Doctoral Thesis

Architecture, Modeling, Planning, and Dynamic Provisioning of Softwarized 5G Mobile Core Networks

Author:
Jonathan Prados Garzón

Supervisors:
Dr. Juan Manuel López Soler
Dr. Pablo Ameigeiras Gutiérrez

A thesis submitted in fulfillment of the requirements
to obtain the International Doctor degree as part of the
Programa de Doctorado en Tecnologías de la Información y las Comunicaciones
in the

Wireless and Multimedia Networking Lab Research Group

Departamento de Teoría de la Señal, Telemática y Comunicaciones

Granada, September 10, 2018

Editor: Universidad de Granada. Tesis Doctorales
Autor: Jonathan Prados Garzón
ISBN: 978-84-1306-018-7
URI: http://hdl.handle.net/10481/53865

Declaration of authorship

The doctoral candidate Mr. Jonathan Prados Garzón, and the thesis supervisors: Dr. Juan Manuel López Soler and Dr. Pablo Ameigeiras Gutiérrez.

Guarantee, by signing this doctoral thesis:

that the research work contained in the present report, entitled ***Architecture, Modeling, Planning, and Dynamic Provisioning of Softwarized 5G Mobile Core Networks***, has been done by the doctoral candidate under the direction of the thesis supervisors and, as far as our knowledge reaches, in the performance of the work, the rights of other authors to be cited (when their results or publications have been used) have been respected.

Granada, 10th September, 2018.

Jonathan Prados Garzón
Ph.D Candidate

Dr. Juan Manuel López Soler
Full Professor of Telematics

Dr. Pablo Ameigeiras Gutiérrez
Tenured Professor of Telematics

To my Parents, Aurora and José Manuel,
my Sister, Raquel,
and my Nephew, Iván.

Acknowledgements

I would like to thank those people that in one way or another have contributed to carry out this work. Their support has been vital for the realization of this thesis.

First, I am most grateful to my supervisors, Prof. Juan M. López Soler and Prof. Pablo Ameigeiras Gutiérrez, for their guidance, valuable help, and confidence in me throughout the realization of this thesis. Prof. Pablo Ameigeiras has worked side by side with me all along the doctoral journey. He has taught me practically everything I know about scientific research. Prof. Juan M. López has always pursued my interests as if they were his own and given me wise advices. There are not enough words to express my gratitude to him for his patience, understanding, and words of encouragement during the last stages of this thesis, when I have been through tough times.

I would like to extend my gratitude to the other members of the group WiMuNet (Wireless and Multimedia Networking Lab) for their comradeship. Special thanks go out to Prof. Jorge Navarro Ortiz, Prof. Juan J. Ramos Muñoz, José A. Ordóñez Lucena, Pilar Andrés Maldonado (thank you very much for staying with me and helping me until the very last moment of every paper submission), and Óscar Adamuz Hinojosa for their invaluable assistance, responsiveness, and technical help. I would like to highlight that Prof. Juan J. Ramos Muñoz and Pilar Andrés Maldonado have provided me with chocolate and sweets all along this journey, for which I would like to express my gratitude to them once again.

I am deeply indebted to the members of the MOSA!C Lab which is led by Prof. Tarik Taleb. During my research stay at Finland, they made me feel at home and helped me to grow both personally and professionally. I would like to especially thank Prof. Tarik Taleb who gave me the chance to work in his lab and the freedom to work on what I wanted. Special thanks go also to Dr. Miloud

Bagaa (When I grow up, I want to be like you), Abdelquoddouss Laghrissi, Abderrahmane Boudi, Hamed Hellaoui, Ibrahim Afolabi, Illias Benkacem, and Oussama Bekkouche who provided me countless fruitful discussions during my research stay.

Another shout of gratitude goes out to my friends with whom I have had a lot of fun and unforgettable moments during the realization of this thesis. I want to thank Iván López Espejo for his unconditional friendship, wise advices, and interesting discussions. Also distinguished thanks to the Papitos group comprised of José Manuel Albarrán Vázquez (MiJoseManuel), José Manuel Melero Martínez (Poxito), Rafael Yáñez Figueroa (Zarapeta), and Raúl Quesada Vázquez (Rubico) who add a pinch of delicious madness to my life that has yielded a myriad of great memories. I am also particularly thankful for their company, generosity, and good times over a substantial part of this thesis lifetime to María Romero Garzón, Carlos Rivero García, Fernando Rivero García, Jonathan Molinero Castillo, Paola María López Álvarez, Sonia Melero Martínez, Teresa Ladrón De Guevara Castro, Rocío Rodríguez Tamayo, Ángel José Calero Molina, and Marimar Ponferrada Vivanco. Of course, I cannot forget my dear friend Manuel Rodríguez Prieto. I know I can always count on him.

Last, but not least, loving thanks to my parents, sister, and nephew who have always expressed me their unconditional support and love.

Abstract

Today's 4G mobile networks offer poor scalability, flexibility, elasticity and cost effectiveness due to the current networking approach which relies on proprietary and vertically integrated hardware, management operating systems, and control features that offers limited or no programmability. Moreover, the monolithic architecture and one-size-fits-all approach of current 4G networks are ill-suited to satisfy the diverse and unprecedented future service demands. In an attempt to fully meet the service and business demands of 2020 and beyond over a common network infrastructure and in an effective and cost-efficient manner, the attention of the mobile research community is now shifting towards what will be the next generation, the so-called fifth generation (5G).

To meet the above challenging goal, network softwarization (NetSoft) paradigm is envisaged as the cornerstone to build the 5G technology. The concept of NetSoft is mainly based on i) Network Functions Virtualization (NFV), which decouples network functions from proprietary hardware enabling them to run as software on virtualization containers like Virtual Machines, and ii) Software Defined Networking (SDN), which fully separates control and data planes in network nodes allowing network programmability. Under the NetSoft approach, isolated, fully automated, programmable, flexible, and service-customized networks known as network slices can be deployed on top of a common physical infrastructure. This is referred to as Network Slicing, which will allow the mobile operators to cover the different market scenarios and use cases which demand diverse requirements. Additionally, NetSoft promises to enable mobile operators to: i) reduce capital and operational expenditures, ii) accelerate time-time-to-market of new services, iii) foster innovation, iv) deliver agility and flexibility, and v) scale up/down services on demand.

As a result, the main objective of this thesis is to study the integration of NetSoft paradigm into the future 5G mobile network architectures and its application

to the automation of the network management.

First, an architecture for 5G mobile core networks based on SDN and NFV paradigms is proposed. The proposed architecture follows a partially virtualized approach, *i.e.*, the control plane functionalities are deployed as Virtualized Network Functions (VNFs) running on commodity hardware, whereas the user plane consists of SDN commodity switches. In contrast to current Third Generation Partnership Project (3GPP) mobile networks, the architecture proposal includes significant changes such as the removal of the GTP-U and the improved support of the internal communications (between devices attached to the same network). The mobility support for the softwarized architecture is addressed and a novel Handover procedure that relies on the OpenFlow protocol is defined. The proof of concepts carried out validate the feasibility of the softwarized architecture in terms of performance.

Second, a stochastic characterization of the future signaling, and traffic demands is performed. To that end, two compound traffic models are defined to emulate the traffic demands for the future 5G mobile networks. In addition, analytic expressions are derived to estimate the signaling workload from the compound traffic models and network setup. The results show that the aggregated signaling arrival process is roughly Poissonian, whereas the aggregated DP traffic arrival process exhibits Self-Similarity and Long-Range Dependence.

Third, simulation and analytic performance models for compositions of VNFs are developed. The analytic models are based on queuing networks. To solve the resulting network of queues, several techniques (*e.g.*, Jackson networks methodology, mean value analysis algorithm, and queuing network analyzer method) are studied and compared in terms of accuracy. The models developed are validated experimentally. To that end, a typical scenario for a 4G mobile network is considered, where the different functionalities are virtualized and interconnected through an SDN switch. The experimental procedures and the testbeds carried out are detailed. The validation results show that the analytical model proposed exhibits an estimation error lower than 20% to predict the response time of a composition of VNFs. This level of error is tolerable for resources dimensioning purposes.

Finally, based on the analytical models developed, integral solutions to automate the deployment and scaling of the softwarized mobile networks are pro-

posed. More precisely, we propose a solution for planning the virtualized mobile core networks, which is dubbed “Planner for the Evolved Packet Core (EPC) as a Service” (PES), and another solution for the Dynamic Resources Provisioning of the network services. The correctness of the operation of both solutions is validated by means of simulations. Additionally, their time complexity and degree of optimality are also assessed.

Contents

Declaration of authorship **I**

List of Abbreviations and Symbols **XV**

List of Figures **XX**

List of Tables **XXIV**

1 Introduction and Motivation **1**

1.1 5G Mobile Systems: Motivation and Requirements 3

1.1.1 Service Groups and Requirements 3

1.2 Network softwarization paradigm 6

1.2.1 Software-Defined Networking (SDN) 7

1.2.2 Network Functions Virtualization (NFV) 10

1.3 Objectives of this Thesis 14

1.4 Dissertation Road-map 16

1.5 List of publications 18

2 Softwarized Mobile Core Network Architecture Design **21**

2.1 Basic LTE System Architecture 22

2.1.1 Evolved Packet Core (EPC) 23

2.1.2 Evolved-Universal Terrestrial Radio Access Network (E-UTRAN) 24

2.1.3 LTE User Plane Operation 25

2.2 Softwarized Architecture for the 5G Core 27

2.2.1 Main Network Entities 27

2.2.2 User Plane Operation 29

2.3 Mobility Support 30

2.3.1 OF-based Handover Procedure 30

2.3.2 X2 Interface Considerations 32

2.3.3 HO Procedure Remarks 33

2.4 Architecture Analysis 34

2.4.1 Qualitative Comparison with the Long-Term Evolution (LTE) Evolved Packet Core (EPC) Architecture 34

2.4.2 Quantitative Evaluation 36

2.5 Conclusions 41

3 Modeling and Estimation of the Workload in Mobile Networks 43

3.1 Source Traffic Models 44

3.1.1 Source Traffic Models for MBB Services 45

3.1.2 Traffic Model for massive Machine Type Communication (mMTC) Services 49

3.2 EPC Signaling 50

3.3 Mobile Broadband (MBB) Workload Generation Model 53

3.4 Signaling Rates Estimation 55

3.4.1 λ^U_{SR} and λ^U_{S1R} Estimation 56

3.5 Compound Traffic Models 58

3.6 Aggregated Workload Characterization 59

3.6.1 LTE Workload Generator 59

3.6.2 Signaling Rates Validation 61

3.6.3 Aggregated LTE Workload Processes Characterization . . . 63

3.7 Conclusions 69

4 Performance Evaluation of a Three-Tiered vMME 71

4.1 Architecture and Operation of Three-Tiered Virtual Network Function (VNF)s 72

4.2 MME Processing Tasks 74

4.2.1 Service Request (SR) 75

4.2.2 S1-Release (S1R) 75

4.2.3 X2-based Handover (HO) 76

4.3 System Model 77
4.4 Performance Analysis of Three-Tiered VNFs 80
4.4.1 Queuing Model 80
4.4.2 Mean Response Time 81
4.4.3 Worker Tier Dimensioning 83
4.4.4 Scalability 84
4.5 Performance Evaluation of a virtualized Mobility Management Entity (vMME) with a Three-tiered Design 86
4.5.1 Experimental Setup 86
4.5.2 vMME Capacity 88
4.5.3 vMME Scalability 91
4.6 Conclusions 92

5 Performance Modeling of Softwarized Networks 95
5.1 Fundamentals of Queuing Networks 98
5.1.1 Open Jackson Networks 99
5.1.2 Closed Gordon-Newell Networks 100
5.2 Related Works 104
5.3 System Model 105
5.4 Analytical Model for Chains of VNFs 108
5.4.1 Queuing Model 108
5.4.2 System Response Time 110
5.5 Case Study: a Three-Tier vMME 113
5.5.1 Queuing Model for the vMME 114
5.6 Experimental Procedures 116
5.6.1 Experimental Setup 117
5.6.2 Parameter Estimation 119
5.6.3 Experiments 121
5.7 QNA Method Implementation Verification 122
5.7.1 Execution Time of the Queuing Network Analyzer (QNA) Methods 122
5.7.2 QNA Method Accuracy 124
5.8 Model Validation 124
5.8.1 Measured Input Parameters for the Model 125

5.8.2 Analytical Model Evaluation 125
5.9 Conclusions 128

6 Planning of the Virtualized EPC 129
6.1 Related Works 132
6.1.1 Modeling of the virtualized Evolved Packet Core (vEPC) . 132
6.1.2 Algorithms for the vEPC Embedding 134
6.2 System Model 135
6.2.1 System Architecture 135
6.2.2 Performance Requirements 137
6.3 Problem Formulation 139
6.4 LTE Control Plane (CP) Modeling 140
6.4.1 Transition Probabilities for the LTE CP Queuing Model . . 142
6.5 LTE Data Plane (DP) Modeling 144
6.6 PES: Planner for the EPC as a Service 145
6.7 Experimental Setup 148
6.7.1 Service Consumption and Spatial Distribution of the Users 149
6.7.2 E-UTRAN Deployment 151
6.7.3 Network Entities Instances Simulation 151
6.8 PES Validation and Runtime 152
6.8.1 PES Runtime 153
6.8.2 PES Validation 154
6.9 Conclusions 156

7 Dynamic Resource Provisioning of Softwarized Networks 159
7.1 Related Works 160
7.2 Dynamic Auto Scaling Algorithm (DASA) for Network Services . . 162
7.2.1 Problem Formulation for the Resources Dimensioning . . . 162
7.2.2 Dynamic Resource Provisioning 163
7.3 Study Case: Dynamic Provisioning of an E2E Network Slicing Orchestration System 166
7.3.1 5G!Pagoda project's Network Slicing Orchestration System (NSOS) 168
7.3.2 NSOS Queuing Model 169

7.4 Results 173
7.4.1 Experimental Setup 173
7.4.2 Performance of the Dimensioning Algorithm 175
7.4.3 Proper Operation Verification of the Dynamic Resource Provisioning (DRP) Algorithm 178
7.5 Conclusions 181

8 Conclusions and Outlook **185**
8.1 Main Conclusions 185
8.2 Research Contributions 191
8.3 Future Work 192

Appendices **195**

A Resumen **197**
A.1 Introducción y Motivación 197
A.1.1 Motivaciones en la Definición de las Redes Móviles 5G . . . 198
A.1.2 Limitaciones de las Redes Móviles Actuales 199
A.1.3 Paradigma de Softwarización de la Red 201
A.2 Objetivos 202
A.3 Conclusiones 204

Bibliography **214**

List of Acronyms and Symbols

Acronyms

3GPP	Third Generation Partnership Project
AAP	Application Activity Period
ANRF	Automatic Neighbour Relation Function
API	Application Programming Interface
AS	Access Stratum
BN	Backhaul Network
CAPEX	Capital Expenditure
CN	Core Network
COTS	commercial-off-the-shelf
CP	Control Plane
CPU	Central Processing Unit
cPGW	CP functionality of the PGW
cSGW	CP functionality of the SGW
DASA	Dynamic Auto Scaling Algorithm
DB	DataBase
DC	Data Center
DL	Downlink
DLT	Device Location Table
DP	Data Plane
DPGW	Data Plane Gateway
DRB	Data Radio Bearer
DRP	Dynamic Resource Provisioning
DSeNBs	Domain-Specific eNBs

DSNFVO	Domain-Specific Network Function Virtualization Orchestrator
DSO	Domain-Specific Orchestrator
DSRRO	Domain-Specific Radio Resource Orchestrator
DSSDN-C	Domain-Specific Software-Defined Networking Controller
DSVIM	Domain-Specific Virtualized Infrastructure Manager
E2E	End-to-End
EC	Edge Cloud
ECGI	Evolved Cell Global Identifier
eMBB	enhanced Mobile Broadband
EMM	EPS Mobility Management
eNB	evolved NodeB
ENE	Edge Network Element
EPC	Evolved Packet Core
EPCaaS	Evolved Packet Core as a Service
EPS	Evolved Packet System
ES	Edge Switch
ESM	EPS Session Management
E-UTRAN	Evolved-Universal Terrestrial Radio Access Network
fBm	fractal Brownian motion
FCFS	First-Come, First-Served
FE	Front-End
GO	Global Orchestrator
GPRS	General Packet Radio Service
GTP	GPRS Tunneling Protocol
GTP-U	GTP layer for the user plane
HO	Handover
HSS	Home Subscriber Server
HSS	Home Subscriber Service
HTML	HyperText Markup Language
IAST	Inter-Arrival Session Time
ICIC	Inter-Cell Interference Coordination
InP	Infrastructure Provider
IP	Internet Protocol
IPv4	Internet Protocol v4

KPI	Key Performance Indicator
KVM	Kernel-based Virtual Machine
LCFS	Last-Come-First-Served
LTE	Long-Term Evolution
M2M	Machine-to-Machine
MBB	Mobile Broadband
METIS	Mobile and wireless communications Enablers for 2020 Information Society
MME	Mobility Management Entity
MMPP	Markov-modulated Poisson process
mMTC	massive Machine Type Communication
MPLS	Multiprotocol Label Switching
MTC	Machine-Type Communications
MVA	Mean Value Analysis
NAS	Non-Access Stratum
NetSoft	Network Softwarization
NF	Network Function
NFV	Network Functions Virtualization
NFVI	NFV Infrastructure
NIB	Network Information Base
NIC	Network Interface Card
NIT	Neighbour Information Table
NS	Network Softwarization
NSOS	Network Slicing Orchestration System
OF	OpenFlow
OI	Output Interface
OPEX	Operational Expenditure
OS	Operating System
PCI	Physical Cell Identifier
PCRF	Policy and Charging Rules Function
PCRF	Policy and Charging Rules Function
P-GW	Packet Data Network Gateway
PGW	PDN (Packet Data Network) Gateway
PM	Physical Machine

PNF	Physical Network Function
pNIC	physical Network Interface Card
PS	Processor Sharing
QN	Queuing Network
QNA	Queuing Network Analyzer
QoE	Quality of Experience
QoS	Quality of Service
QT	Queuing Theory
RAE	Resource Awareness Engine
RAN	Radio Access Network
RR	Regional Router
RRC	Radio Resource Control
S1R	S1-Release
SAE	System Awareness Engine
SCV	Squared Coefficient of Variation
SDB	State DataBase
SDI	Software Defined Infrastructure
SDN	Software Defined Networking
SDNC	SDN Controller
SeNB	Source eNB
SGW	Serving Gateway
S-GW	Serving Gateway
SL	Service Logic
SLA	Service Level Agreement
SNR	Signal-to-Noise Ratio
SOA	Service-Oriented Architecture
SoftNet	Softwarized Network
SP	Service Provider
SR	Service Request
SRR	Service Release
TA	Tracking Area
TAU	Tracking Area Update
TeNB	Target eNB
TAU	Tracking Area Update

