios”, IETF Internet Draft, November 2012.

[14] B. Ahlgren et al., ”A survey of information-centric networking”, IEEE Communication Magazine vol. 50 pages 26-36, July 2012.

[15] T. Koponen et al., “A Data-Oriented (and Beyond) Network Architecture,” Proc. SIGCOMM ’07, Kyoto, Japan, Aug. 27–31, 2007.

[16] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs and R. Braynard, “Networking named content”, Proc. of 5th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT’09), 2009.

[17] M. Ain et al., “D2.3 – Architecture Definition, Component Descriptions, and Requirements,” Deliverable, PSIRP 7th FP EU-funded project, Feb. 2009.

[18] B. Ahlgren et al., “Second NetInf Architecture Description,” 4WARD EU FP7 Project, Deliverable D-6.2 v2.0, Apr. 2010, FP7-ICT-2007-1-216041- 4WARD / D-6.2, http://www.4ward-project.eu/.

[19] M. Gritter and D. R. Cheriton, “TRIAD: A New Next-Generation Internet Architecture”, Jan 2000.

[20] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters: A Survey", Internet Mathematics **1** (4): 485–509, 2005.

[21] M. Gritter and D.R. Cheriton, “An Architecture for Content Routing Support in the Internet”, Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems, 2001.

[22] T. Braun, V. Hilt, M. Hofmann, I. Rimac, M. Steiner and M. Varvello, “Service-Centric Networking”, Proc. of 4th International Workshop on the Network of the Future, 2011.

[23] R. Moskowitz and P. Nikander, “Host Identity Protocol (HIP) Architecture”, RFC4423, RFC Editor, 2006.

[24] A. Pathak et.al., “Host Identity Protocol for Linux”, Linux Journal, Volume 2009 Issue 187, Article No. 5, November 2009.

[25] P. Jokela et.al, “Using the Encapsulating Security Payload (ESP) transport format with the Host Identity Protocol (HIP)”, IETF RFC 5202, April 2008.

[26] S. Novaczki et.al., “Design and Evaluation of a Novel HIP-Based Network Mobility Protocol”, Journal of Networks, Vol 3, No 1 (2008), 10-24, Jan 2008.

[27] J. Laganier and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", IETF RFC 5204, April 2008.

[28] P. Nikander, J. Laganier: "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", IETF RFC 5205, April 2008.

[29] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005.

[30] R. Atkinson, S. Bhatti and S. Hailes, "ILNP: Mobility, Multi-Homing, Localised Addressing and Security Through Naming", Telecommunication Systems, vol. 42, no. 3-4 pp273-291, Springer, 2009.

[31] T. Dreibholz, A. Jungmaier, and Michael Tüxen, "A New Scheme for IP-based Internet-Mobility", Proc. of the 28th Annual IEEE International Conference on Local Computer Networks (LCN '03), IEEE Computer Society, 2003.

[32] R. Stewart, "Stream Control Transmission Protocol", IETF RFC 4960, September 2007.

[33] H. Schulzrinne and E. Wedlund, "Application-Layer Mobility Using SIP", ACM Mobile Computing and Communication Review, Volume 4, Issue 3, pp. 47–57, July 2000.

[34] F. Vakil et.al.,"Supporting Service Mobility with SIP", IETF Internet Draft, December 2000.

[35] S. Gundavelli et.al., "Proxy Mobile IPv6", IETF RFC 5213, August 2008.

[36] D. Johnson et.al, "Mobility Support in IPv6", IETF RFC 3775, June 2004.

[37] J. Lei and X. Fu, "Evaluating the Benefits of Introducing PMIPv6 for Localized Mobility Management", Wireless Communications and Mobile Computing Conference, 2008.

[38] Open Networking Foundation,"Software-Defined Networking: The New Norm for Networks", April 2012.

[39] Cariden Technologies," Infrastructure SDN with Cariden Technologies: Providing the benefits of software-defined networking in resource-constrained and dynamic demand environments today", August 2012.

[40] N. McKeown et.al.," OpenFlow: Enabling Innovation in Campus Networks", Computer Communication Review - CCR , vol. 38, no. 2, pp. 69-74, 2008.

[41] OpenStack: The Open Source Cloud Operating System. Retrieved February 26, 2013, from http://www.openstack.org/software/.

[42] Nova. Retrieved February 26, 2013, from https://wiki.openstack.org/wiki/Quantum https://wiki.openstack.org/wiki/Nova.