UE	User Equipment
UL	Uplink
UP	User Plane
URLLC	Ultra-Reliable and Low Latency Communications
vEPC	virtualized Evolved Packet Core
VM	Virtual Machine
vMME	virtualized Mobility Management Entity
VNE	Virtual Network Embedding
VNF	Virtual Network Function
VNFC	Virtual Network Function Component
VNFD	VNF Descriptor
VNFFG	VNF Forwarding Graph
vNIC	virtual Network Interface Card
VoIP	Voice over IP
vP-GW	virtualized P-GW
vS-GW	virtualized S-GW
W	Worker

List of Figures

1.1 Examples of usage Scenarios of IMT for 2020 and Beyond (extracted from [1]). 5

1.2 The importance of KPIs for the three main 5G service groups [2]. . 6

1.3 Comparison between traditional networks and software-defined networks. 8

1.4 The basic architecture of a software-defined network proposed by the Open Networking Foundation (ONF) [3]. 9

1.5 Vision for Network Functions Virtualization (NFV). 10

1.6 ETSI ISG NFV Reference Architectural Framework [4]. 11

2.1 Basic Evolved Packet System (EPS) architecture. 23

2.2 The LTE User Plane (UP) protocol stack [5]. 25

2.3 The overall EPS bearer service architecture [5]. 26

2.4 Envisioned Architecture for the 5G mobile core network. 28

2.5 Openflow-based Handover procedure. 31

2.6 Scenario to evaluate the HO execution time. 39

2.7 Handover Execution Time. 41

3.1 Example of Markov-modulated Poisson process (MMPP) with three states. 50

3.2 EPS control plane for Evolved-Universal Terrestrial Radio Access Network (E-UTRAN) access [6]. 51

3.3 MBB workload generation model for a single user. 53

3.4 Markov chain based model for social networking. 61

3.5 Control procedures arrival rates versus user inactivity timer. 62

3.6 Scenario used to study the aggregated workload generation process for a population density of 500 users per km^2. 64

3.7 Variance coefficient α and Hurst parameter H measurements versus the number of users N_U for the aggregated DP arrival process. 65

3.8 CDF of the signaling procedures inter-generation times in the mobile network. 67

3.9 Generation rate per User Equipment (UE) for the different LTE signaling procedures versus the population density for the considered scenario. 68

4.1 Architecture and operation of a three-tiered VNF. 73

4.2 UE triggered SR procedure [7]. 74

4.3 S1R procedure [7], [8]. 76

4.4 X2-based HO procedure - Completion Phase [7], [9]. 77

4.5 Overall system model. 78

4.6 Feedforward open queuing network modeling a vMME with a three-tiers architecture. 80

4.7 State transition diagram for the birth-death process of an M/M/m queue. 82

4.8 Capacity of the vMME *versus* the number of Worker (W) instances. 89

4.9 Mean response time of the vMME versus the number of users. . . . 90

4.10 Cost per second versus number of users. 92

4.11 Scalability $\Psi(k)$ versus number of vMME Service Logic (SL) instances. 92

5.1 Performance modeling process [10]. 96

5.2 Modeling of user sessions in a closed network. For simplicity, only one queue is used to capture the behavior of each node of the system, though more complex models could be considered for the system. 104

5.3 Chain of VNFs that is composed of the VNFs *X*, *Y*, and *Z*. VNFs *X*, *Y*, and *Z* have respectively 3 (*e.g.*, *X1*, *X2*, and *X3*), 2 (*e.g.*, *Y1*, and *Y2*), and 1 (*e.g.*, *Z1*) VNFCs. VNFCs *Y2* and *Z1* have respectively 3 and 2 instances. 106

5.4 Top of Rack switch interconnecting two Physical Machines (PMs). Each PM might host several Virtual Machines (VMs) which are in turn interconnected through a virtual switch. 107
5.5 Queuing model for the chain of VNFs shown in Fig. 5.3. 109
5.6 Queuing model for the vMME with a three-tier design shown in Fig. 4.1. 114
5.7 Experimental setup to validate the performance model for chains of VNFs. 118
5.8 The service time samples of a W instance when the real-time configuration is enabled and disabled. 119
5.9 Mean response time of the three-tiered vMME estimated by using different approaches for analyzing networks of queues and measured by simulations. 123
5.10 Comparison between the estimation error of the different methodologies for analyzing networks of queues. 124
5.11 Service time process for each Virtual Network Function Component (VNFC). 126
5.12 Overall mean system response time. 127
5.13 Model validation. 127

6.1 Network slicing concept: several network slices tailored for different use cases running on a common substrate infrastructure [11]. . 130
6.2 Domains of a mobile network. 131
6.3 E-UTRAN deployment and the workload distribution among the Edge Clouds (ECs) sites. 136
6.4 Assumed LTE network architecture. 137
6.5 LTE CP queuing model. 141
6.6 E-UTRAN deployment for a population density of 500 users per km^2. 151
6.7 E-UTRAN deployment for a population density of 1000 users per km^2. 152
6.8 PES execution time. 153
6.9 Total number of dedicated CPU instances and virtualization containers. 154

6.10 Total number of dedicated CPU instances for the DP. 155
6.11 95 percentile of the CP workload for the most loaded EC. 155
6.12 Maximum DP workload for the most loaded EC. 156
6.13 Mean response time of the CP. 156
6.14 Quality of Service (QoS) of the DP. 157

7.1 Main functional blocks of the DRP solution for a network service. . 164
7.2 5G!Pagoda project's Slice Orchestration Request call-flow. 170
7.3 5G!Pagoda project's Slice Resources Allocation and Path Creation call-flow. 171
7.4 Queuing model of the 5G!Pagoda's project NSOS. 172
7.5 The temporal distribution of the aggregated mean arrival rate at the network services (*e.g.*, NSOS -slices orchestration requests per second- and vEPC CP -LTE signaling procedures per second-) for one day period [12]. 175
7.6 Comparison of the required computational resources estimated by our proposed dimensioning algorithm and the optimal solution. . . 175
7.7 Comparison of the overall mean response time of the system achieved by our algorithm and the optimal solution. 177
7.8 Resources dimensioning algorithm time complexity. 177
7.9 Total number of processing instances required by the NSOS to met the delay budget (2 ms). 179
7.10 Number of processing instances allocated to each NSOS entity to met the delay budget (2 ms). 179
7.11 NSOS response time. 180
7.12 Number of dedicated CPU instances per entity of the vEPC CP. . 180
7.13 Mean response time of the LTE CP to move a UE from IDLE state to ACTIVE state. 181
7.14 Comparison between individual dimensioning per entity and joint dimensioning of all entities in terms of computational resources. . . 182

List of Tables

1.1 Enhancement of key capabilities from IMT-Advanced to IMT-2020 [2]. 7

2.1 Performance requirements for Software Defined Networking (SDN) Switches. 37
2.2 Parameters configuration to estimate the HO execution time. . . . 40

3.1 Web browsing source traffic model characterization. 46
3.2 Youtube source traffic model characterization. 47
3.3 Social networking source traffic model characterization. 48
3.4 Mobile instant messaging source traffic model characterization. . . 49
3.5 Compound traffic model for MBB and mMTC services. 59
3.6 Compound traffic model for MBB services. 60
3.7 RMSE for predicted arrival rate per device (see Fig. 3.5). 63

4.1 Parameters configuration. 86
4.2 Number of instructions executed by an Mobility Management Entity (MME) and the corresponding processing times in *m3.xlarge* instance for the different signaling messages. 88
4.3 Cloud service configuration and cost calculation. 91

5.1 Main notation for Queuing Networks (QNs). 98
5.2 Model input parameters. 110
5.3 Main Parameters Configuration for Simulations 122
5.4 Execution times for solving the queuing network that models a vMME. 123

5.5 Measured input parameters for the model. 126
5.6 Characterization of the relative error for the estimation of the Squared Coefficient of Variations (SCVs) of the internal arrival processes. 126

6.1 Parameters Configuration of the LTE simulator 149

7.1 Primary notation used in Algorithm 5. 167
7.2 Transition probabilities between entities for the NSOS queuing model. 173
7.3 Parameters configuration. 176

Chapter 1

Introduction and Motivation

Cellular systems have radically changed the way people communicate. Since the introduction of mobile communications in the early 1950s in Europe, US and Japan [13], they have evolved at an astonishing pace, moving from elementary analog communication systems to today's complex, high speed and all-IP fourth generation (4G) systems. Nowadays, mobile networks keep evolving, though they already offer high spectral efficiency, high data rates, low latency, efficient and high speed mobility support, and strong security mechanisms to give support to mobile broadband (MBB) services. Despite this, the monolithic architecture and one-size-fits-all approach of today's 4G networks are ill-suited to satisfy the diverse and unprecedented future service demands. In an attempt to fully meet the future service needs, the attention of the mobile research community is now shifting towards what will be the next generation, the so-called fifth generation (5G).

In addition, today's mobile networks offer a poor scalability, flexibility, elasticity and cost effectiveness due to the current networking approach which relies on proprietary and vertically integrated hardware, operating systems, and control features that offers limited or no programmability [14]. More precisely, current mobile networks have the following limiting features:

1. **Coupling between network functions and vendor-dependent proprietary hardware**. Today's networks consists of a large variety of proprietary hardware networking devices, each implementing specific network functions (e.g., routing, load balancing, mobility management, transcoding,

firewalling, deep packet inspection, etc.). These devices relies on specialized, vendor-specific, hard-to-configure equipment, highly dependent of the software that defines the network function(s). When a network improvement upgrade or a new service is required, network operators must acquire new expensive proprietary hardware devices and find room and power supply for them. Moreover it is necessary highly qualified network managers to operate the increasingly complex infrastructure.

Because of the above mentioned current mobile networks are statically and long-term dimensioned during the planning phase to cope with the peak workload demand expected for the next years. Once the traffic demand is close to the network capacity limit (*e.g.*, 70% of its utilization), the hardware-based network equipment will be upgraded to meet the future demands. This *modus operandi* leads to a waste of resources, as most of the time the network is overdimensioned. It also requires network operators to invest heavily in infrastructure during the network deployment, and to expect network will remain operational for ten years or more to ensure the return on investment is reasonable. Last, the mobile network availability might be disrupted due to any unforeseen workload surge.

2. **Coupling between Control Plane (CP) and User Plane (UP)**. The UP refers to the networking devices such as routers and switches that are responsible for forwarding data, whereas the CP represents the protocols used to populate the forwarding tables of the networking devices [15]. In today's networks, the CP is decentralized and tightly coupled with the UP on every networking device. This approach has two major drawbacks. On one side, decision making and policy enforcement must be performed on a device-by-device basis. On the other side, it hampers the addition of new functionalities to the network as CP has to be individually modified on every device through the installation of new firmware and/or with hardware upgrade. Thus, the tightly coupling between UP and CP involves time-consuming, error-prone, and device-by-device network management tasks, leading to static and rigid networks.

The aforementioned limitations compel to revisit the design of current mobile networks, *i.e.*, it is necessary to re-architect them.

1.1 5G Mobile Systems: Motivation and Requirements

5G mobile networks are expected to play a paramount role in the global industrial digitalization by covering all the vertical market needs in a cost effective and efficient way. Given a historical 10-year cycle for every generation of cellular advancement, 5G networks are assumed to be deployed around 2020 [16]. Although the future is still uncertain, there is a wide consensus on the view that 5G will not only be a natural evolution of current mobile networks.

The trends that motivate the definition of 5G technology are the following [2] [17]:

- The explosive growth of mobile data traffic, which will increase more than 200-fold between 2010 and 2020, and about 20000 times from 2010 to 2030.
- The increasing adoption of the Internet of Things (IoT), whose number of connections will reach 7 billion in 2020.
- The continuous emergence of new services (e.g., 3D ultra-high definition video, mobile cloud, mobile health, augmented reality, Tactile Internet applications) and application scenarios (e.g., ultra-dense and high speed moving scenarios).

1.1.1 Service Groups and Requirements

5G technology has the ambition to accommodate a wide range of services and applications which demand diverse requirements. Several European Union (EU) funded projects [18–24] and standardization bodies [25–27] have defined pioneering use cases that have been key to determine the requirements of 5G networks. For instance, the Third Generation Partnership Project (3GPP) standardization body has identified over 70 different use cases [27]. Although we can find a large number of use cases for 5G, they have been broadly categorized into the following three groups [2]:

- **Enhanced Mobile Broadband (eMBB)**. MBB comprises human-centric use cases for access to multimedia content, services and data. Today's mobile networks has been designed and optimized to support MBB services.

In fact, 4G technology is usually referred to as MBB-driven networks. However, the continued increasing demand for MBB due to the explosive growth of mobile devices and the advent of enhanced multimedia & entertainment soon will exhaust the capacity of current mobile networks. The eMBB group include traditional MBB services and new application areas and requirements. Besides the required increase in capacity, 5G technology should provide seamless connectivity to the end user anytime, anywhere, regardless the mobile device used.

- **Massive Machine Type Communications (mMTC)**. Machine Type Communications (MTC) refers to devices communicating without human intervention. The mMTC group is characterized by a huge number of connected low-cost devices equipped with long-life batteries typically transmitting infrequent, small, and non-delay-sensitive data. The operation of current mobile networks is not optimized for this kind of applications. Additionally, current cellular systems offer a poor connection density (i.e., total number of connected and/or accessible devices per unit area). Future 5G systems are expected to overcome such limitations.

- **Ultra-Reliable and Low Latency Communications (URLLC)**. The URLLC group encompasses services with stringent requirements for some performance metrics such as throughput, latency, dependability and availability. This group includes remote medical surgery and wireless control of industrial manufacturing or production processes, among many others. Today's mobile networks cannot support these services.

The above three service groups impose diverging and conflicting performance constraints that difficult their coexistence into future 5G networks. Thus, their integration using a common infrastructure is broadly considered as one of the major challenges in 5G. Figure 1.1 depict some examples of usage scenarios for each service group defined above.

Based on the defined use cases, the Radiocommunication Sector of International Telecommunication Union (ITU-R) has already established a set of Key Performance Indicators (KPIs) and their targets for 5G networks, also referred to as IMT-2020 networks in ITU terminology, in [2]. Table 1.1 summarizes these

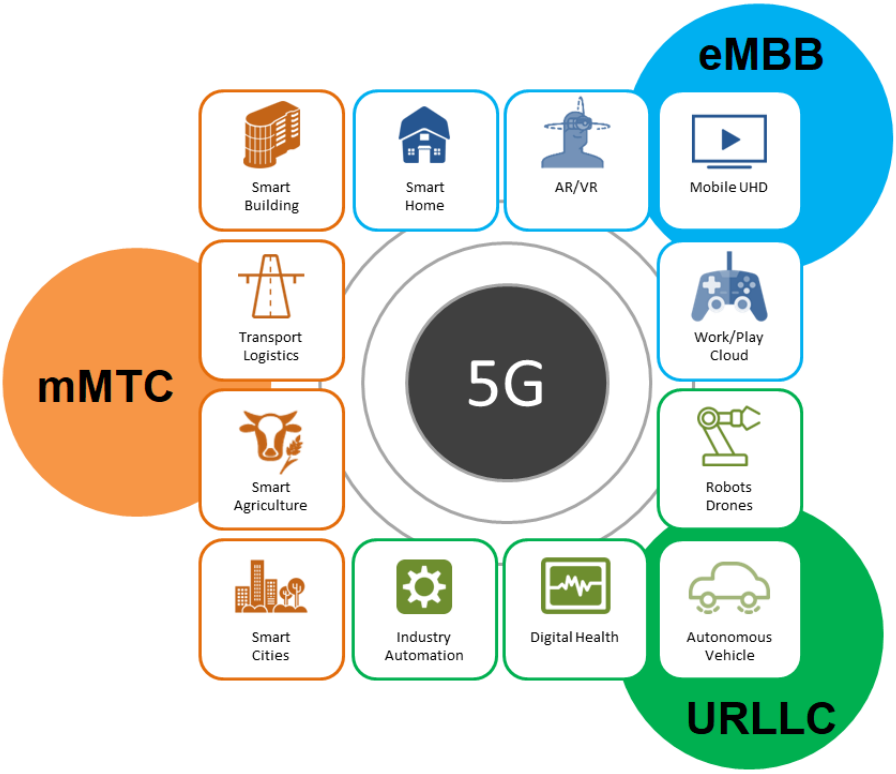


Figure 1.1: Examples of usage Scenarios of IMT for 2020 and Beyond (extracted from [1]).

performance targets and compares them with the IMT-advanced (4G) technology ones. As it is shown in Fig. 1.2, the relevance of the different KPIs differs for each service group.

Besides the performance requirements, 5G technology is expected to offer higher flexibility and scalability, and x100 increase in cost effectiveness compared to 4G networks [17]. Towards these goals, network softwarization (NetSoft) paradigm is envisaged as a key cornerstone for the future 5G mobile networks [28].

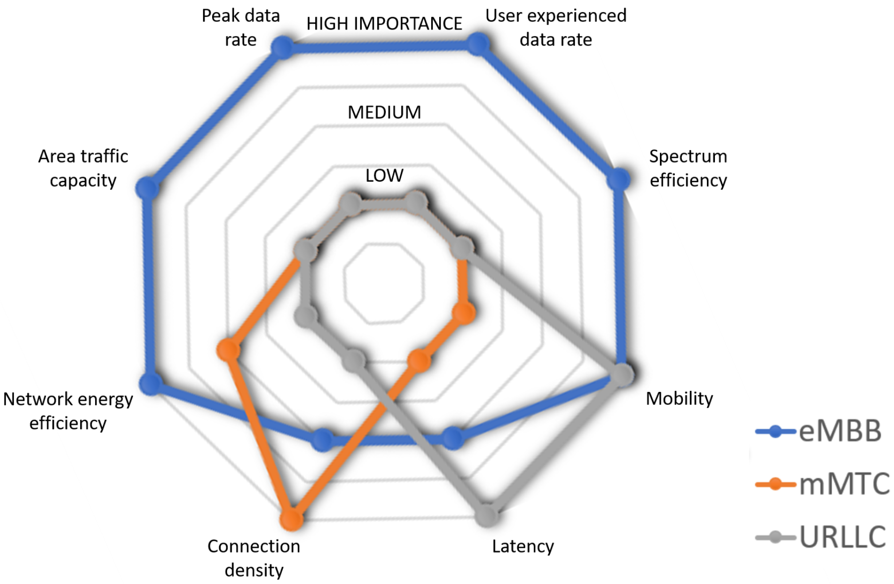


Figure 1.2: The importance of KPIs for the three main 5G service groups [2].