[43] Swift. Retrieved February 26, 2013, from https://wiki.openstack.org/wiki/Swift.

[44] Cinder. Retrieved February 26, 2013, from https://wiki.openstack.org/wiki/Cinder.

[45] Quantum. Retrieved February 26, 2013, from https://wiki.openstack.org/wiki/Quantum.

[46] Quantum NEC openflow plugin. Retrieved February 26, 2013, from https://wiki.openstack.org/wiki/Quantum_NEC_OpenFlow_Plugin.

[47] T. Torii, "OpenStack with OpenFlow/SDN", Retrieved February 26, 2013, from http://events.csdn.net/OpenStack/Takashi%20Torii-OpenStack%20and%20OpenFlowSDN.pdf.

[48] M. J. Freedman, M. Arye, P. Gopalan, S. Y. Ko, E. Nordstrom, J. Rexford, and D. Shue, "Service-Centric Networking with SCAFFOLD", Technical Report TR-885-10, Princeton University, CS, September 2010.

[49] N. Gude et.al, "NOX: Towards an Operating System for Networks", ACM SIGCOMM Computer Communication Review, Volume 38 Issue 3, July 2008.

[50] OpenEPC–Open Evolved Packet Core. Retrieved March 8, 2013, from http://www.openepc.net

[51] Intelligent Offload. Retrieved August 16, 2013 from http://www.movik.com/node/28

[52] M. Liebsch, Z. Yousaf, "Runtime Relocation of CDN Serving Point – Enabler for Low Costs Mobile Content Delivery", WCNC, 2013.

[53] Mobile Cloud Networking Project, https://www.mobile-cloud-networking.eu/

[54] MCN D2.2 "Overall Architecture Definition, Release 1, European Commission", deliverable 2.2, EU FP7 Mobile Cloud Networking public deliverable, November 2013.

[55] MCN D3.1, "Infrastructure Management Foundations – Specifications & Design for Mobile Cloud framework", deliverable 3.1, EU FP7 European Commission, EU FP7 Mobile Cloud Networking public deliverable, November 2013.

[56] MCN D4.1, "Mobile Network Cloud Component Design", deliverable 4.1, European Commission, EU FP7 Mobile Cloud Networking public deliverable, November 2013.

[57] MCN D5.1, "Design of Mobile Platform Architecture and Services, European Commission", deliverable 5.1, European Commission, EU FP7 Mobile Cloud Networking public deliverable, November 2013.

[58] R. Haw and C.S. Hong, "A seamless content delivery scheme for flow mobility in Content Centric Network", Network Operations and Management Symposium, 2012.

[59] LTE-EPC Network Simulator. Retrieved August 18, 2013 from http://iptechwiki.cttc.es/LTE-EPC_Network_Simulator_(LENA)

[60] CTTC, "LTE Simulator Documentation, Release 5", January 23, 2013.

[61] NS-3 based Named Data Networking Simulator. Retrieved August 17, 2013 from http://ndnsim.net/.

[62] Ebone ISP. Retrieved August 17, 2013 from http://www.ebonenet.com/.

[63] Rocketfuel Maps. Retrieved August 17, 2013 from http://cs.washington.edu/research/networking/rocketfuel/interactive/.

[64] D. Wischik and N. McKeown, "Part I: Buffer Sizes for Core Routers", ACM SIGCOMM Computer Communication Review, Volume 35 Issue 3, July 2005.

[65] NGNM Allicance, "NGMN Radio Access Performance Evaluation Methodology", January 2008.

[66] M.M. Zonoozi and P. Dassanayake, "User Mobility Modeling and Characterization of Mobility Patterns", IEEE Journal on Selected Areas in Communications, vol. 15, no. 7, September, 1997.

[67] G. Watts et.al, "Effects of speed distributions on the Harmonoise model predictions", The 33rd International Congress and Exposition on Noise Control Engineering, August 2004.

[68] M. Zhao and R.J. Figueiredo, "Experimental Study of Virtual Machine Migration in Support of Reservation of Cluster Resources", Proceedings of the 2nd international workshop on Virtualization technology in distributed computing, 2007.

[69] T.G. Pham, "Integration of IEC 61850 MMS and LTE to support smart metering communications", Master thesis report, University of Twente, 2013.

[70] A.D. Nguyen, "Integration of IEC 61850 MMS and LTE to support remote control communications in electricity distribution grid", Master thesis report, University of Twente, 2013.