1.2 Network softwarization paradigm

Network Softwarization (NetSoft) paradigm is an overall approach for designing, implementing, deploying, managing, maintaining network equipment and/or network components by software programming [29]. NetSoft exploits the nature of software such as flexibility and rapidity all along the lifecycle of network equipment and/or components, for the sake of creating conditions that enable the re-design of network and services architectures, the optimization costs and processes, self-management and bring added values in network infrastructures [29].

At present, NetSoft is radically transforming the network concept and its adoption can be considered as one of the most relevant technical challenges for Telecommunications Industry and scientific community. Under the NetSoft approach, isolated, fully automated, programmable, flexible, and service-customized networks known as *network slices* can be deployed on top of a common physical infrastructure [11,30,31]. This is referred to as Network Slicing, which will allow the mobile operators to cover the different market scenarios and use cases which

Table 1.1: Enhancement of key capabilities from IMT-Advanced to IMT-2020 [2].

KPI	IMT-2020	IMT-advanced
Peak data rate ($Gbps$)	20	1
Area traffic capacity ($Mbps/m^2$)	10	0.1
Connection density ($devices/km^2$)	10^6	10^5
Latency (ms)	1	10
Mobility (km/h)	500	350
User experienced data rate ($Mbps$)	100–1000	10
Spectrum efficiency	3x	1x
Network energy efficiency	100x	1x

demand diverse requirements [32]. The key enablers of the Network Softwarization paradigm are Network Functions Virtualization (NFV) and Software Defined Networking (SDN).

1.2.1 Software-Defined Networking (SDN)

In today's networks, the CP is distributed and embedded into network devices (*e.g.*, switches and routers) [14]. Although this approach was adopted because it provides a fairly good network resilience [15], it also involves time-consuming, error-prone, and device-by-device network management tasks, leading to static and rigid networks. The SDN tendency emerges to overcome this limitation of the traditional networking approach.

SDN is based on four principles [33]: the decoupling of CP and UP, the logical centralization of the CP, the programmability of the network, and the use of open interfaces. SDN tendency fully separates control and user planes in network nodes allowing network programmability (see Fig. 1.3). The network devices may now be programmed by an external entity, such as an SDN Controller (SDNC) [34,35]. Specifically, in SDN, the CP of the network consists of a logically centralized SDNC implemented in software that controls a set of low-cost and simple network devices that make up the UP of the network. The SDNC runs on a single or

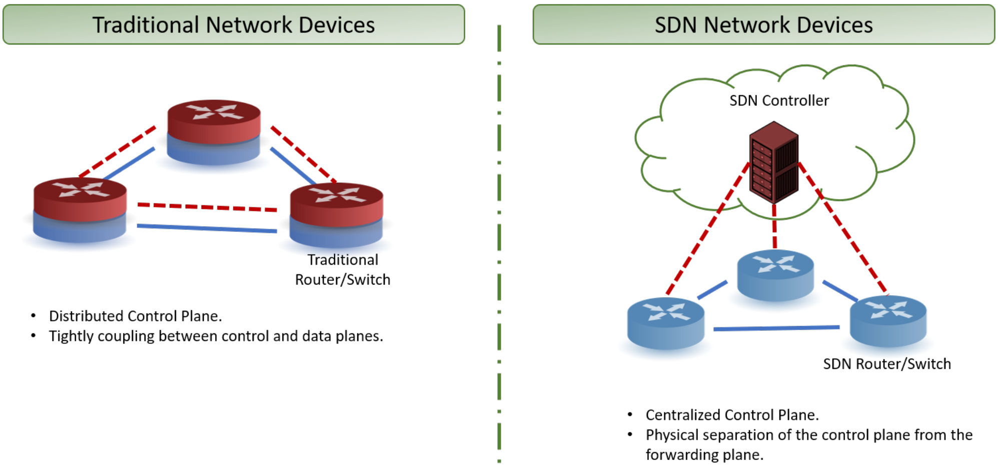

Traditional Network Devices
Traditional Router/Switch
• Distributed Control Plane.
• Tightly coupling between control and data planes.
SDN Network Devices
SDN Controller
SDN Router/Switch
• Centralized Control Plane.
• Physical separation of the control plane from the forwarding plane.

Figure 1.3: Comparison between traditional networks and software-defined networks.

cluster of servers, has a global view of the network, and makes traffic management decisions according to operational policies [14].

Figure 1.4 shows the layered reference model for a software-defined network proposed by the Open Networking Foundation (ONF), which is the main standardization body of SDN [34]. The architecture of a software-defined network is vertically split into three functional layers: infrastructure, control, and application. The infrastructure layer, which provides the UP functionality, consists of a set of network devices, which are configured and monitored by the SDNC via the Southbound interface. There are several protocols to implement the Southbound interface such as OpenFlow (OF) from ONF [36], OpFlex [37], or OnePk from Cisco. In SDN, network devices no longer need to understand and process a large number of protocols standards, but merely accept instructions from the SDNC. The SDNC plays the central role in the control layer. It provides a logically centralized network control through a set of network services. For SDN, a network service is a module that the controller has at its disposal to accomplish the required network control functionality. The SDNC provides an abstraction of the network to the application layer. Then, the SDN applications leverage network services and capabilities without being tied to the details of their implementation. Last, the application layer consists of SDN applications that allow SDN clients to

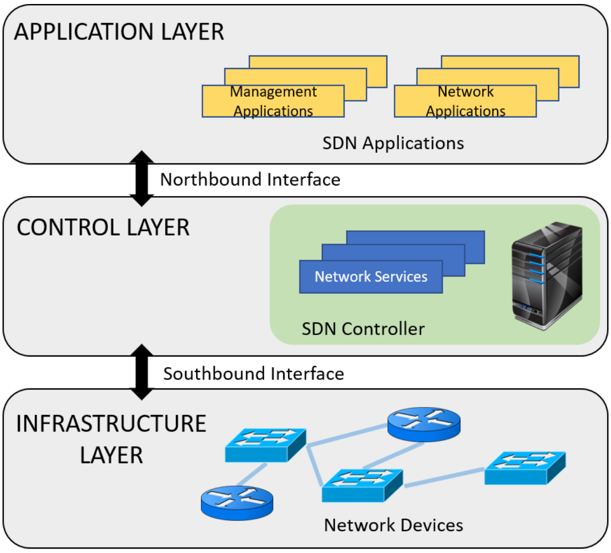


Figure 1.4: The basic architecture of a software-defined network proposed by the Open Networking Foundation (ONF) [3].

program the SDNC. Examples of SDN applications include network monitoring (*e.g.*, packet statistics), routing protocols, and policy enforcement, among many others.

SDN brings substantial benefits to operators, including [3]:

i) Cost savings. On one side, the SDN network devices (*e.g.*, switches and routers) are cheaper. On the other side, SDN enables operators to automate the complex, device-by-device, time-consuming, and error-prone configuration tasks, thus reducing the operational expenditures of the network.

ii) Acceleration of innovation. SDN relies on the use commodity hardware and open interfaces, and both of these features foster innovation. SDN eases and accelerate the inclusion of new network protocols and functionalities.

iii) Improvement of the user experience. By centralizing network control and making state information available to higher-level applications, an SDN infrastructure can better adapt to dynamic user needs.

iv) More granular service policies [38].

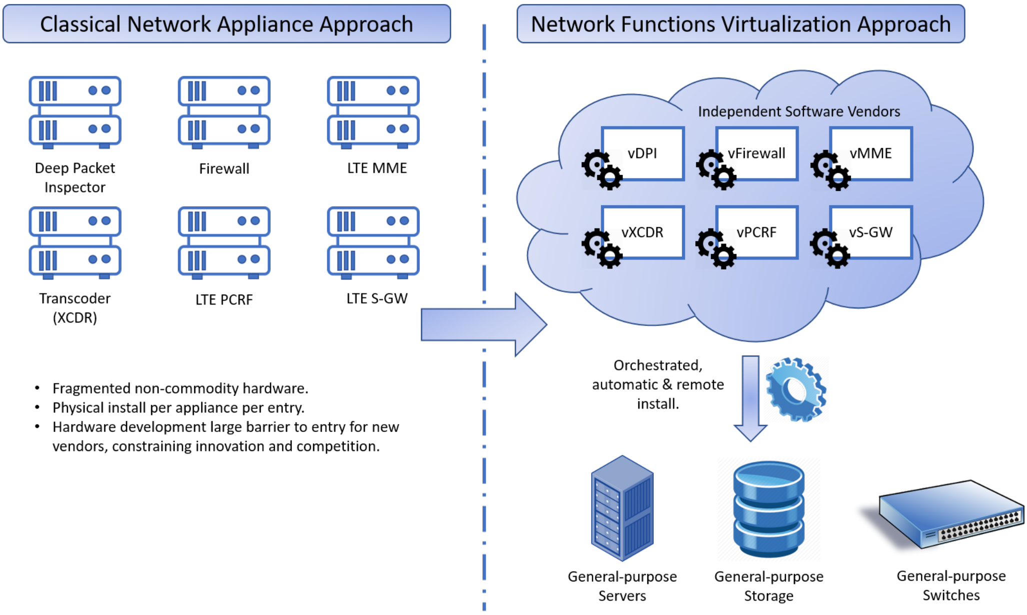

Classical Network Appliance Approach
Deep Packet Inspector
Firewall
LTE MME
Transcoder (XCDR)
LTE PCRF
LTE S-GW
Fragmented non-commodity hardware.
Physical install per appliance per entry.
Hardware development large barrier to entry for new vendors, constraining innovation and competition.
Network Functions Virtualization Approach
Independent Software Vendors
vDPI
vFirewall
vMME
vXCDR
vPCRF
vS-GW
Orchestrated, automatic & remote install.
General-purpose Servers
General-purpose Storage
General-purpose Switches

Figure 1.5: Vision for Network Functions Virtualization (NFV).

1.2.2 Network Functions Virtualization (NFV)

As mentioned earlier in this chapter, in today's networks there is a strong coupling between network functions (*e.g.*, firewalling, load balancing, routing, mobility support, policy enforcement, deep packet inspection, traffic optimization, etc.) and vendor-dependent proprietary hardware. This makes the cost of any network upgrade or the inclusion of a new service prohibitive in today's networks. On the one hand, every time a network improvement upgrade or a new network service is required, network operators have to acquire new expensive proprietary hardware devices, and find room, power supply and cooling systems for them. On the other hand, it is necessary highly qualified personnel with the required skills to design, integrate, and operate an increasingly complex infrastructure [39]. Thus, the current approach hinders operators from introducing new network features and services with agility and reduced costs.

NFV paradigm arose as an effort to overcome the aforementioned issues. NFV decouples network functions from proprietary hardware enabling them to run as software components, which are called Virtual Network Functions (VNFs), on

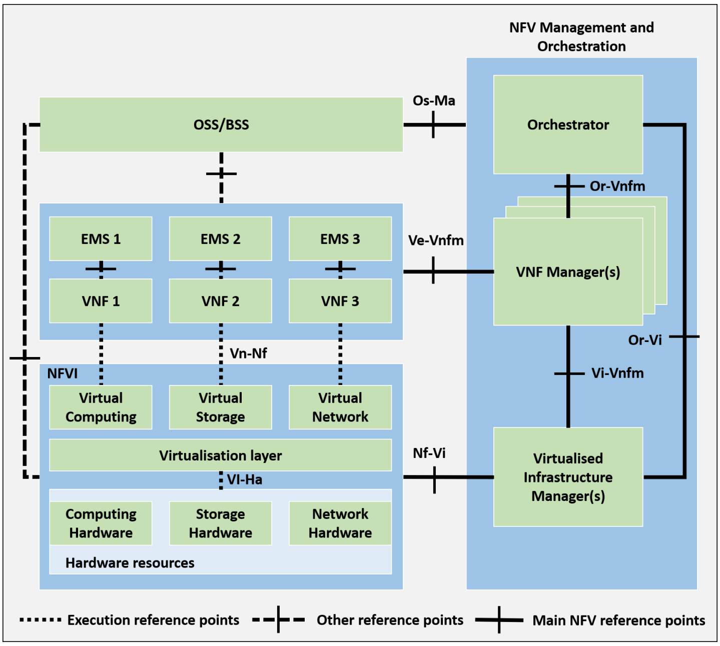


Figure 1.6: ETSI ISG NFV Reference Architectural Framework [4].

commodity servers [39, 40] (see Fig. 1.5). Figure 1.6 shows the NFV reference architectural framework standardized by ETSI ISG NFV [4]. The main functional blocks of the NFV reference architectural framework are briefly described below:

- **NFV Infrastructure (NFVI)**: The combination of both hardware and software resources which build up the environment in which VNFs are deployed, managed, and executed. The physical resources mainly consists of commodity computing and storage hardware, and commodity network devices and links (network hardware) that provide processing, storage, and connectivity to VNFs. These resources are abstracted through a virtualization layer (a hypervisor or virtual machine monitor) yielding virtual computing, storage and network resources. From the VNFs' perspective, the underlying physical infrastructure looks like a single entity providing them with desired virtual resources.

- **Virtual Network Functions (VNFs)**: The software implementations of the network functions that have well-defined external interfaces and functional behavior, and run over the NFVI. A VNF might consist of several Virtual Network Function Components (VNFCs), each performing a well-defined part of the VNF functionality [35]. In turn, a VNFC might have several instances. Each VNFC instance is hosted in a single virtualization container like a Virtual Machine (VM) or an Operating System-level container.

- **Element Management System (EMS)**: This block is responsible for FCAPS (Fault, Configuration, Accounting, Performance, and Security) management functionalities for a VNF [41].
- **Operations Support System/Business Support System (OSS/BSS)**: The collection of systems and management applications that service providers use to operate their business and network services [41]. A network service is a set of interconnected network functions (either virtualized -VNFs- or non-virtualized -Physical Network Functions (PNFs) [42]-).
- **NFV Management and Orchestration (NFV MANO)**: The collection of all functional blocks and data repositories used by these functional blocks, and reference points and interfaces through which these functional blocks exchange information for the purpose of managing and orchestrating NFV [41]. The NFV MANO mainly consists of the following main functional blocks:
    - **NFV Orchestrator**: This functional block is in charge of the lifecycle management of one or more network services, and the coordination in the management of their constituents VNFs and underlying NFVI resources.
    - **VNF Manager**: This block is responsible for the lifecycle management of one or more VNFs (*e.g.*, instantiation, update, query, scaling, termination).
    - **Virtualized Infrastructure Manager (VIM)**: This functional block is responsible for the management of the NFVI resources.

The benefits of NFV paradigm include:

i) Reduced Capital Expenditure (CAPEX). The CAPEX is reduced thanks to the use of general-purpose equipment and the pay-as-you-go billing models. On the one hand, commodity hardware is cheaper than special-purpose hardware, as it is mass-produced. On the other hand, the pay-as-you-go model for the Infrastructure as a Service offered by cloud providers prevents network operators from heavily investing in infrastructure during the network deployment. Moreover, the network has not to be overprovisioned anymore, which leads to a better resources utilization. This also will ease the market entry for new operators, thus increasing the competition on the market and benefiting the customer.

ii) Reduced Operational Expenditures (OPEXs). On one side NFV enables operators to fully automate the network management tasks such the deployment and scaling of network services and functionalities. On the other side, NFV reduces the power consumption through workload and equipment consolidation.

iii) Greater elasticity, flexibility, and scalability. NFV enables operators to adapt the resources of the network automatically depending on the workload with great agility and reduced costs, while guaranteeing that a set of performance requirements are always fulfilled. Additionally, the network operators has more flexibility to decide where deploy a VNFs. For instance, the operator could deploy an instance of a given network service at every point of presence available in order to improve the Quality of Service (QoS) (reduced propagation latency).

iv) Greater agility to deploy new network services and functionalities. Under NFV, services can be created and terminated in a matter of minutes without human intervention. Moreover, NFV allows to run production, test and reference facilities on the same underlying infrastructure. Then, it provides much more efficient test and integration, which leads to reduced development costs and time-to-market.

1.3 Objectives of this Thesis

NetSoft paradigm is expected to play a paramount role in 5G mobile networks. On the one hand, under the NetSoft approach, isolated, fully automated, programmable, flexible, and service-customized networks known as network slices can be deployed on top of a common physical infrastructure. This will allow the mobile operators to cover the different market scenarios and use cases which demand diverse requirements, which is one of the main challenges of future 5G networks. On the other hand, NetSoft promises to enable mobile operators to:

i) reduce capital and operational expenditures,

ii) accelerate time-time-to-market of new services,

iii) foster innovation,

iv) deliver agility and flexibility, and

v) scale up/down services on demand.

In this vein, the main objective of this thesis is to study the integration of NetSoft paradigm into the future 5G mobile network architectures and its application to the automation of the network management. To that end, this thesis addresses the following specific objectives:

1. Design of an architecture for 5G mobile core networks based on SDN and NFV paradigms. This objective is decomposed into the following subobjectives:

   1.1 Review and definition of use cases for future 5G mobile networks..

   1.2 Design and analysis of the softwarized 5G mobile core network architecture. We will design a network architecture tailored to the needs of the 5G mobile networks. These needs will be shaped by the use cases studied in Subobjective 1.1. Moreover the following design principles will be adopted:

      - Hierarchical and distributed design to improve network scalability and to reduce the propagation delays, respectively.

- Subsidiarity principle, meaning that decisions should always be taken at the lowest possible level or closest to where they will have their effect. That is to reduce the latency and to reduce the workload of the network elements at aggregation layers.

1.3 Design of the mobility support for the softwarized 5G mobile network architecture. The mobility support is one of the most important features of the mobile networks. We will define a Handover (HO) procedure for the softwarized architecture and analyze the impacts of mobility support on the softwarized architecture.

1.4 Realization of proof of concepts to verify the feasibility of the architecture in terms of performance.

2. Performance modeling and evaluation of softwarized 5G mobile core networks. Within this objective we will develop and assess theoretical and simulation models to estimate the main performance metrics of the softwarized networks. This objective is subdivided into the following subobjectives:

   2.1 Design and development of analytical and simulation models. The theoretical models will be based on Queuing Theory (QT) or Network Calculus.

   2.2 Models validation. In order to validate the models, we will deploy a typical scenario for a 4G mobile network. We will virtualize the different network entities involved and employ SDN switches to interconnect them. It will also be necessary to design, develop, carry out testbeds to measure the estimation error of the models. Depending upon the results obtained, further refinement of the models may be required.

3. Design, implementation, and evaluation of solutions based on NFV and SDN that enable the automation of the management operations of the 5G mobile core networks, thus reducing the OPEX of the network. To that end, we will use the models developed in Objective 2. More precisely, this objective consists of the following two sub-objectives:

   3.1 Design and evaluation of a solution for the planning of 5G mobile core network slices. The solution will estimate the amount of computational, network, and virtual resources required by the core network to provide service within a given geographical area. Then, the solution will take the decision of where to allocate such resources considering a set of candidates Data Centers (DCs). The solution will guarantee that the main performance requirements defined by Third Generation Partnership Project (3GPP) for mobile networks are fulfilled.