[71] https://github.com/triadimas/src/tree/master/lte

[72] https://github.com/tgpham/gen-udp.git

[73] D. Ammar et.al, "A new tool for generating realistic internet traffic in NS-3", Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, 2011.

[74] iOS Developer Library. Retrieved October 16, 2013, from https://developer.apple.com/library/ios/technotes/tn2224/_index.html#//apple_ref/doc/uid/DTS40009745-CH1-BITRATERECOMMENDATIONS

[75] Fundamental of Digital Video. Retrieved October 16, 2013, from http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/pktvideoaag.html.

[76] https://github.com/triadimas/src/tree/master/applications

[77] https://github.com/triadimas/src/tree/master/topology-read/examples

[78] IEEE 802.11-03/802r23, Retrieved October 28, 2013 from https://mentor.ieee.org/802.11/bp/StartPage.

Appendix A

Network Topology

The following are the details of topologies and routing scenarios implemented in the experiments. These topologies and routing scenarios are divided into three sets of experiments based on position of VCP:

- Position 1: VCP and SGW/PGW are located in the same data centre
- Position 2: VCP and target eNodeB are located in the same data centre
- Position 3: VCP and source eNodeB are located in the same data centre

For the set of experiments that focuses on “Mobility and Content Migration Support”, all positions of VCP are considered, while for the set of experiments that focuses on "Mobility, VM (container) and Content Migration Support", only position 2 is considered.

A.1 Distance between Source and Target eNodeBs is 1 hop

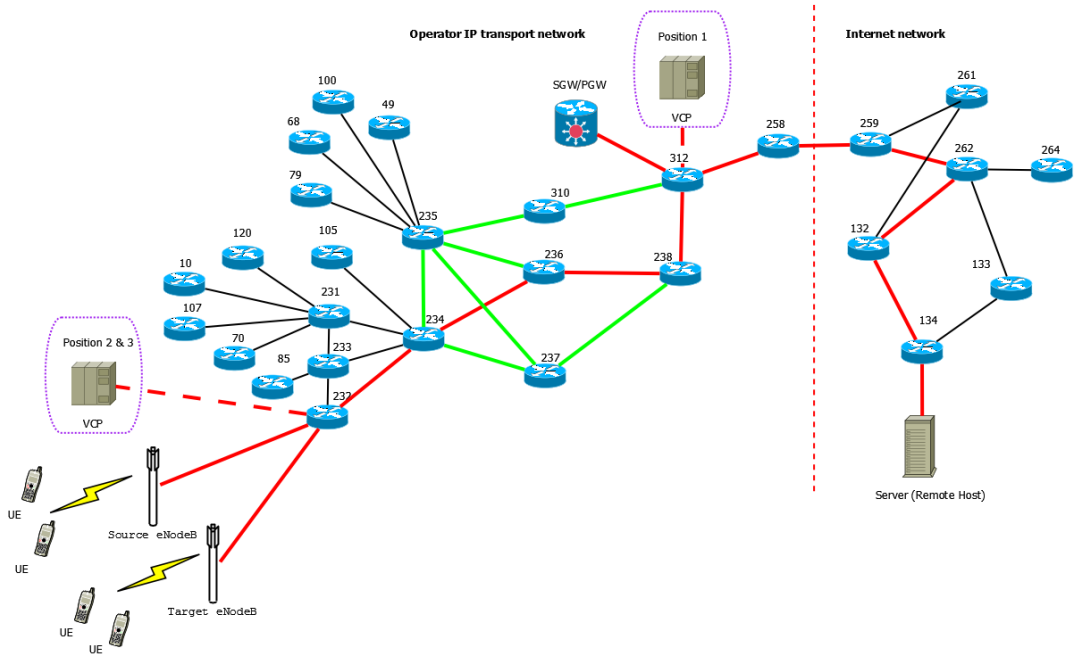

Figure 81: Topology and routing scenario for 1-hop

The red line in Figure 81 shows the route used by the network entities (eNodeB, SGW/PGW, VCP, and server) to reach each other. In this topology, the distance between two data centres serving two different eNodeBs is one hop. In the experiments, the position of VCP is varied from position 1 to position 3.

Since the distance between source and target eNodeB is 1 hop, so that position 2 and position 3 of VCP are the same.