3.2 Design and evaluation of a solution for the Dynamic Resource Provisioning (DRP) of softwarized networks. The solution will be able to adapt autonomously the amount of resources allocated to a network service to cope with a given traffic workload so that a set of performance requirements, specified in the form of a Service Level Agreement (SLA), are met.

1.4 Dissertation Road-map

The rest of this thesis is structured as follows:

Chapter 2. Softwarized Mobile Core Network Architecture Design. This chapter mainly focuses on the design of a 5G mobile network architecture applying the NFV and SDN tendencies. Moreover, this chapter lists some use cases, benefits and challenges for the adoption of network softwarization in cellular networks; reviews other architectures for softwarized mobile networks proposed in the literature; addresses the mobility support in softwarized mobile networks; and includes an analysis of the designed architecture and proof-of-concepts.

Chapter 3. Modeling and Estimation of the Workload in Mobile Networks. This chapter faces the modeling and estimation of the signaling and Data Plane (DP) traffic foreseen for the future 5G mobile networks. First, the main features of the traffic generated by enhanced Mobile Broadband (eMBB) and massive Machine Type Communication (mMTC) services are studied and its source traffic modeling addressed. Second, analytic expressions are derived to estimate the signaling workload from the DP traffic characteristics, users' behavior and mobility, and Radio Access Network (RAN) setup. Third, compound traffic models are proposed by aggregating and adapting source traffic models developed in the literature. The source traffic models used were derived directly from real traffic traces. The considered services within the compound traffic model account for more than 70% of the peak aggregate traffic in America mobile access networks. Last, the expressions to estimate the signaling workload are validated, and the aggregated signaling and data traffic workload generation processes in the future 5G mobile networks are characterized.

Chapter 4. Performance Evaluation of a Three-Tiered vMME. This chapter introduces the concept of VNF decomposition and tackles the scalability and capacity assessment of a Long-Term Evolution (LTE) virtualized Mobility Management Entity (vMME) with a three-tiered design. The scalability and capacity assessment is carried out by means of both numerical and simulation results.

Chapter 5. Performance Modeling of Softwarized Networks. A queuing theory based model for softwarized networks is developed in this chapter. Firstly, the standard methodologies of analysis in queuing theory for network of queues are properly revisited. Then, a detailed system system model for chains of VNFs is presented. Moreover, the model is described and particularized for a vMME with a three tiered architecture use case. Finally, the model is validated by simulation and experimentally for that use case.

Chapter 6. Planning of the Virtualized EPC. This chapter includes an integral framework for the network slice planning of the virtualized LTE Evolved Packet Core (EPC). This framework includes performance models of the whole LTE network, and an algorithm to jointly perform the resources dimensioning and network and the allocation of these resources among a set of candidates DCs. The correctness and performance of the solution is respectively verified and evaluated by means of simulations. Additionally, the algorithm implemented in the solution to perform the workload partitioning among the candidates DCs is compared with other baseline approaches in terms of amount of computational and network resources required, QoS, and workload imbalances.

Chapter 7. Dynamic Resource Provisioning of Softwarized Networks. This chapter presents a novel solution to carry out the DRP of Softwarized Networks (SoftNets). The proper operation of the DRP solution is validated by means of simulations for two different use cases:

i) Dynamic provisioning of an End-to-End (E2E) Network Slicing Orchestration System (NSOS) whose performance modeling is also addressed in this chapter.

ii) Dynamic provisioning of the virtualized Evolved Packet Core (vEPC) CP whose performance modeling is tackled in Chapter 6.

Chapter 8. Conclusions and Outlook. Finally, this chapter draws the main conclusions, and outline the main contributions of this thesis and the future steps.

Appendix A. Resumen. This appendix is a comprehensive summary written in Spanish in order to meet with the requirements imposed by the University of Granada regarding the drafting of the doctoral dissertation.

1.5 List of publications

The following publications have been produced as a result of the work in this thesis:

1. P. Ameigeiras, J. J. Ramos-munoz, L. Schumacher, J. Prados-Garzon, J. Navarro-Ortiz and J. M. Lopez-soler, ”Link-level access cloud architecture design based on SDN for 5G networks,” in IEEE Network, vol. 29, no. 2, pp. 24-31, March-April 2015.

   DOI: 10.1109/MNET.2015.7064899

2. J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado and J. M. Lopez-Soler, ”Latency evaluation of a virtualized MME,” 2016 Wireless Days (WD), Toulouse, 2016, pp. 1-3.

   DOI: 10.1109/WD.2016.7461500

3. J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado and J. M. Lopez-Soler, ”Modeling and Dimensioning of a Virtualized MME for 5G Mobile Networks,” in IEEE Transactions on Vehicular Technology, vol. 66, no. 5, pp. 4383-4395, May 2017.

   DOI: 10.1109/TVT.2016.2608942

4. P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, ”MME support for M2M communications using network function virtualization,” in IARIA The Twelfth Advanced

International Conference on Telecommunications (AICT 2016), 2016, pp. 106-111.

ISBN: 978-1-61208-473-2

5. P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "Virtualized MME Design for IoT Support in 5G Systems," in Sensors, vol. 16, no. 8, pp., August 2016.

   DOI: 10.3390/s16081338

6. J. Prados-Garzon, O. Adamuz-Hinojosa, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado and J. M. Lopez-Soler, "Handover implementation in a 5G SDN-based mobile network architecture," 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, 2016, pp. 1-6.

   DOI: 10.1109/PIMRC.2016.7794936

7. J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado and J. M. Lopez-Soler, "Analytical modeling for Virtualized Network Functions," 2017 IEEE International Conference on Communications Workshops (ICC Workshops), Paris, 2017, pp. 979-985.

   DOI: 10.1109/ICCW.2017.7962786

8. J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, P. Andres-Maldonado and J. M. Lopez-Soler, "Performance Modeling of Chains of Virtualized Network Functions Based on Queuing Theory with Experimental Validation," Submitted to IEEE/ACM Transactions on Networking, 2018.

9. Jonathan Prados-Garzon, Abdelquoddouss Laghrissi, Miloud Bagaa, Tarik Taleb, Juan Manuel Lopez-Soler, "A Complete LTE Mathematical Framework for the Network Slice Planning of the EPC," Submitted to IEEE Transactions on Mobile Computing, 2018.

10. Ibrahim Afolabi, Jonathan Prados-Garzon, Miloud Bagaa and Tarik Taleb, "Modeling and Dynamic Provisioning of a Scalable E2E Network Slicing Or-

chestration System," Submitted Submitted to IEEE Transactions on Mobile Computing, 2018.

11. Jonathan Prados-Garzon, Abdelquoddouss Laghrissi, Miloud Bagaa, and Tarik Taleb, "A Queuing based Dynamic Auto Scaling Algorithm for the LTE EPC Control Plane," Accepted on GLOBECOM 2018 - 2018 IEEE Global Communications Conference, Abu Dhabi, 2018.

Chapter 2

Softwarized Mobile Core Network Architecture Design

As stated in Chapter 1, the monolithic architectures and *one-size-fits-all* approach of today's mobile networks cannot meet the diversity and unprecedented demands of the future services. Moreover, they offer a poor scalability, flexibility and cost effectiveness due to the current networking approach. To overcome this limitations, Software Defined Networking (SDN) and Network Functions Virtualization (NFV) paradigms are considered as key cornerstones for the future 5G mobile networks. For this reason, recent architecture proposals for such networks have made use of these paradigms [38, 43–48].

In 2015 *Requena et al.* [45] anticipated that the integration of SDN and NFV in the mobile networks should be progressive to favor the migration from legacy networks. To that end, they proposed a 3-step migration scheme for the adoption of SDN and NFV technologies. In the fist step, the network entities are virtualized. In the second step, SDN paradigm is included, while still maintaining legacy nodes. In the last step, the network is fully SDN-compliant, and GPRS Tunneling Protocol (GTP) and legacy nodes at the User Plane (UP) of the core network such as the Serving Gateway (S-GW) are removed.

In this chapter we propose a softwarized architecture for the mobile core network and address its mobility support. The architecture proposal falls into the aforementioned last step for the adoption of the SDN and NFV concepts. Additionally, we evaluate the overall handover mean delay of our architecture by simulation in a particular scenario. In this way, we provide insights about that our softwarized architecture can fulfill the Control Plane (CP) delay budget [49] in 5G networks. The rest of the chapter is organized as follows. Section 2.1 briefly reviews the basic Long-Term Evolution (LTE) architecture. Section 2.2 introduces the proposed architecture and its UP operation. Section 2.3 addresses the mobility support in the proposed architecture. In order to show the benefits and to study the feasibility of the softwarized core network architecture, Section 2.4 provides a qualitative analysis of it along with the results of some initial proofs of concepts. Last, Section 2.5 draws the main conclusions.

2.1 Basic LTE System Architecture

The LTE release 8, so-called LTE, is the standard specified by the 3GPP on the way towards 4G mobile [50]. The LTE was often referred to as “4G”, but the true 4G technology is LTE release 10, so-called LTE-Advanced, since it is the first Third Generation Partnership Project (3GPP) specification that meets the ITU/IMT-Advanced requirements issued by the ITU-R for 4G mobile networks [51]. That is why many referred to LTE (release 8) as “3.9G”. Despite this, LTE-Advanced could not be considered a different technology, but the next major step in the evolution of LTE to reach the IMT-Advanced targets. The major improvements added in LTE-Advance were multiple carrier aggregation, which provides a wider bandwidth, and enhanced Multiple Input Multiple Output (MIMO) antenna techniques.

Besides the enhancements added in the radio technology, LTE standards include disruptive architectural changes compared to the predecessor mobile generations. One of these changes is the split of functionality between the Radio Access Network (RAN) and the Core Network (CN) (see Fig. 2.1). This separation brings some benefits like sharing the same CN among several radio access technologies.

The LTE RAN is known as E-UTRAN (Evolved Terrestrial Radio Access Network), whereas the LTE CN is referred to as the Evolved Packet Core (EPC). Jointly, the E-UTRAN and the EPC make up the EPS.

The E-UTRAN is in charge of all radio-related functionalities (*e.g.*, radio-resources scheduling, MIMO schemes, and coding, among others). For its part,

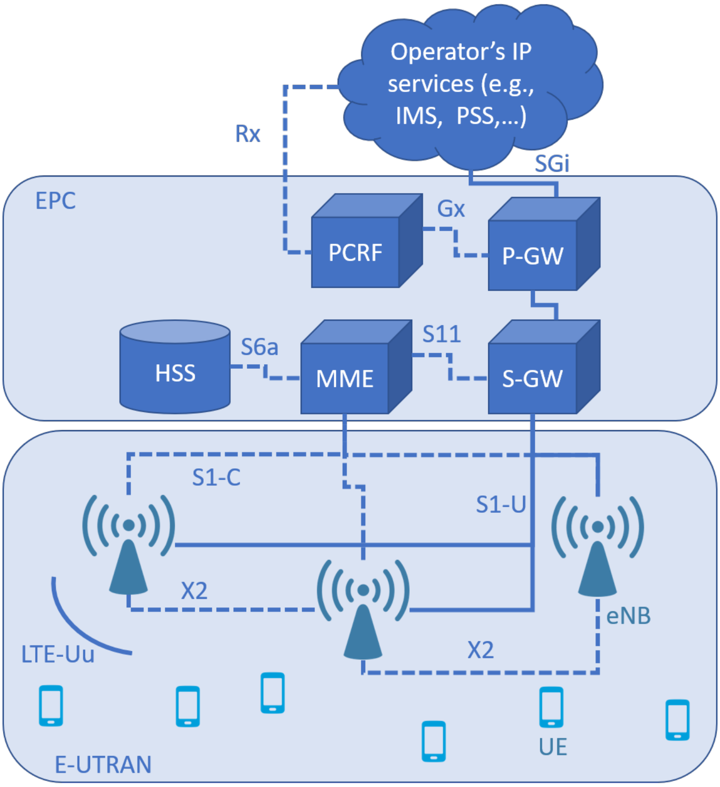


Figure 2.1: Basic EPS architecture.

the EPC is responsible for non-radio functions (*e.g.*, authentication, charging, and bearer setup, among others) that are required for providing a complete Mobile Broadband (MBB) network.

2.1.1 Evolved Packet Core (EPC)

The EPC is an all-Internet Protocol (IP) network supporting access to the packet switched domain only. The basic architecture of the EPC consists of five logical nodes (see Fig. 2.1) which are listed and briefly described below [5, 52]:

i) **Mobility Management Entity (MME)** is the central control entity of the EPC. It interacts with the evolved NodeB (eNB), the S-GW, and the Home Subscriber Service (HSS) within the LTE network to realize functions such as Non-Access Stratum (NAS) signaling, user authentication and authorization, mobility management (*e.g.* paging, user tracking), and bearer management, among many others.

ii) **Serving Gateway (S-GW)** connects the EPC to the Evolved-Universal Terrestrial Radio Access Network (E-UTRAN) at UP level. The S-GW acts as a mobility anchor when User Equipments (UEs) move between its attached eNBs (LTE base stations). The S-GW it is also responsible for collecting information for charging purposes.

iii) **Packet Data Network Gateway (P-GW)** connects the EPC to the operator's IP services at UP level. It is in charge of allocating the IP address for a specific UE, acting as mobility anchor for non-3GPP radio access technologies connected to the EPC, and the Quality of Service (QoS) enforcement according to the policy controlled by the Policy and Charging Rules Function (PCRF).

iv) **Home Subscriber Service (HSS)** is a database containing subscriber information such as the EPS-subscribed QoS profile and any access restrictions for roaming. It also holds information about the Packet Data Networks (PDNs) to which the user can connect and dynamic information such as the identity of the MME to which the user is currently attached.

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

2.1.2 Evolved-Universal Terrestrial Radio Access Network (E-UTRAN)

The E-UTRAN has a single type of logical node, the eNB that provides all the radio functionality in one or more cells (a typical eNB implementation is a three-sector site). As shown in Fig. 2.1, the eNB is connected to the EPC through the

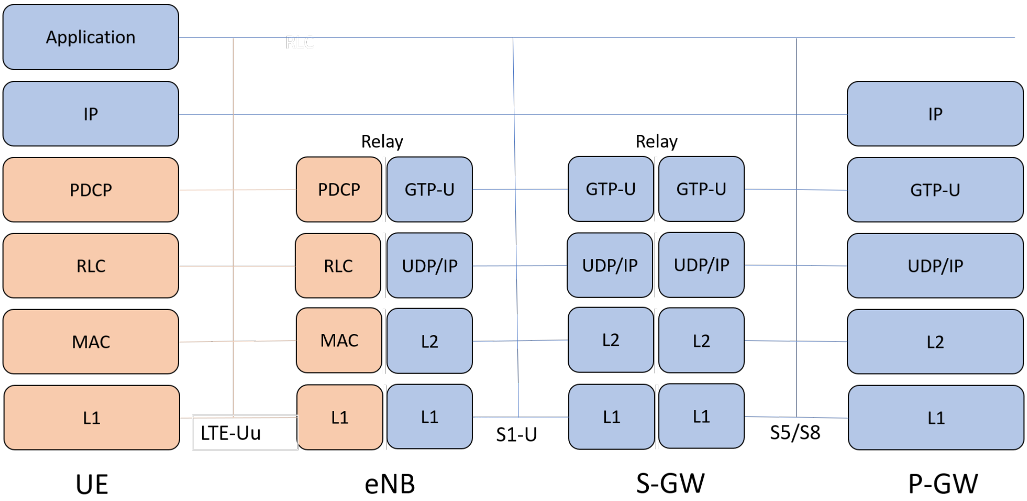


Figure 2.2: The LTE UP protocol stack [5].

S1 interface. More specifically, the eNB is connected respectively to the S-GW and MME by means of the S1-U (UP level) and S1-C (CP level). One eNB might be connected to multiple MMEs/S-GWs for purposes of load balancing and redundancy.

The eNBs are interconnected between them by means of the X2 interface. This interface is mainly used to support active-mode and lossless mobility, though it can be also used for supporting coordinated functionalities such as Inter-Cell Interference Coordination (ICIC) or Load Balancing among eNBs.

2.1.3 LTE User Plane Operation

Figure 2.2 shows the LTE UP protocol stack. Every IP packet destined to a UE is encapsulated in the GTP-U protocol and tunneled between the P-GW and the eNB [5]. Then, the packet is transmitted to the UE from the eNB through the radio interface encapsulated in the Packet Data Convergence Protocol (PDCP). The same process, but in the opposite direction, is carried out to transmit a packet in Uplink (UL) (i.e., from the UE to an external network).

The LTE network provides QoS support through the concept of EPS bearer. An EPS bearer refers to all the IP flows of a UE that have the same QoS class [53].

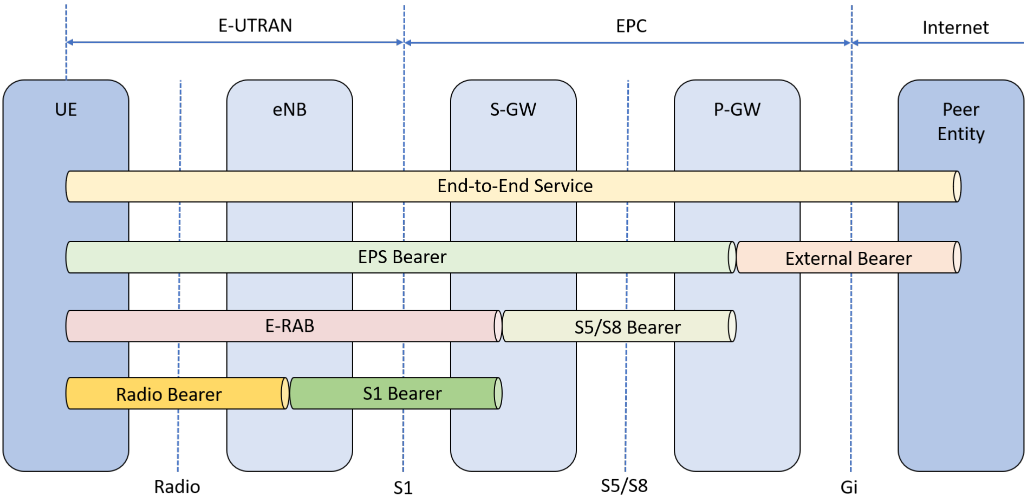


Figure 2.3: The overall EPS bearer service architecture [5].

Thus, all the packets of the IP flows that belongs to the same EPS bearer will receive the same treatment when they travel across the LTE network.

The overall EPS bearer service architecture is depicted in Fig. 2.3. An S5/S8 bearer transports the packets of an EPS bearer between a P-GW and a S-GW. The S-GW stores a one-to-one mapping between an S1 bearer and an S5/S8 bearer. The bearer is identified by the GTP tunnel ID across its different interfaces.