A.2 Distance between Source and Target eNodeBs is 2 hops

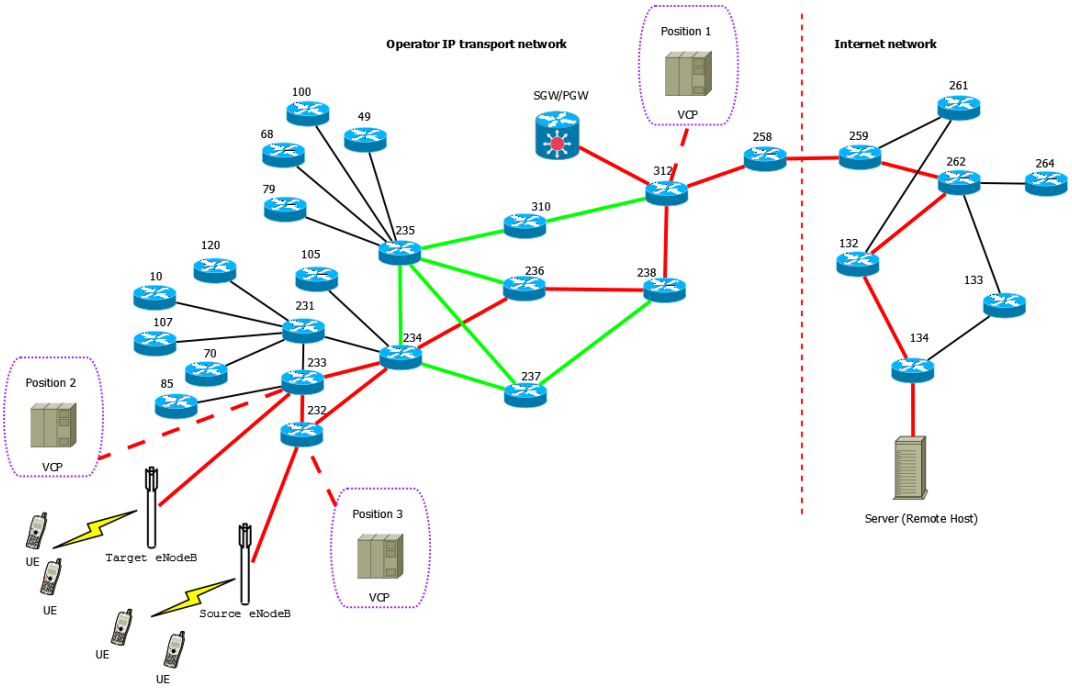

Figure 82: Topology and routing scenario for 2 hops

The red line in Figure 82 shows the route used by the network entities (eNodeB, SGW/PGW, VCP, and server) to reach each other. In this topology, the distance between two data centres serving two different eNodeBs is two hops. In the experiments, the position of VCP is varied from position 1 to position 3.

A.3 Distance between Source and Target eNodeBs is 4 hops

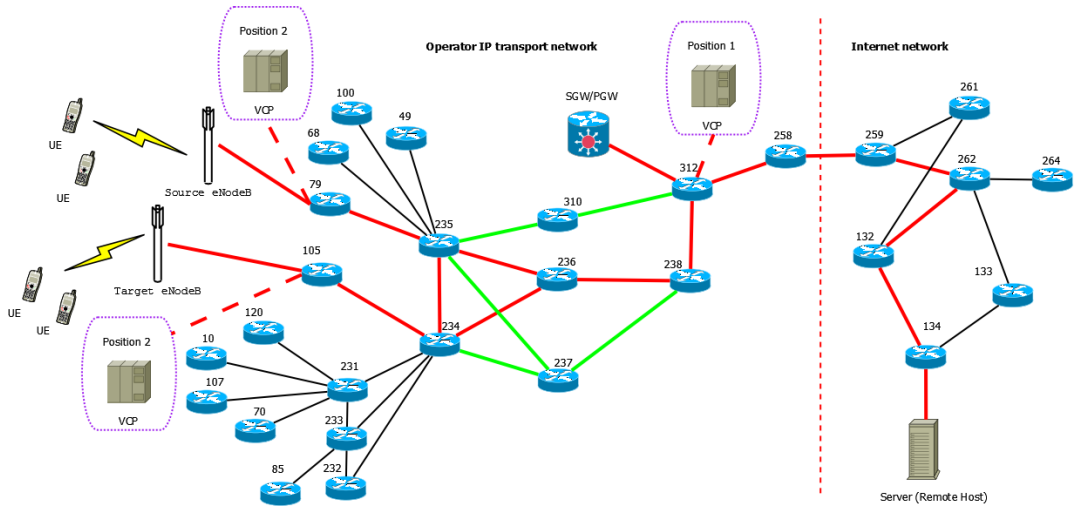

Figure 83: Topology and routing scenario for 4 hops

The red line in Figure 83 shows the route used by the network entities (eNodeB, SGW/PGW, VCP, and server) to reach each other. In this topology, the distance between two data centres serving two different eNodeBs is four hops. In the experiments, the position of VCP is varied from position 1 to position 3.

A.4 Distance between Source and Target eNodeBs is 6 hops

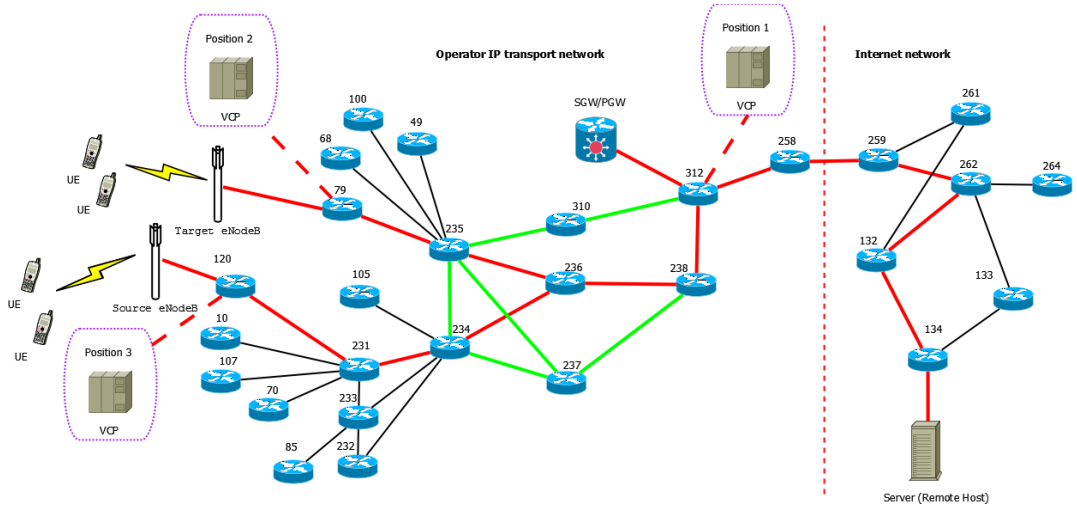

Figure 84: Topology and routing scenario for 6 hops

The red line in Figure 84 shows the route used by the network entities (eNodeB, SGW/PGW, VCP, and server) to reach each other. In this topology, the distance between two data centres serving two different eNodeBs is six hops. In the experiments, the position of VCP is varied from position 1 to position 3.

A.5 Distance between Source and Target eNodeBs is 8 hops

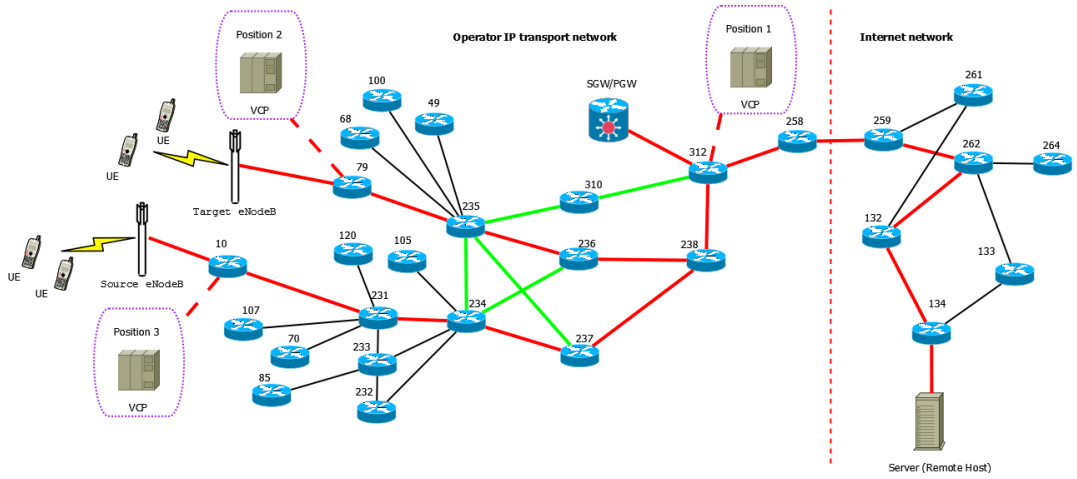

Figure 85: Topology and routing scenario for 8 hops

The red line in Figure 85 shows the route used by the network entities (eNodeB, SGW/PGW, VCP, and server) to reach each other. In this topology, the distance between two data centres serving two different eNodeBs is eight hops. In the experiments, the position of VCP is varied from position 1 to position 3.