An S1 bearer transports the packets of an EPS bearer between an S-GW and an eNB, whereas a radio bearer transports the packets of an EPS bearer between a UE and an eNB. The E-UTRAN Radio Access Bearer (E-RAB) is the concatenation of an S1 bearer and a radio bearer. An eNB stores a one-to-one mapping between a radio bearer ID and a S1 bearer.

A default bearer is established for every UE during its first attachment to the LTE network. The default bearer establishment lasts until the UE detaches from the network. The default bearer comes with an IP address an provides best-effort delivery. When a user attempts to use a service (*e.g.*, VoIP) which requires higher QoS than that one offered by the default bearer, a dedicated bearer will be established on demand. Dedicated bearers are linked to a specified default bearer, thus they do not require separate IP addresses. Dedicated bearers can be Minimum Guaranteed Bit Rate (GBR) bearers or Non-GBR bearers, whereas the default bearer is always a Non-GBR bearer. Unlike Non-GBR bearers, GBR bearers offer dedicated network resources and a minimum bandwidth even when there is no traffic.

2.2 Softwarized Architecture for the 5G Core

This section introduces an architecture proposal for the future 5G mobile core network (see Fig. 2.4). We will adopt the same CP functional entities and operation as LTE [54]. The architecture follows a *partial virtualization* model [46]. That is, only the LTE CP functional entities (*i.e.*, LTE control plane) are implemented as Virtual Network Function (VNF)s running on commercial-off-the-shelf (COTS) servers, whereas the UP consists of SDN commodity switches.

2.2.1 Main Network Entities

The UP is implemented through a SDN-based link-level architecture. The UP comprises the eNBs (*i.e.*, base stations in LTE), the Backhaul Network (BN), and the Regional Router (RR). Additionally, there is an SDN switch for each eNB that interconnects the eNB with the BN (see Fig. 2.4), referred to as Edge Switch (ES). The RR is a high performance SDN router that acts as the mobility anchor and provides access to external networks. The ESs and RR entities are located at the edge of the core network, and they will be hereafter referred to as Edge Network Elements (ENEs).

Regarding the CP, there is an SDN Controller (SDNC) that is the interface between the CP and the UP. The CP network entities interact with the SDNC through the Northbound Application Programming Interface (API). Accordingly, these entities can be seen as network applications running on the top of the SDNC. The SDNC controls all the UP switches through the Southbound API (*e.g.*, OpenFlow). For instance, the SDNC establishes optimal routes between the ENEs by configuring the forwarding tables of the BN switches. This enables communication by each eNB and its attached UEs to the rest of the core network. Note that routes between pairs of eNBs can be established enabling the RR offloading.

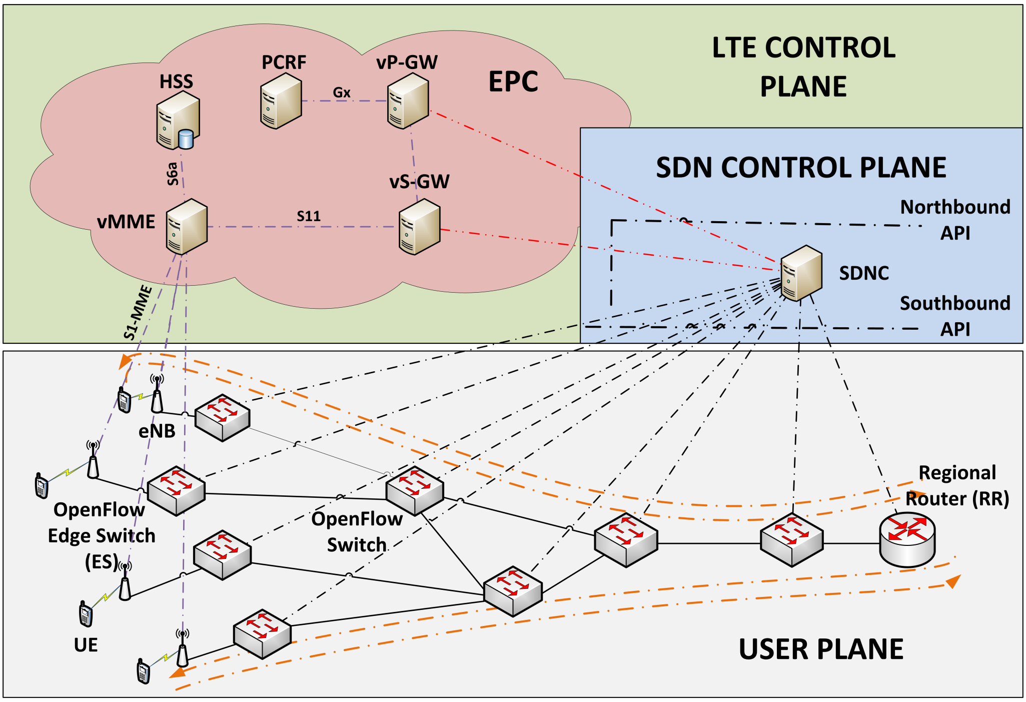


Figure 2.4: Envisioned Architecture for the 5G mobile core network.

To enhance the scalability and robustness of the CP, its virtualized entities might follow a *1:3 mapping* architectural design [46], inspired by web services. That is they are split into 3 logical components: Front-End (FE), Worker (W) (also referred to as Service Logic (SL)), and State DataBase (SDB).

The FE could be implemented with a SDN switch acting as a communication interface with other network entities and balancing the load among several SLs, which are in charge of processing the different control messages. The SDB stores the user session state making the SLs stateless. Therefore, the number SLs can grow without affecting on in-session users, while the whole entity is seen like a single component from the rest of the network.

Like in LTE, UEs are the terminals which allow each user to connect to the network via the eNBs. The UEs run the users' applications which generate or consume UP traffic. This process along with the user mobility trigger the LTE control procedures such as Handover (HO), Service Request (SR) or S1-

Release (S1R) [54].

2.2.2 User Plane Operation

The communications between the network elements are established through Multiprotocol Label Switching (MPLS) tunnels, which are handled by the SDNC. The SDNC is in charge to store and update device location information by defining a Device Location Table (DLT) that has an entry for each UE attached to the network. Each entry includes the UE IP address and the MPLS label that specifies the eNB/ES to which the UE is attached. An entry with the RR's BN interface IP address is also included. In other words, the MPLS label identifies the egress ENE to which the packet should be forwarded to reach the destination IP address.

Whenever a SR procedure is triggered, the SDNC creates a tunnel by adding entries in the flow table of the ENEs involved in the communication. As an example, assuming that there are no internal communications between UEs, *i.e.*, all the communications passes through the RR, the SDNC will create an entry in the RR flow table during every SR procedure. Just as the DLT entries, this entry matches the UE IP address with the MPLS label identifying the ES where the UE is attached to. Consequently, the RR will have a flow entry per each user to which packets have been destined recently (*i.e.*, users in ECM/Radio Resource Control (RRC)-Connected State). These entries will be deleted during service release and detach control procedures to avoid scalability issues (especially for Machine-Type Communicationss (MTCs)).

For every incoming packet, the ingress ENE pushes a MPLS header with a label field. The label identifies the egress ENE where the packet will be forwarded to. As mentioned, the ingress ENE has a lookup table that matches every destination IP with the corresponding egress ENE MPLS label. Once the MPLS header is pushed, the ingress ENE forwards the packet on the port towards the BN. Notice that these operations might be straightforward implemented by using the *push-MPLS* and *output* OF actions [55]. At every BN switch, the packet is processed and forwarded on the corresponding port according to its label.

Finally, the egress ENE simply pops the MPLS header and delivers the packet. If the egress ENE is an ES, the frame is passed to the radio link protocols for

Downlink (DL) transmission toward the destination terminal. If the egress ENE is the RR, the IP packet is passed to the IP layer. If any switch receives a packet whose destination address is not available in its flow table, the switch queries the controller for the corresponding entry, and the controller replies with the associated MPLS label. Please note that other tags could be used, though MPLS header is lightweight and its 20 bit label field enables a large number of routes.

It shall be noted that the SDNC is in charge to allocate MPLS labels for each ENE during network configuration. A routing application running on the SDNC computes and installs the routes for the BN. This application also monitors the links state of the BN.

2.3 Mobility Support

One of the most important features of the mobile networks is the ability to provide mobility support. In this vein, this section addresses the implementation of the HO procedure in a partially virtualized core network. The HO is the signaling procedure performed when a terminal moves from the coverage area of a Source eNB (SeNB) to the one of a Target eNB (TeNB).

In our case, OpenFlow (OF) protocol [55] is considered for the Southbound API because of its popularity and because it offers enough functionality to be used in the new architecture.

Even though the same CP as for LTE networks is assumed, the mobility support in such architecture entails some changes such as adding new processes and extension of functionality of the some LTE network entities. Additionally, the operation and update of the UP during HO procedure widely differs from LTE networks. The particular modifications and differences are described next.

2.3.1 OF-based Handover Procedure

Here we describe the HO procedure performed when a UE is in EMM-Registered and ECM/RRC-Connected States and moves from the coverage area of a SeNB to the one of a TeNB (Fig. 2.5). It is assumed that MME and S-GW are not relocated, thus the HO procedure addressed is equivalent to the X2-based HO for LTE networks [54]. It is also supposed that HO processes associated with the radio interface protocols are the same as in the LTE system.

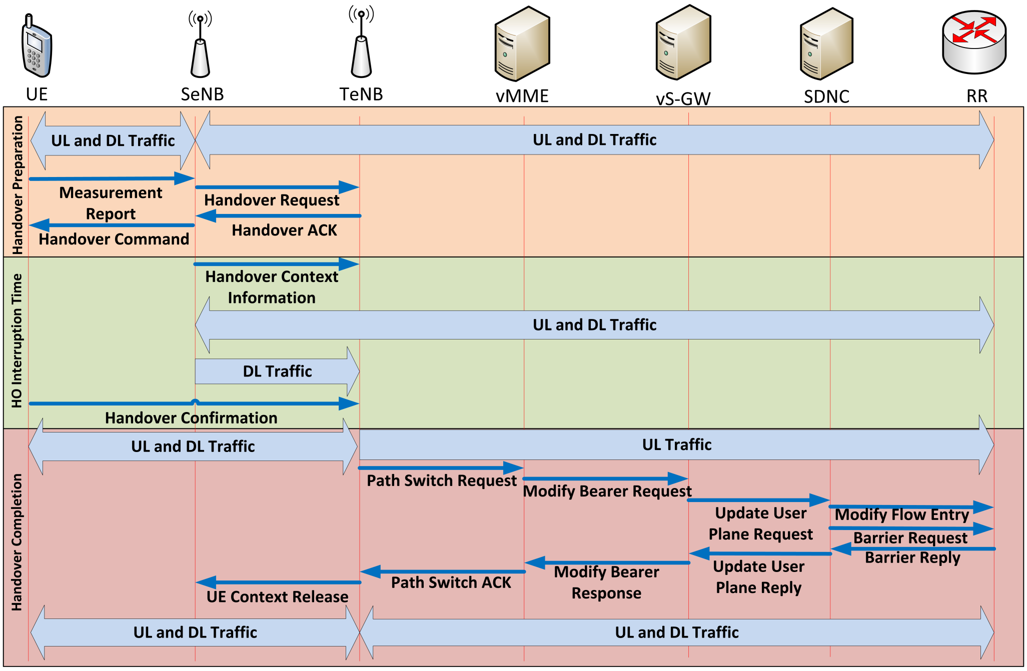


Figure 2.5: Openflow-based Handover procedure.

The main steps of the OF-based HO procedure are listed bellow (see Fig. 2.5):

- Firstly, when the signal level from TeNB overcomes a threshold, the UE sends a `Measurement Report` to the SeNB. Then, the SeNB makes the HO decision and forwards a `Handover Request` message to the TeNB.

- Secondly, the TeNB executes an admission control procedure to determine whether it has available resources to support the incoming UE. If the TeNB admits the UE, it acknowledges the HO sending `Handover ACK` message to the SeNB, which in turn confirms it to the UE.

- At this point, the SeNB begins a redirection procedure forwarding buffered and incoming downlink frames for the UE to the TeNB. To support lossless HO, the SeNB can provide the sequence numbers of the forwarded frames through the `Handover Context Information` message. At the same time, the `HO interruption time` takes place, where the UE carries out a synchronization process with the TeNB. During this period the UE cannot send or receive any data frame. Once the UE synchronizes with the TeNB, it sends the `Handover Confirmation` message to the TeNB. From this time on, the TeNB can directly send UE uplink frames to the RR.

- Next, the TeNB sends a `Path Switch Request` message to virtualized Mobility Management Entity (vMME) to notify that the UE has performed an eNB change. After receiving this message, the vMME informs the virtualized S-GW (vS-GW) that the downlink S1 bearer has been switched, and asks to switch the bearer path accordingly by sending a `Modify Bearer Request` message. After processing this message, the vS-GW sends an `Update User Plane Request` to the SDNC to modify the corresponding flow table entry of the RR, which acts as the mobility anchor. Once the SDNC concludes the operation, it generates the `Update User Plane Reply`, which is sent to the vS-GW to confirm the UP update. The vS-GW in turn acknowledges the Path Modify Bearer Request with *Path Modify Bearer Response* message.

- Finally, the vMME notifies the TeNB that the new path has been established with a `Path Switch Request ACK` message. The TeNB, in turn, sends a `UE Context Release` message to the source eNB. Now the source eNB can release radio and CP resources allocated for the UE and the HO procedure concludes.

2.3.2 X2 Interface Considerations

The eNBs are interconnected with each other through the logical X2 interface [54]. The same UP SDN-based network infrastructure is used to support the X2 interface. Each eNB stores the Neighbour Information Table (NIT) that relates the Physical Cell Identifier (PCI), Evolved Cell Global Identifier (ECGI) and the IP address of its neighbouring eNBs.

On the one hand, the eNB might use the LTE Automatic Neighbour Relation Function (ANRF) to discover automatically its adjacent eNBs and to establish the corresponding relation between ECGI and PCI [5]. On the other hand, the eNB might learn the IP addresses of its neighboring eNBs by requesting them directly to the SDNC. The SDNC needs to store a table, named as the Network

Information Base (NIB), that contains all the identifiers allocated for all of the network entities (*e.g.*, PCIs, ECGIs, IP addresses, MPLS labels).

Whenever an eNB requests an IP address of a neighboring eNB, the SDNC replies this message and adds an entry in the flow table of the ES associated with the requester eNB. This entry uses the destination IP address as match field and its value is fixed to the neighboring eNB IP address. Like UP case, the actions specified for this entry are push MPLS header and forwards the packet to the BN. In this way, the reachability between neighboring eNBs is enabled to support X2 interfaces.

2.3.3 HO Procedure Remarks

The S-GW functionality must be split into control and user planes. The control functionality is implemented by the vS-GW , while the UP functionality is implemented by the RR. Additionally, the control functionality at the vS-GW needs to be extended to allow the interaction with the SDNC through the northbound API. For instance, the vS-GW is able to generate the `Update User Plane Request` message (see Fig. 2.5), which contains the target eNB's ECGI and the IP address of the UE performing the handover. This entity also process the `Update User Plane Reply` message, which is a confirmation that the path switch has been performed correctly at UP.

The SDNC is in charge to carry out the UP update in the HO procedure. That is to modify the corresponding entry in the RR flow table. Whenever the SDNC receives an `Update User Plane Request` message, it sends an OF `Modify Flow Entry` message to update the corresponding entry at the RR. In addition, the SDNC sends an OF `Barrier Request` message to the Regional Router. That is to receive notification from the RR when the operation is completed.

The packets destined to SDNC are not associated with any flow at any OF switch of the network. Consequently, they will be encapsulated in an OF Packet `IN` message [55] (*i.e.*, our default action configured in case of OF table missmatch) and directly sent to the SDNC via the OpenFlow interface. The control messages sent to the virtualized control LTE entities (*e.g.*, vMME and vS-GW) do not need to passing through the SDNC controller. That is because MPLS tunnels are also employed to support connectivity between the primary network entities. In this regard, the virtualized control entities could be considered as ENEs.

Finally, the TeNB needs to buffer the DL data packets sent by the SeNB during the HO interruption time. To differentiate the packets forwarded by the SeNB from other packets, the SeNB can use the EXP field of the MPLS header. This allows the TeNB to prioritize the transmission of these packets in the air interface.

2.4 Architecture Analysis

2.4.1 Qualitative Comparison with the LTE EPC Architecture

The current centralized design of the mobile core network is not completely adequate to fulfill the latency, flexibility, cost-effectiveness, and scalability requirements foreseen for the 5G mobile networks. The softwarized architecture introduced in the present chapter leverages NFV and SDN concepts to overcome the limitations of today's mobile core network. Specifically, the presented softwarized approach helps to meet the aforementioned requirements as follows.

- *Costs savings.* NFV and SDN leads to large costs savings by embracing commodity hardware, and enabling the automation of the network management operations and the scaling of the network on demand.

- *Latency reduction.* The cheapening of provisioning costs makes viable a distributed mobile core network. This leads a radical latency reduction by moving the core network closer to the users. In addition, local breakouts can be deployed within the softwarized core network for accessing local services (such as content caching) with low delay.

- *Agility to deploy new services and network functionalities.* On the one hand, the use of open source VNFs running on commodity hardware provides flexibility to operators to rapidly incorporate new network features in response to changing business needs and user demands. Then, evolution of the network is no longer constrained by the relatively slow product cycles of vendor equipment, which may take several years. On the other hand, the softwarization paradigm allows to automate and standardize typical management operations like a service creation. This means that a skilled network manager is no longer needed to carry out some of the time-consuming and error-prone configuration and management tasks on a device by device basis every time a service has to be deployed.

- *Scaling network services on demand.* Today's cellular networks are over-dimensioned to face the expected increase in the traffic load over the next few years during the peak hours. Moreover, the network entities are statically deployed and configured. Hence, there is a lack of network elasticity to deal with the highly dynamic traffic patterns that results in a waste of resources. Since NFV tendency allows to create and scale network components on-demand, it can put an end to this problem. Then, the mobile operators could adapt and optimize their resources in accordance with the given traffic conditions.

Besides the previous benefits, the softwarized architecture introduced in this chapter has the following particularities:

- *GTP layer for the user plane (GTP-U) removal.* The EPC still heavily relies on GTP-U to achieve terminal mobility. However, GTP tunnel management is cumbersome when handovers frequently occur [5, 56]. Additionally, tunnels impair time-sensitive applications and introduce header overhead. On the contrary, the softwarized core network described in this chapter uses MPLS tunnels whose management and header overhead are more lightweight. For instance, considering Internet Protocol v4 (IPv4), the MPLS tunnels introduces only 4 bytes of overhead to every UP packet, whereas GTP-U adds an overhead of 36 bytes. The GTP-U header overhead jeopardizes the transmission efficiency, specially in MTC scenarios characterized by sensors infrequently transmitting low volume of information.

- *Core gateways offloading.* In the EPC, even internal communications passes through the core gateways (*e.g.*, S-GW and P-GW), as they act as the endpoints of every GTP-U tunnel, becoming the UP bottleneck of the network. The softwarized core network introduced in this chapter supports the creation of MPLS tunnels between any pair of ENEs, thus reducing the latency of internal communications. Additionally, this feature enables the deployment of local breakouts within the softwarized core network for accessing local services with low delay, as mentioned earlier.

- *Three-tiered CP entities design.* The design of the VNFs is substantial when exploiting all the advantages NFV offers. Accordingly, the softwarized mobile core network approach envisages a three-tiered design for the CP entities. A three-tier design offers a higher scalability, robustness, and flexibility compared to a single-tier design [46]. One of the main benefits of this approach is that it allows to automatically scale the processing capacity of the network entities without affecting on in-session users and the configuration of other network entities. As its main drawback, this approach increases the VNF response time, since every packet has to pass through several nodes. Despite this, the three-tier design is well-suited for CP entities because of their delay constraints are not as stringent as for the UP entities. In Chapter 4, we will study the response time and scalability of the CP entities following a three-tiered design.

Finally, although the softwarized approach entails great advantages, it also increases the processing delays and makes the real time operation of the network entities difficult. In contrast to the traditional approach, hardware and software are no longer developed together, but the VNFs should run on commodity hardware. On the one hand, commodity hardware, by default, is not optimized to perform the common functions of a network element such as the transmission or reception of network packets [57]. Moreover, the commodity network stacks and drivers struggle to keep up with increasing hardware speed [58]. On the other hand, the general purpose hardware does not natively support real time operation. This latency issue is one of the main challenges for the adoption of the Network Softwarization (NetSoft) paradigm.

2.4.2 Quantitative Evaluation

2.4.2.1 SDN Switches Throughput and Memory Consumption

This subsection includes an estimation of the performance requirements for the SDN switches considering a 5G scenario. The scenario is based on the dense urban

Table 2.1: Performance requirements for SDN Switches.

Memory consumption		
Element	**Requirement**	**Sample scenario**
BN switches	Flow tables: 1 entry per ENE (worst case 297 bytes/entry [55])	31 KB
RR	Flow table: 1 entry per active user (worst case 297 bytes/entry [55])	145 KB
SDNC	Device location table: 1 entry per UE and sensor	6.2 MB
Throughput		
Element	**Requirement**	**Sample scenario**
BN switches	For each packet flow matching, and packet forwarding.	Type-1 switch: 2.5 Mpackets/s Type-2 switch: 12.5 Mpackets/s
RR	For each packet DL: flow matching, push MPLS header, and packet forwarding. UL: pop MPLS header, and packet forwarding	DL:10.6 Mpackets/s UL:1.9 Mpackets/s

information society *TC2* and Massive deployment of sensors and actuators *TC11* test cases defined in the (Mobile and Wireless Communications Enablers for the TwentyTwenty Information Society) METIS project [59, 60]. The scenario considers the stringent requirements in terms of connections density and throughput demands for 5G mobile networks, according to the vision of the METIS project. More precisely, the scenario includes 100 eNBs distributed in a rectangular area of 0.215 km^2, 50000 users, and 300000 sensors. The BN network consists of 5 Type-1 switches, each of which aggregates the traffic of 20 eNBs, and 1 Type-2 switch, which interconnects the Type-1 switches with the RR. The traffic load is 150 Gb/s (85% DL, 15%UL).

Notice that the BN switches will have one entry per ENE in their flow tables at worst, whereas, assuming there is no internal communications, the RR will have an entry per active UE. Therefore, the RR flow table size L_{RR} will follow a binomial distribution,

$$f_{L_{RR}} = \binom{N_{UE}}{L_{RR}} \cdot {P_{UA}}^{L_{RR}} \cdot (1 - P_{UA})^{N_{UE}-L_{RR}} \tag{2.1}$$

where P_{UA} and N_{UE} respectively denote the probability that a user is active at a given time and the number of UEs in the network.

In order to check the feasibility and practicability of using the SDN technology to realize the UP of the softwarized mobile core network, the throughput and memory consumption requirements of the SDN switches were estimated given the scenario described above, see Table 2.1. The packet rate passing through each SDN switch was computed by assuming a packet size of 1500 bytes. For instance, a Type-1 switch will have to support a packet rate of $150\,Gbps/(1500\,bytes/packet \cdot 8\,bits/byte) = 2.5\,Mpackets/s$ in the considered scenario. Regarding the memory consumption, the BN switches will have an entry in their flow table per ENE. In the considered scenario there are 106 ENEs (100 eNBs, 1 RR, and 5 CP entities -vMME, vS-GW, virtualized P-GW (vP-GW), HSS, and PCRF). The SDNC stores the DLT that includes the IP addresses (IPv6 addresses were considered -128 *bits*-) of every user and sensor and the MPLS label (20 *bits*) associated with the ENE they are attached to. Last, the RR will have an OF entry per active user and sensor (textiti.e., a UE that has recently consumed data traffic). In the considered scenario, the probability that a user is active P_{UA} equals 0.01 [59]. Then, the number of entries in the RR was estimated as the mean of a binomial distribution (*i.e.*, $\overline{L}_{RR} = N_{UE} \cdot P_{UA}$). The OF entries of the RR stores the same information as the DLT for the active UEs.

Today's OpenFlow switches can comfortably support the estimated requirements [61]. We can therefore conclude that the SDN technology is feasible and practical to implement the UP of the future 5G mobile networks.

2.4.2.2 Handover Execution Time

This subsection contains an evaluation of the HO execution time in order to verify if the softwarized core network architecture meets the latency constraints defined for the CP of the 5G mobile networks.

To carry it out, a simulator of a mobile network with the softwarized core architecture described in this chapter was built. Specifically, it simulates the processing and messages exchange of the different network entities instances for both the UP and CP. The links interconnecting two network entities instances were implemented as latency-rate servers [62]. The simulator was developed within the ns-3 simulation environment [63].

The proper operation of the HO procedure described in Section 2.3 was verified by using the simulator described above. Moreover, the HO execution time was measured. Figure 2.6 depicts the tree topology with two layers considered in the evaluation. Again the scenario considered is based on the dense urban information society *TC2* defined in METIS project [59]. The BN had a tree topology which is a common choice for BN deployments and offers a high level of scalability and reduced latency [64, 65]. For the sake of simplicity, the ESs are not shown in Fig. 2.6. The CP and UP share the same network infrastructure, and there is no differentiated treatment for the signaling messages. Table 2.2 contains the configuration of the main parameters of the simulator.

The UE measurement reports are triggered when the LTE event A3 occurs [66], *i.e.*, the Reference Signal Received Power (RSRP) of the TeNB becomes stronger than the RSRP of the SeNB by an offset. Finally, please note that Intra-SW HO and Inter-SW HO parameters included in Table 2.2 respectively stand for the probability of occurrence of an Intra-Switch Handover and an Inter-Switch Handover. The intra-SW HO happens when the SeNB and TeNB are connected to the same BN switch, otherwise the inter-SW HO takes place. These parameters are actually simulation outputs.

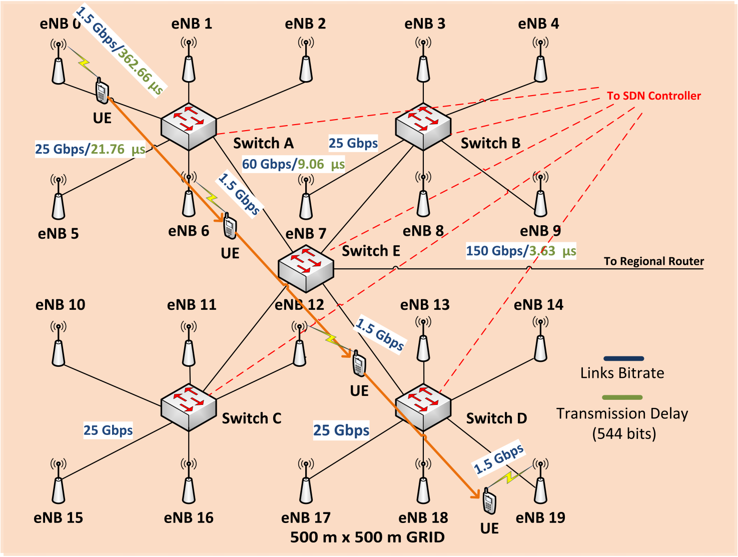


Figure 2.6: Scenario to evaluate the HO execution time.

Table 2.2: Parameters configuration to estimate the HO execution time.

Network Topology		**UE Mobility**	
eNBs layout	Regular grid 500 × 500 m^2 [59]	Mobility model	Fluid-flow model [67]
eNB coverage area	100 × 125 m^2	UE speed	6 m/s
Number of eNBs	20	**Traffic Model**	
eNBs positions $i \in \{0, ..., 19\}$	$[50 + (i\%5) \cdot 100, 62.5 + \lfloor i/5 \rfloor \cdot 125]$	Traffic model	ON-OFF
RR position	25 km away from SW E	ON and OFF periods	Uniformly distributed between (0, 1) seconds
Number of UEs	100	**Propagation delays**	
BN Topology	Tree (2 Levels) [68]	Speed of light (air)	300000 km/s
BN switches positions	A: (125, 125) m; B: (125, 375) m C: (375, 125) m; D: (375, 375) m E: (250, 250) m	Speed of light (wire)	200000 km/s
Processing delays		**Handover measurement reports**	
eNB	960 μs [69]	Event type	A3 with $offset = 2\ dB$ and $hysteresis = 0\ dB$
BN switches	5 μs [70] [71]	Carrier frequency	2.12 GHz
ENE switches	10 μs [72]	eNB Tx Power	30 dBm
SDNC	3 ms [73]	Path loss model	Friis
vMME	1 ms	**Handover cases**	
vS-GW	1 ms	Intra-SW HO	40/63
RR	100 μs	Inter-SW HO	23/63

The HO preparation and HO completion times were measured for different UE traffic rates at UP (see Fig. 2.7). The HO execution time is the sum of these contributions and the HO interruption time. It was assumed a constant HO interruption time of 15 ms [69]. The results show that HO preparation and completion times are almost constant for data traffic rates per UE up to 1 Gbps, with values 6.94 ms and 8.31 ms respectively. From this point on the HO execution time increases because the BN begins to exhibit congestion. Then, it is necessary to scale the link capacities of the BN to fulfill the CP latency target of the network.

According to [74], the CP delay budget for X2-based HO preparation and completion phases is 31 ms in LTE networks. Also, it is expected that latency requirements for the CP in 5G networks will be two times more rigid [49]. Consequently, the delay obtained for HO preparation and completion phases (15.25 ms) meets the CP latency requirements for 5G networks.

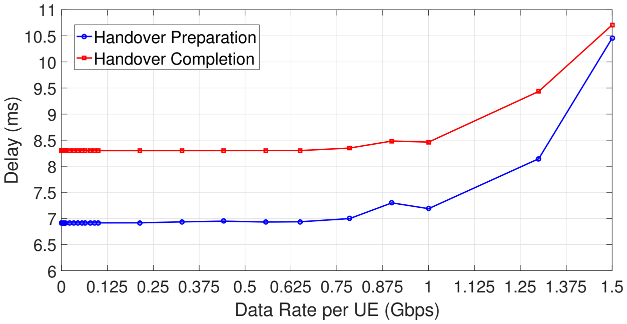


Figure 2.7: Handover Execution Time.

2.5 Conclusions

The requirements for 5G mobile networks have stringent requirements in terms of flexibility, scalability, cost effectiveness and energy efficiency. To meet these goals, SDN and NFV concepts have been considered as enabling technologies for the future 5G mobile networks. In this chapter, a partially virtualized architecture for the mobile core network has been proposed. The architecture integrates SDN and NFV concepts. Among its particularities are: i) the use of MPLS tunnels for the UP instead instead of GTP tunnels, which introduce significant overhead and their management might be cumbersome; ii) flexibility in the definition of the MPLS tunnels endpoints, thus offloading the core gateway (*e.g.*, RR) and enabling the deployment of local breakouts within the core network; and iii) the use of a three-tiered design CP entities that offers a higher flexibility, scalability , and robustness compared to a single tier design [46, 75, 76]. Additionally, the mobility support has been addressed for the proposed architecture. Specifically, an OpenFlow-based implementation of the X2-based HO procedure has been proposed and impacts of the mobility support on the architecture have been analyzed and listed.

Some initial proofs of concept has been also carried out. Firstly, a qualitative assessment of the architecture has been detailed. Secondly, the performance requirements of the SDN switches have been evaluated in foreseen 5G scenario. The results suggest that today's OpenFlow switches performance exceeds the requirements of the SDN switches of the softwarized core network. And lastly, the operation of the OpenFlow-based HO procedure has been validated and its execution time has been measured by means of simulations. The values obtained for the HO preparation time and the HO completion time have been respectively 6.94 ms and 8.31 ms when the network is in non-congested conditions. According to [74] and [49], these latencies meet the expected requirements concerning CP delay budgets for 5G networks.

Although further work is still needed, the proposed softwarized mobile core networks seem feasible for the future 5G mobile networks in light of the conclusions of this chapter.

Chapter 3

Modeling and Estimation of the Workload in Mobile Networks

Understanding the properties of the workload demands is essential for designing and optimizing future mobile networks (including Quality of Service (QoS) requeriments) with the goal of provisioning adequate communication services. This chapter deals with the modeling and estimation of the traffic demands for 5G mobile networks. We will focus on the traffic modeling of Mobile Broadband (MBB) and massive Machine Type Communication (mMTC) services. As it was explained in Chapter 1, enhanced Mobile Broadband (eMBB) and mMTC service groups are key drivers to develop future 5G systems. To the best of our knowledge, Ultra-Reliable and Low Latency Communications (URLLC) services are characterized by extremely stringent performance requirements, but the features of the traffic they generate are similar to the MBB traffic or mMTCs traffic depending on the specific service. However, the features and nature of the mMTCs traffic are different to MBB traffic as we will see in this chapter. This is the reason why the traffic generated by URLLCs is not explicitly addressed in this chapter.

The rest of the chapter is organized as follows. Section 3.1 reviews the main MBB services and the key parameters for their characterization. It also includes the traffic modeling of mMTC services. Not only data traffic will be relevant for future mobile networks, but also the signaling traffic. Particularly, it is expected to experiment an explosive growth due to the increasing adoption of mMTC services [77], and due to the background activities on smartphones as well [78]. In this regard, Section 3.2 provides a brief overview of the Long-Term Evolution (LTE) Control Plane (CP) and its main signaling procedures for session and bearers management, which provides insights into the nature of signaling traffic in mobile networks. Section 3.3 provides an abstract model of the CP and the User Plane (UP) workload generation process for MBB services. This model defines a general framework to capture the MBB user behavior and estimate its traffic and signaling demand. Next, Section 3.4 includes the derivation of analytical expressions to estimate the mean signaling rates for the different control procedures based on the UP traffic demands, user's behavior and mobility, and Radio Access Network (RAN) deployment. Section 3.5 defines compound traffic models for future mobile networks. These compounds traffic models aim to resemble the foreseen traffic demands for 5G mobile networks. Section 3.6 includes the validation of the mean signaling rates derived in this chapter and the stochastic characterization of the aggregated workload generation processes for both the CP and the UP. Finally, Section 3.7 draws the main conclusions.

3.1 Source Traffic Models

This section presents a selection of source traffic models developed in the literature for the most representative MBB and mMTC services. The source traffic modeling refers to model the traffic of each user or Machine-Type Communications (MTC) device individually. This approach is in general more accurate than its counterpart, aggregated traffic modeling where the accumulated traffic from all the users or MTC devices is modeled as a single stream. The main disadvantage of source traffic models is that they exhibit higher computational complexity than aggregated traffic models. For the interested reader we recommend the reference [79] which includes a deep review of traffic models for mobile networks.

3.1.1 Source Traffic Models for MBB Services

According to [80], Web browsing, video streaming, video calling, social networks, and mobile instant messaging services account for more than 70% of peak aggregate traffic in America mobile access networks. This section briefly reviews the literature related to the characterization and source traffic models for these services.

3.1.1.1 Web Browsing

The traditional source traffic model for web browsing is included in [81]. In [82], the authors carry out a characterization of this service f for mobile networks using real traces.

During a web browsing session, a user launches a browser, then issues a series of web page download requests and reads the downloads; and closes the browser finally. The time intervals where the user consumes data traffic are referred to as activity periods. The amount of data downloaded for an activity period (*i.e.*, web page size) of a web browsing session is determined by the main object size (*i.e.* the HyperText Markup Language (HTML) file), the number of embedded objects and their sizes. A user might download several web pages during a session. The download time of a web page is determined by its size, the End-to-End (E2E) data rate, and the parsing time. The parsing time is defined as the time interval the web browser takes to parse the embedded objects. The reading times are the inactivity periods (*i.e.*, no data traffic is generated) between two consecutive web pages downloads. During this period, the user typically is reading the first web page downloaded. The statistical characterization of a web browsing traffic source is included in Table 3.1.

3.1.1.2 Video Streaming

For this service, we will consider the YouTube which was the top one application in 2016 in North America mobile access networks, accounting for 20.87% of peak downstream traffic. The download profile for this application is described in [83, 84] for fixed networks and in [83, 85] for mobile networks. Youtube's video clips are transferred at a constant and limited rate during a *throttling phase* after an initial period of high downloading rate, called *initial burst.*

Table 3.1: Web browsing source traffic model characterization.

Parameter	Statistical Characterization
Main object size (bytes)	Truncated Lognormal Distribution [81]: $f_x = \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}}$, $x_{min} = 100Bytes$, $x_{max} = 2Mbytes, \mu = 8.37, \sigma = 1.37$ Generalized Pareto Distribution [82]: $f_x = \frac{1}{\sigma}\left(1 + \frac{\epsilon(x-\mu)}{\sigma}\right)^{\left(-\frac{1}{\epsilon}-1\right)}$, $\epsilon = 0.45$, $\sigma = 3000$, $\mu = 1$
Embedded object size (bytes)	Truncated Lognormal Distribution [81]: $f_x = \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}}$, $x_{min} = 50Bytes$, $x_{max} = 2Mbytes, \mu = 6.17, \sigma = 2.36$ Generalized Pareto Distribution [82]: $f_x = \frac{1}{\sigma}\left(1 + \frac{\epsilon(x-\mu)}{\sigma}\right)^{\left(-\frac{1}{\epsilon}-1\right)}$, $\epsilon = 0.85$, $\sigma = 1200$, $\mu = 0.1$
Number of embedded objects per page	Truncated Pareto Distribution [81]: $f_x = \frac{\alpha k^\alpha}{x^{alpha+1}}$, $k \leq x < m$; $f_x = \left(\frac{k}{m}\right)^\alpha$, $x = m$; $\alpha = 1.1$, $k = 2$, $m = 55$ Weibull Distribution [82]: $f_x = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$, $\lambda = 2.7392$, $k = 0.5409$
Parsing time (seconds)	Exponential Distribution [81]: $f_x = \lambda e^{-\lambda x}$, $\lambda = 7.69$
Reading time (seconds)	Exponential Distribution [81]: $f_x = \lambda e^{-\lambda x}$, $\lambda = 0.033$ Lognormal Distribution [82]: $f_x = \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}}$, $\mu = 8.37, \sigma = 1.37$
Session duration (seconds)	Weibull Distribution [82]: $f_x = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$ $x > 3.066s$, $k = 80000$, $\lambda = 0.4$

Table 3.2: Youtube source traffic model characterization.