A.6 Distance between Source and Target eNodeBs is 10 hops

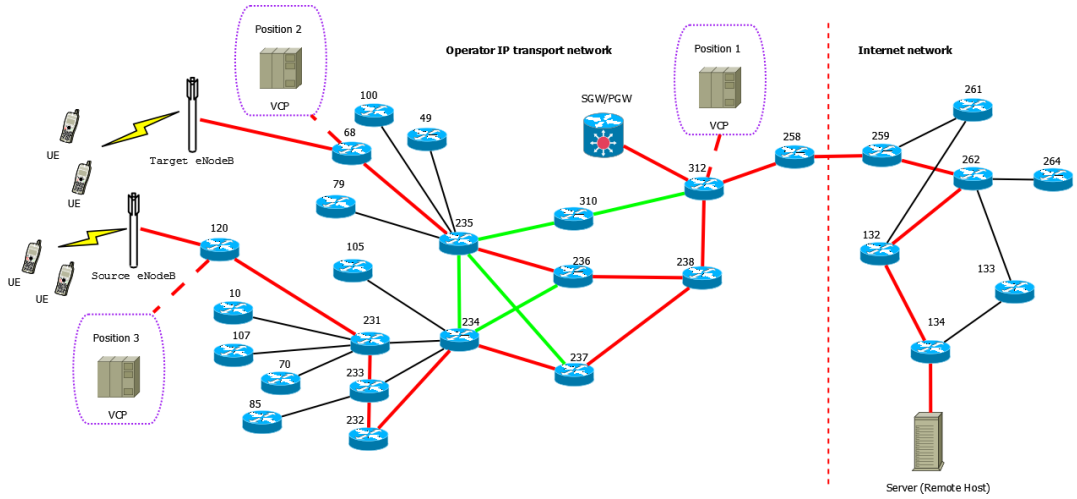

Figure 86: Topology and routing scenario for 10 hops

The red line in the Figure 86 shows the route used by the network entities (eNodeB, SGW/PGW, VCP, and server) to reach each other. In this topology, the distance between two data centres serving two different eNodeBs is ten hops. In the experiments, the position of VCP is varied from position 1 to position 3.

Appendix B

User Guide: Implementation of Content and/or VM Migration Support in NS-3 LENA

B.1 Installation

LENA is the LTE module implemented in ns-3. In order to implement CCN concept in LENA, the customized version of ns-3 called ndnSIM is required. It is important to note that the ndnSIM used in this project is based on ns-3.17 release. The installation guide of ndnSIM can be found in the following link: `http://ndnsim.net/getting-started.html`

After the ndnSIM is completely installed, download the customized source codes of lte and applications modules from `https://github.com/triadimas/src`. There are three directories that can be found from that link, namely "applications", "lte" and "topology-read". Within the "applications" directory there are source codes used for generating traffic (CCN, signalling and wired link background traffic), while within "lte" directory, there are source codes of LENA that have been modified for deploying IP transport network and integrating CCN concept in LTE systems. In addition, the topology-read contains the topology files implemented in the experiments.

Download the files within those directories and put them into the right directory/module of ns-3. Replace the existing files with the new files downloaded from those three directories. Important to note that the wscript files within the "applications" and "lte" modules of ns-3 have to be replaced by the wscript files placed in the "applications" and "lte" directories.

B.2 Deploying CCN Based LTE Systems to Support Content and/or VM Migration in NS-3 LENA

The way in deploying CCN based LTE systems in ns-3 is very similar to the way in deploying the existing LTE systems. The following shows the steps of how to deploy CCN based LTE systems together with IP transport networks placed in the middle of EPS components.

To simulate content and/or VM migration, the following configurations have to be set:

```
Config::SetDefault("ns3::EpcX2::VmMigration",UintegerValue (vmMigration));
Config::SetDefault("ns3::UdpEchoServer2::VmMigration",UintegerValue(vmMigration));
Config::SetDefault("ns3::EpcX2::NumOfPacket",UintegerValue (numOfPacket));
```

The variable "`vmMigration`" is an unsigned integer value used to set the type of simulation. If we want to simulate a content migration, the value of "`vmMigration`" has to be set as 0, while if we

want to simulate content and VM migrations, "vmMigration" has to be set as a positive integer. In addition, "numOfPacket" is used to specify the number of packets (chunks of VM) that has to be sent from a source data centre to a target data centre.

Furthermore, the steps how to implement a topology of IP transport network in the middle of EPS components are shown on the following code. Important to note that the eNodeB, SGW/PGW and VCP nodes can be included directly into a topology file. In this example, the topology file which is used is sim6hop-vcp-target.txt, and eNodeB, SGW/PGW and VCP nodes are already included in that file.

```
std::string format ("Rocketfuel");
std::string input ("src/topology-read/examples/sim6hop-vcp-target.txt");
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;
}
```

After the topology of routers (nodes) is created, the internet stack has to be installed into the routers as follows:

```
NS_LOG_INFO ("creating internet stack");
InternetStackHelper stack;
stack.Install (backboneNodes);
```

Afterwards, the next step is assigning an IP address to each router and connecting them to each other, such as follow:

```
NS_LOG_INFO ("creating ip4 addresses");
Ipv4AddressHelper bbAddress;
```

```
bbAddress.SetBase ("10.0.0.0", "255.0.0.0");
Ipv4AddressHelper internetAddress;
internetAddress.SetBase ("167.0.0.0", "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];
PointToPointHelper p2p;

for (int j = 0; j < totlinks; j++)
{
   ndc[j] = p2p.Install (nc[j]);
}

NS_LOG_INFO ("creating ipv4 interfaces");
Ipv4InterfaceContainer* ipic = new Ipv4InterfaceContainer[totlinks];

for (int i = 0; i < totlinks; i++)
{
  if (i < 12)
  {
     ipic[i] = bbAddress.Assign (ndc[i]);
     bbAddress.NewNetwork ();
  }
  else
  {
     ipic[i] = internetAddress.Assign (ndc[i]);
     internetAddress.NewNetwork ();
  }
}
```

After the topology of routers is ready, the next step is creating a remote host (server) node located on the internet and connecting it to the one of routers/nodes on the network as follows:

```
NodeContainer remoteHostContainer;
remoteHostContainer.Create (1);
Ptr<Node> remoteHost = remoteHostContainer.Get (0);
stack.Install (remoteHostContainer);

PointToPointHelper p2ph;
Ipv4AddressHelper ipv4h;
ipv4h.SetBase ("90.0.0.0", "255.0.0.0");

NetDeviceContainer internetDevices=p2ph.Install(remoteHost,backboneNodes.Get(16));
Ipv4InterfaceContainer internetIpIfaces=ipv4h.Assign (internetDevices);
Ipv4Address remoteHostAddr=internetIpIfaces.GetAddress (0);
```

Afterwards, the LTE nodes (UE, eNodeB and SGW/PGW nodes) are created using the following code:

```
Ptr<LteHelper> lteHelper = CreateObject<LteHelper> ();
Ptr<EpcHelper> epcHelper = CreateObject<EpcHelper> (backboneNodes.Get(0));
lteHelper->SetEpcHelper (epcHelper);
NodeContainer ueNodesMove;
NodeContainer enbNodes;
enbNodes.Add(backboneNodes.Get(2));
enbNodes.Add(backboneNodes.Get(4));
ueNodesMove.Create(numberOfUeMove);
```

There are several steps more that need to be accomplished in order to work with LTE in ns-3, such as:

- installing mobility model for eNodeBs and UEs
- installing LTE Devices to the eNodeB and UE nodes
- installing the IP stack on the UEs
- assigning IPv4 addresses on UEs
- setting default gateways for UEs
- attaching UE nodes to the eNodeB(s)
- installing application to the end nodes (e.g. UEs, remote host and VCP)

Furthermore, to simulate an X2 handover, the X2 interface between two eNodeBs has to be added, and then the time when the handover is triggered has to be specified.

The details of the steps to simulate content and/or VM migration can be found in the following link: `https://github.com/triadimas/examples/blob/master/simple-vm-migration.cc`.