Parameter	Statistical Characterization
Video Duration (seconds)	The CDF is a rational function [84, 85]: $F_x = \frac{5.813\cdot10^{-2}+2.747\cdot10^{-3}x+2.082\cdot10^{-5}x^2}{1+2.318\cdot10^{-3}x+2.088\cdot10^{-5}x^2}$, $x < 5000\,s$
Video Encoding Rate (bps)	91% of the videos have a 3GP format (itags 17 and 36) [85]: 90% of the videos with itag 17: encoding rate $\in$ $[60, 100]$ kbps 80% of the videos with itag 36: encoding rate $\in$ $[200, 275]$ kbps
Session duration (seconds)	The mean session duration is 1200 seconds [86]. Half of the sessions exceed 256 seconds [86]. Coefficient of variation of 2.1 [86].

The size of each video is calculated from its duration and encoding rate. The video encoding rate depends on the video format selected. Each video format, identified by an `itag` number, determines a container file format, an encoding algorithm, and a video resolution. The video download time (*i.e.*, activity period) is determined by the bottleneck link data rate during the initial burst and limited by the media server during the throttling phase [84, 85]. For mobile networks, roughly 34 s of the video clip are downloaded during the initial burst, and the rest of the video clip is transmitted by the media server at a rate of 2 times the video encoding rate in chunks of 64 KBytes during the throttling phase [85]. The user might watch several video clips during a session. The duration of a Youtube user session (the series of requests issued by a user to YouTube site in a single visit to the site) is characterized in [86]. The statistical characterization of a YouTube traffic source is included in Table 3.2.

3.1.1.3 Video Calling

In this application, a session starts when the user opens a video calling client app and makes a single call to someone else. This application generates roughly constant bit rate [87], which depends on the codec used. For instance, 1.5 Mbps is the recommended download/upload speed of Skype for HD video calling. The analysis of Skype traffic (*e.g.*, voice and video streams, user behavior, and signaling characterizations) is addressed in [87]. The call duration or call holding time determines the application activity period duration. The statistical characterization for a Voice over IP (VoIP) call duration is included in [88]. Specifically, the

Table 3.3: Social networking source traffic model characterization.

Parameter	Statistical Characterization
Session inter-arrival time (seconds)	Lognormal Distribution [89]: $f_x = \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}}$, $\mu = 2.245$, $\sigma = 1.133$
Number of requests in a session	Zipf Distribution [89]: $f_x = \beta x^{-\alpha}$, $\beta = 4.888$, $\alpha = 1.765$
Request inter-arrival time (seconds)	Lognormal Distribution [89]: $f_x = \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}}$, $\mu = 1.789$, $\sigma = 2.366$

authors in [88] show that the exponential model fails to capture the characteristics of the call holding times. Instead, they suggest the use of the Generalized Pareto Distribution for the modeling of this process. More precisely, the CDF of the call holding times is given by:

$$F_x = 1 - \left(1 - \frac{k(x-m)}{s}\right)^{\frac{1}{k}}, \, m = 0, \, k = -0.39, \, s = 69.33.$$

3.1.1.4 Social Networking

Through social networks, users connect with each other, share and find content, and disseminate information [89]. Typically, a user begins a session browsing scrapbook, friends' profile, photos, messages, etc. Some of these activities have higher traffic demands than others. During the session, the user might carry out several and different types of activities. Then, the service can be characterized by the amount of data downloaded for each type of activity, the inter-arrival times of activities, and the probability that a user switch from one type of activity to another. The authors in [89] provide a characterization for this service, see Table 3.3.

Table 3.4: Mobile instant messaging source traffic model characterization.

Parameter	Statistical Characterization
Message inter-arrival time	Lognormal Distribution [90]: $f_x = \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}}$, $\mu = 2.411, \sigma = 2.276$
Message length (Kbytes)	Power-law Distribution [90]: $f_x = \frac{\alpha-1}{x_{min}} \left(\frac{x}{x_{min}}\right)^{-\alpha}$, $x_{min} = 0.4823$, $\alpha = 2.2566$

3.1.1.5 Mobile Instant Messaging

This service have become increasingly popular around the world, and has generated significant traffic demands on cellular networks [90]. Examples of instant messaging applications include WhatsApp, Facebook Messenger, WeChat, and QQ Mobile. This service might be characterized by two parameters, the inter arrival times and the size of the messages. The authors in [90] have tackled its characterization. Specifically, the heavy-tailed user behaviors have been characterized by lognormal distribution for the message inter-arrival time and power-law distribution for the message length, respectively [90] (see Table 3.4).

3.1.2 Traffic Model for mMTC Services

MTCs refers to those services that do not necessarily require human interaction. This section addresses the source traffic modeling of mMTCs which is characterized by a huge number of connected low-cost devices equipped with long-life batteries typically transmitting infrequent, small, and non-delay-sensitive data. Examples of mMTC applications include healthcare, smart metering, and fleet tracking. Common types of mMTC devices are sensors, actuators, personal electronic devices, and smart home appliances.

MBB traffic and mMTC traffic have two major differences [91]:

- MBB traffic is heterogeneous, whereas MTC traffic is highly homogeneous (all machines running the same application behave similarly), and further,
- MBB communication is uncoordinated on small timescales, whereas MTC may be coordinated, that is, many machines react on global events in a syn-

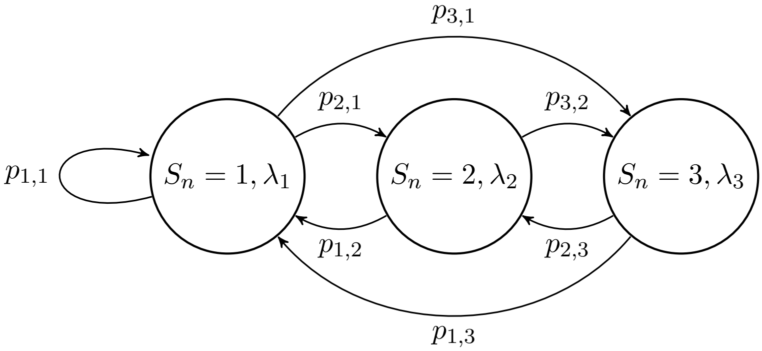


Figure 3.1: Example of MMPP with three states.

chronized fashion. Moreover, each MTC device usually transmits infrequent and small volumes of information.

It has been proven that Markov-modulated Poisson processs (MMPPs) are adequate to model the mMTC traffic patterns [92]. Further, the authors in [91,93] introduce the concept of coupled MMPPs to capture the above mentioned coordinated behavior of mMTCs. MMPP-based approaches for modeling MTCs offer good accuracy and flexibility, and linear computational time complexity [93].

Under an MMPP-based modeling approach, each mMTC device with index $j = \{1, 2, ..., N_D\}$ is modeled by an MMPP. Let n be the time index resulting of time discretization $n = \frac{t}{\Delta_T}$ for any constant time interval Δ_T. An MMPP is a Poisson process modulated by the rate $\lambda_j^{MTC}[n]$, which is given by the state of a Markov chain $s_j[n]$. Then, $\lambda_j^{MTC}[n] = \lambda_i$ when $s_j[n] = i$, where $i = \{1, 2, ..., I\}$ denotes the index of Markov state and λ_i denotes a constant rate associated with the state i. Figure 3.1 shows a Markov chain with three states driving a MMPP process.

3.2 Evolved Packet Core (EPC) Signaling

The CP consists of protocols to control the radio access bearers and the connection between the User Equipment (UE) and the network, *i.e.*, signaling between E-UTRAN and EPC [94].

Figure 3.2 shows the protocol stack for the CP between the UE and Mobility

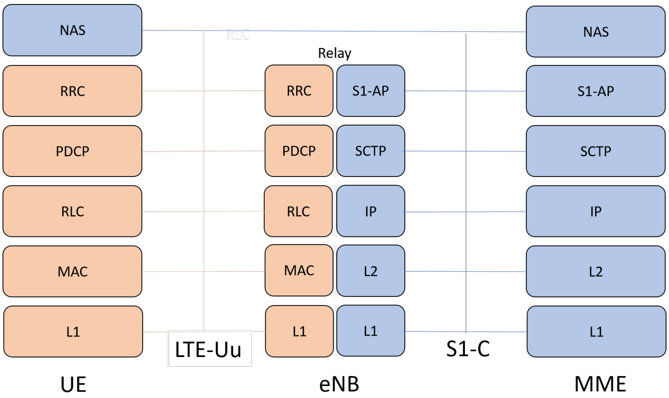


Figure 3.2: EPS control plane for E-UTRAN access [6].

Management Entity (MME). The orange-colored region of the stack refers to Access Stratum (AS) protocols. The Radio Resource Control (Radio Resource Control (RRC)) protocol is the main controlling function in the AS, being responsible for establishing radio bearers and configuring all the lower layers using RRC signaling between the evolved NodeB (eNB) and the UE [5].

The Non-Access Stratum (NAS) (*i.e.*, blue-colored region) forms the highest layer of the control plane between UE and MME at the radio interface [95] (see Fig. 3.2). The main functionalities of the NAS protocols are the EPS Mobility Management (EMM) to support mobility of the UE, and the EPS Session Management (ESM) to establish and maintain Internet Protocol (IP) connectivity between the UE and a P-GW.

In order to provide the aforementioned functionalities, the LTE standard defines several signaling procedures, each involving the exchange of control messages between several LTE logical entities. Here, we shall focus on those procedures most directly concerned with the session and bearers management, which are listed below [7]:

1. Attach. Once a given UE is subscribed to an LTE network, the UE initiates this procedure at power on or during the initial access of the network to get registered. After the successful completion of this procedure, the MME has a context for the UE, and a default bearer is established between the UE and the Packet Data Network Gateway (P-GW). Additionally, an IP

address is allocated to the UE.

2. Detach. This procedure allows a UE to break the attach procedure. Once this procedure ends, the user's Evolved Packet System (EPS) bearer(s) are released, and his state is cleared.

3. Tracking Area Update (TAU). This procedure updates the location of the UE within the network at Tracking Area (TA) (*i.e.*, a group of neighbor eNBs) level. A UE initiates a TAU when it detects that it enters into a new TA or periodically, even when the UE stays within the same TA.

4. Service Request (SR). This procedure is performed when an inactive UE in idle state wishes to get activated to handle traffic when there is new traffic [96]. In other words, this procedure is triggered when the UE is registered at the network, does not have available resources (S1 and RRC connections were released before), and new traffic is generated either from the UE or the network. If SR concludes successfully, the UE can receive or send traffic. Service request can be triggered by a UE or the network, depending on where the traffic is generated. For the network-triggered case, the network has to let the UE know that it has new traffic through the paging procedure.

5. Paging. Its purpose is to request the establishment of a NAS signaling connection to the UE. To that end, the MME sends a `Paging Request` message to all eNBs associated with the last known Tracking Area.

6. S1-Release (S1R). This procedure is typically triggered by eNB due to user inactivity, though it includes other cases and might be also triggered by MME. Its purpose is to release data radio bearers and downlink S1 bearer in the UP, S1-C and RRC connections in CP.

7. X2-based Handover (HO). It is carried out when the UE is in active state and moves from the coverage area of a Source eNB (SeNB) to a Target eNB (TeNB) in an intra-LTE environment. Additionally, both source and target eNBs are connected to the same MME/Serving Gateway (S-GW). Its purpose is to switch the data bearers' end points from the SeNB to the TeNB.

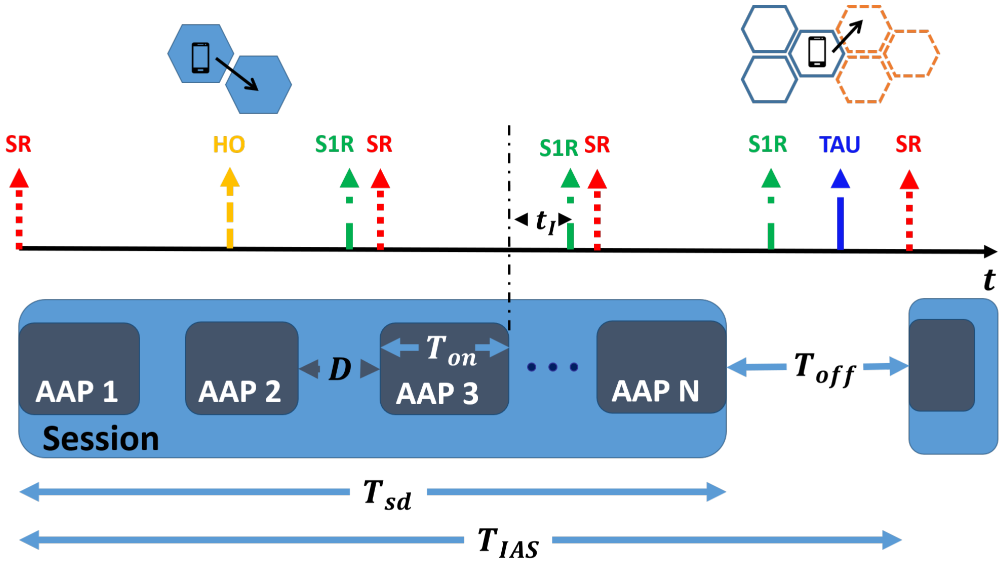


Figure 3.3: MBB workload generation model for a single user.

8. S1-based Handover. It is similar to X2-based handover in its purpose and triggering condition, but it applies when SeNB and TeNB are not connected with the same MME or there is not X2 interface available between them or when X2-based HO fails.

3.3 MBB Workload Generation Model

This section presents an abstract model for the signaling and UP traffic generation processes in LTE networks. Given that LTE networks are tailored for MBB services, this abstract model is also valid for the 5G eMBB use cases.

Figure 3.3 shows the abstract view of the workload generation process by a single user. While a given user is attached to the network, she runs network applications that generate or consume data traffic. The user chooses a certain application, hereafter referred to as *app* (*e.g.*, web browser, messaging app, social network app, and many others) of a set A ($app \in A$) with a given probability P_{app}. The duration of the connection of a given user to the network is divided into session intervals of duration T_{sd} and inter-session periods of length T_{off}. The inter arrival session time T_{IAS} is the time interval between the start of two consecutive sessions. Then, it holds that

$$T_{IAS} = T_{sd} + T_{off}.$$

Thus, the mean session arrival rate is defined as

$$\lambda_S = 1/E[T_{IAS}] = 1/(E[T_{sd}] + E[T_{off}]).$$

A session is defined as the user activity interval from the user launches a network application until she closes it. A session consists of N Application Activity Periods (AAPs) of length T_{on} separated by $(N-1)$ reading times of duration D.

An AAP is the time interval in which the application generates or consumes all necessary network traffic to perform a given task (*e.g.*, to download the profile of a user, to download a web page, to make a call, to send a message, or to stream a video, and so on). The reading time is the temporal interval during which the user performs any action that does not require to generate network traffic, such as reading a message, reading a web page, or deciding the next video to watch.

Assuming that N, D, and T_{on} are statically independents, then it holds that

$$E[T_{sd}] = E[N] \cdot E[T_{on}] + (E[N] - 1) \cdot E[D]$$

where $E[\cdot]$ denotes the expectation of a random variable.

Regarding the signaling workload, both the user activity and mobility trigger the LTE Control Plane (CP) procedures. Here, we shall only consider UE-triggered SR, S1-Release (S1R), X2-based Handover (HO), and Tracking Area Update (TAU) procedures. Although other procedures such as Attach and S1-based Handover are heavier in terms of computational resources consumption, they do not occur frequently in LTE networks [97].

Once the UE is registered in the network, an SR procedure is triggered during its idle-to-connected (*i.e.*, `IDLE` to `ACTIVE`) transitions. Then, whenever a AAP starts and the UE is in idle mode, an SR procedure takes place (see Fig. 3.3).

Conversely, an S1R procedure occurs during UE's connected-to-idle transitions during which the network releases the UE's resources. Here, we take into account the effects of the inactivity timer whose value is denoted as t_{IT}.

Typically, the inactivity timer is placed at each eNB to detect the users' inactivity, *i.e.*, the user does not perform any data communication over a period of length t_{IT}. In other words, the network waits t_{IT} units of time after an AAP finishes before triggering an S1R (see Fig. 3.3).

An HO procedure is triggered when a UE is in connected mode and performs a cell change, but the target cell is attached at the same MME as the source cell.

Finally, here we shall assume that a TAU procedure is triggered whenever a UE carries out a tracking area change, though LTE enables more sophisticated mechanisms to minimize the TAU rate [98]. These tracking areas are predefined and the same for any UE.

3.4 Signaling Rates Estimation

This section contains the derivation of analytical expressions to predict the mean arrival signaling rate at a mobile network from the workload generation model described in Section 3.3.

The mean arrival signaling rate $\lambda^{(CP)}$ is defined as the average number of signaling procedures triggered in the network per unit time.

Let λ_{cp} denote the average arrival rate of the control procedure cp in the set CP. As previously stated, here we shall only consider $cp \in CP = \{SR, S1R, HO, TAU\}$. Then,

$$\lambda^{(CP)} = \sum_{cp \in CP} \lambda_{cp} \tag{3.1}$$

We can also compute the mean arrival rate of control messages at the logical entity $e \in E = \{eNB, \text{MME}, SGW, PGW, HSS, PCRF\}$, or equivalently the number of signaling messages that arrive at the logical entity e per unit time, $\lambda_{CP}^{(e)}$, as

$$\lambda_{CP}^{(e)} = \sum_{cp \in CP} n_{cp}^{(e)} \cdot \lambda_{cp} \tag{3.2}$$

where $n_{cp}^{(e)}$ is the number of packets to be processed by the logical entity e for each type of control procedure cp.

In following subsection we provide the derivation of the average signaling rates generated per user λ_{cp}^{U}, or per MTC device λ_{cp}^{M}, for each type of control procedure

cp considered. Please note that $\lambda_{cp} = N_U \cdot \lambda_{cp}^U + N_M \cdot \lambda_{cp}^M$, where N_U and N_M respectively are the number of users and MTC devices attached to the network.

3.4.1 λ_{SR}^U and λ_{S1R}^U Estimation

SR procedure occurs whenever a user starts an AAP without having network resources assigned. When an AAP finishes, a user inactivity timer (whose value is denoted as t_{IT}) starts.

Let T_{IAAP} be the inter-AAP time, *i.e.*, the time elapsed between the end of an AAP and the beginning of the next one, regardless these activity periods belong to the same session or not. Thereby, $T_{IAAP} = D$ when the two AAPs are within the same session and $T_{IAAP} = T_{off}$, otherwise. If $T_{IAAP} \geq t_{IT}$, the S1R procedure takes place.

Note that in steady state conditions,

$$\lambda_{S1R}^U = \lambda_{SR}^U \tag{3.3}$$

because for every SR triggered by a given user, a corresponding S1R will occur before the next SR happens for that user.

The mean number of SR procedures per session is given by:

$$E[N_{SR}^S] = E[N] \cdot P(T_{IAAP} > t_{IT}) \tag{3.4}$$

where $P(T_{IAAP} > t_{IT})$ is the probability that the inactivity timer expires. Consequently, $\lambda_U^{SR} = \lambda_S \cdot E[N_S^{SR}]$.

Finally, note that for the first activity period $T_{IAAP} = T_{off}$ and for the following $(N-1)$ ones $T_{IAAP} = D$, then it holds that

$$\lambda_{SR}^U = \lambda_S \cdot ((E[N] - 1) \cdot P(D > t_{IT}) + P(T_{off} > t_{IT})) \tag{3.5}$$

3.4.1.1 λ_{HO}^U Estimation

Assuming that each eNodeB serves only one cell, the HO procedure takes place whenever a user performs a cell change while being in active state (*i.e.*, in `EMM-Registered` and `ECM/RRC-Connected` States). Here, we consider a user is in active state since she triggers a SR until her next S1R concludes.

Let P_{UA} denote the probability that a user is in active state at a given time, and let r_{cc} be the mean user cell crossing rate, *i.e.*, the average number of cell crossings per unit time. It holds that

$$\lambda_{HO}^{U} = r_{cc} \cdot P_{UA} \tag{3.6}$$

As an example, if we suppose that each user moves according to the fluid-flow mobility model, *i.e.*, at a constant speed with random direction uniformly distributed between $[0, 2\pi)$, it is known that

$$\overline{r}_{cc} = \frac{\overline{v} \cdot B}{\pi \cdot \Sigma} \tag{3.7}$$

where $\overline{v}$ is the average user speed, and B is the perimeter of the cell coverage area Σ.

To compute P_{UA}, let T_{ua} denote the amount of time that a user remains in active state between the end of a given AAP to the beginning of the next AAP. That is, $T_{ua} = T_{IAAP}$ if $X \leq t_{IT}$ and $T_{ua} = t_{IT}$ otherwise. Thus, T_{ua} will follow the same distribution as T_{IAAP}, but upper truncated to the value of t_{TI}, and its expected value can be computed as

$$\begin{aligned} E[T_{ua}] = \frac{1}{E[N]} \Bigg[(E[N]-1) \cdot \left(t_{IT} \cdot P(D > t_{IT}) + \int_0^{t_{IT}} x \cdot f_D(x)\, dx \right) \\ + t_{IT} \cdot P(T_{off} > t_{IT}) + \int_0^{t_{IT}} x \cdot f_{T_{off}}(x)\, dx \Bigg] \end{aligned} \tag{3.8}$$

Therefore, P_{UA} is λ_S times the amount of time that a user is active within a session:

$$P_{UA} = \lambda_S \cdot E[N] \cdot (E[T_{on}] + E[T_{ua}]) \tag{3.9}$$

3.4.1.2 λ_{SR}^{M} and λ_{S1R}^{M} Calculations

An SR procedure occurs whenever an MTC device is going to transmit a new packet without having network resources allocated. Let $P(t_r > T_I)$ denote the probability that the time interval between two packets transmission for any MTC device t_r be greater than inactivity timer value t_{IT}. It holds that

$$\lambda_{SR}^{M} = \lambda_{S1R}^{M} = P(t_r > t_{IT}) \cdot \sum_{i=1}^{I} \lambda_i \cdot \pi_i \tag{3.10}$$

where π_i is the probability of the state i and I the number of states of the Markov chain.

3.5 Compound Traffic Models

This section includes two compound traffic models based on the traffic models for MBB and mMTC services described in Section 3.1. These compound traffic models might be used to estimate the service consumption in mobile networks when there is no previous knowledge of the workload demand. They are also useful for academic proposes, *e.g.*, to generate synthetic workloads for experimentation (*e.g.*, to stress a virtualized LTE network).

The first compound traffic model is included in Table 3.5. This compound traffic model combines MBB and mMTC services. The MBB services considered are web browsing, video streaming, and video calling. The percentage of traffic generated for each type of MBB service, P_{app}, has been adjusted according to *TC2* scenario defined in the METIS project [59]. Similarly, the parameters of the distributions has been tuned to meet the future demands. For instance, the future sizes of the future web pages were estimated by extrapolating the data series of [99] and the distribution of the main object size was tuned accordingly. Similarly, we have considered the YouTube video formats with the highest encoding rates and resolutions to meet the data rates predicted by the Mobile and wireless communications Enablers for 2020 Information Society (METIS) project for this service. Regarding the mMTC services, a constant packet size of 100 bytes is considered. The setup for mMTC services was extracted from [91], which corresponds to a fleet management service case.

The second compound traffic model is included in Table 3.6. Please note that in Table 3.6 the parameters defined in Section 3.3 for the abstract model of the MBB workload generation process. This is done to show that any MBB service fits the pattern shown in Fig. 3.3. This compound traffic model includes the

Table 3.5: Compound traffic model for MBB and mMTC services.

Com. Type	Traffic Type	Parameters	Statistical Characterization
MBB ($E[T_{IAST}]$ = 1200 s [100])	Web browsing (HTTP) $P_{web} = 0.74$	Main Object Size	Truncated Lognormal Distribution: μ=15.098 σ=4.390E-5 min=100Bytes max=6MBytes
		Embedded Object Size	Truncated Lognormal Distribution: μ=6.17 σ=2.36 min=50Bytes max=2MBytes
		Number of Embedded Objects per Page	Truncated Pareto Distribution: mean=22 shape=1.1
		Parsing Time	Exponential Distribution: mean=0.13seconds
		Reading Time	Exponential Distribution: mean=30seconds
		Number of pageviews per session	Geometric Distribution: p=0.893 mean=9.312
	HTTP progressive video $P_{vs} = 0.03$	Video Encoding Rate	Uniform distribution with ranges: $(2.5, 3.0)$Mbps / (4.0,4.5)Mbps / $(12.5, 16.0)$Mbps / $(20.0, 25.0)$Mbps, for equiprobable itags: 137 / 264 / 266 / 315 respectively.
		Video Duration	Distribution extracted from [84]
		Reading Time	Exponential Distribution: mean=30seconds
		Number of video views per session	Geometric Distribution: p=0.6 mean=2.5
	Video calling $P_{vc} = 0.23$	Call Holding Time	Pareto Distribution: k=-0.39 s=69.33 m=0
		Number of calls per session	Constant = 1
mMTC	Infrequent small data transmissions (Packet Size = 100 B)	Discretization time interval	$\Delta_T = 1$ sec
		Markov chain state transition matrix	$P = \begin{pmatrix} 1-p & q \\ p & 1-q \end{pmatrix}$ where $p = 6.75 \times 10^{-5}$ and $q = 1.47 \times 10^{-4}$
		Markov chain state rates	$\lambda_1 = 0.0015$ packets/s; $\lambda_2 = 0.065$ packets/s

most representative MBB services in current cellular networks. It extends the compound traffic model of Table 3.5 to include mobile instant messaging and social networking services. Figure 3.4 depicts the Markov chain whose states are the user activities considered for social networking service.

The above-presented compound traffic models were used in many evaluations included in this thesis to emulate the workload generation process in mobile networks.

3.6 Aggregated Workload Characterization

3.6.1 LTE Workload Generator

The LTE workload generator is a simulation tool intended to generate synthetic UP and signaling traffic as in LTE networks. It was implemented within ns-3 [63] simulation environment. The data and signaling traffic is generated according to

Table 3.6: Compound traffic model for MBB services.

Traffic Type	Parameters	Statistical Characterization
Social Networking $P_{app} = 0.20$	Inter-arrival session times (T_{sst})	Log-normal distribution: μ=2.245 σ=1.333 (samples in seconds)
	Number of APPs per session (N)	From Markov chain
	Reading time (D)	Log-normal distribution: $\mu = 1.789$, $\sigma = 2.366$ (samples in seconds)
	AAPs length (T_{on})	Request data consumption (Markov Chain): • friend page = 1300 kB • Message page = 1 MB • Scrapbook page = 2 MB • Photo page = 750 kB
Video streaming $P_{app} = 0.20$	Inter-arrival session times (T_{sst})	Log-normal distribution: $\mu = 2.1$, $\sigma = 1.3$ (samples in seconds)
	Number of APPs per session (N)	1 (Constant)
	Reading Time (D)	Since $N = 1$, *no reading times*
	AAPs length (T_{on})	• Video length: power-law ($x_{min} = 32.8285$, $\alpha = 2.2619$) (samples in seconds) • Video resolutions: – 360p: 3 Mb/min – 480p: 5 Mb/min – 720p: 10 Mb/min – 1080p: 15 Mb/min • Download model according to [85]
Mobile Instant Messaging $P_{app} = 0.20$	Inter-arrival session times (T_{sst})	Log-normal distribution: $\mu = 2.411$, $\sigma = 2.276$ (samples in seconds)
	Number of APPs per session (N)	1 (constant)
	Reading Time (D)	Since $N = 1$, *no reading times*
	AAPs length (T_{on})	Message length (in KB): • Power-law distribution ($x_{min} = 0.4823$ KB, $\alpha = 2.2566$)
Web browsing $P_{app} = 0.20$	Inter-arrival session times (T_{sst})	Exponential distribution: $\lambda^{-1} = 1200$ seconds
	Number of AAPs per session (N)	Geometric distribution: $p = 0.893$
	Reading times (D)	Exponential distribution: $\lambda^{-1} = 30$ seconds
	AAPs length (T_{on})	• Main object size: – Truncated log-normal distribution: $\mu = 15.098$, $\sigma = 4.39 \cdot 10^{-5}$, $min = 100$ B, $max = 6$ MB (samples in bytes) • Embedded object size: – Truncated log-normal distribution: $\mu = 6.17$, $\sigma = 2.36 \cdot 10^{-5}$, $min = 50$ B, $max = 2$ MB (samples in bytes) • Number of embedded objects per webpage: – Truncated Pareto distribution: $mean = 22$, $shape = 1.1$ • Parsing time: – Exponential distribution: $\lambda^{-1} = 0.13$ seconds
Video calling $P_{app} = 0.20$	Inter-arrival session times (T_{sst})	Exponential distribution: $\lambda^{-1} = 1200$ seconds
	Number of APPs per session (N)	1 (constant)
	Reading Time (D)	Since $N = 1$, *no reading times*
	AAPs length (T_{on})	Pareto distribution: $k = -0.39$, $s = 69.33$, and $m = 0$ (samples in seconds)

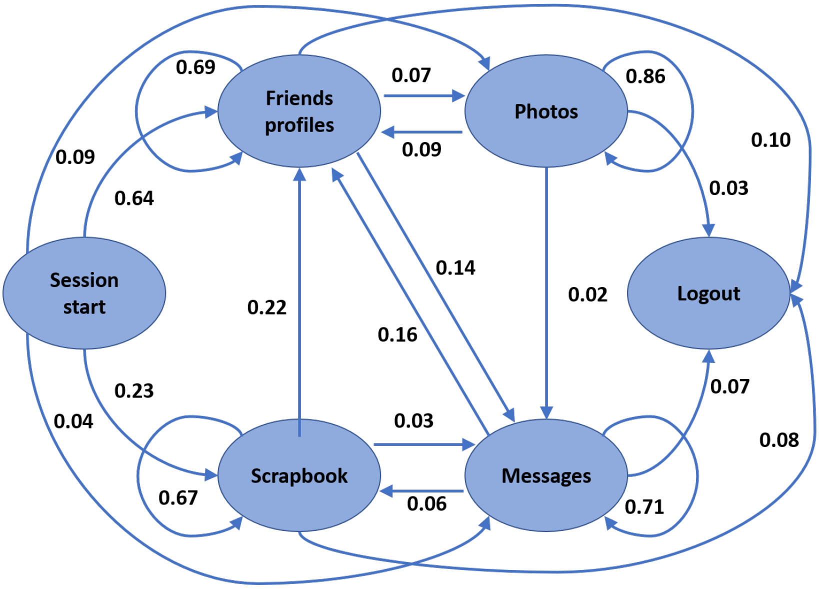


Figure 3.4: Markov chain based model for social networking.

the abstract model detailed in Section 3.3 by using the built-in pseudo-random number generator of ns-3. It allows to flexible define RAN scenarios and compound traffic models like those presented in Section 3.5. The tool supports several user mobility models like fluid-flow motion, random walk, and random waypoint.

A RAN deployment (eNBs positions within a rectangular geographical area) can be specified manually or created automatically from a population density map. To generate a RAN deployment automatically, the simulator implements a heuristic that tries to minimize the number of eNBs deployed, while guaranteeing a minimum Signal-to-Noise Ratio (SNR) and ensuring that the RAN capacity is enough at any location of its coverage area.

The output of this tool is a trace file that includes all the LTE signaling procedures triggered and all the data packets generated during the simulation.

3.6.2 Signaling Rates Validation

In order to validate the signaling rates expressions derived in Section 3.4, a set of simulations were carried out using the LTE workload generator described in the previous section. The considered scenario had 20000 UEs and 20000 sensors. The

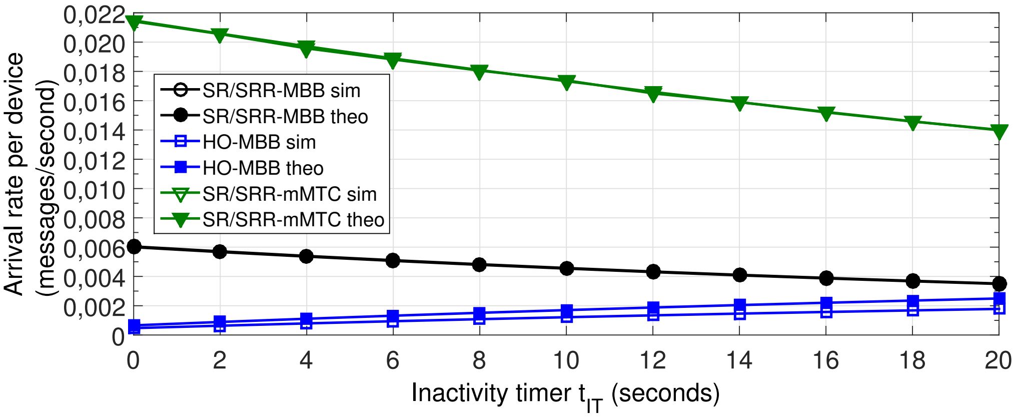


Figure 3.5: Control procedures arrival rates versus user inactivity timer.

RAN deployment used relies on the urban information society use case defined in the METIS project [59]. More precisely, the RAN consists of 12 eNBs distributed regularly in a 4 × 3 grid over a rectangular area of size 387 m × 552 m. The coverage area for each eNB is rectangular with dimensions of 138 m × 129 m. The users move around the area following a fluid-flow mobility model. The user speed is uniformly distributed between 0 and 4.2 m/s. All users have an independent and constant Uplink (UL) and Downlink (DL) data rate of 300 Mbps [59].

Figure 3.5 depicts the mean arrival rates for the different signaling procedures obtained by using the theoretical expressions (3.3)-(3.9) -referred to as label *theo*-, and after the conducted simulations -referred to as label *sim*- as a function of the inactivity timer t_{IT}.

The SRs and S1Rs rates decrease with t_{IT} for both MBB and MTC traffics. This is because the higher the value of the inactivity timer, the smaller the probability the timer runs out within an inter AAP of length D or T_{off}. Thus, the user stays in active state between consecutive AAPs, avoiding the need for triggering procedures to reserve and release resources. Conversely, the HOs rate increases with the timer value, as the user remains in active state longer after an AAP. Consequently, there is a higher chance that a user will be in active state, while she is moving from one cell to another.

Table 3.7 shows the root-mean-square error (RMSE) between the signaling

Table 3.7: RMSE for predicted arrival rate per device (see Fig. 3.5).

$RMSE(\lambda_{SR}^{U} = \lambda_{S1R}^{U})$	$RMSE(\lambda_{HO}^{U})$	$RMSE(\lambda_{SR}^{M} = \lambda_{SR}^{M})$
$4.07 \cdot 10^{-5}$	$15.0 \cdot 10^{-4}$	$6.65 \cdot 10^{-5}$

rates obtained experimentally (by means of simulations) and those estimated ones using expressions (3.3)-(3.10). It shows that the analytic expressions fit the experimental data obtained by simulation. The higher prediction error for the mean arrival HO procedure rate is due to the fluid-flow mobility model implementation of the simulator: a bounce-back strategy is employed when a user reaches an edge of the geographical area. That decreases the $\overline{r}_{cc}$ per user in comparison with that one predicted by the fluid-flow model expression (3.7).

It is worthy of note that the amount of signaling traffic generated by sensors is significantly higher than for MBB services. From Fig. 3.5, we can see that $\lambda_{SR}^{U} = 0.0045$, $\lambda_{HO}^{U} = 0.0012$ and $\lambda_{SR}^{M} = 0.0173$ procedures per second and terminal for $t_{IT} = 10s$, which is a typical value for the inactivity timer. That means each sensor generates about 3.5 times more control messages than a MBB user. This result suggests the definition of new, and more lightweight and energy-efficient signaling procedures for mMTCs [101–103].

3.6.3 Aggregated LTE Workload Processes Characterization

In order to study the aggregated workload generation processes for both CP and UP, we generated signaling and data traffic traces for 10^5 UEs and population densities ranging from 100 to 3000 inhabitants per km^2 using the LTE workload generator. The scenario considered for an overall population density of 500 inhabitants per km^2 is shown in Fig. 3.6. The scenario comprises three urban zones where most of the population is concentrated. The triangles represent eNBs locations and the circles edge clouds where the virtualized Evolved Packet Core (vEPC) is running on. We used the compound traffic model included in 3.6. The simulated measurement period was set to 10^4 seconds.

Following the analysis of the traces in [104], we depicted the rate process on 6 different time scales for both CP and DP traffics. The chosen time scales were

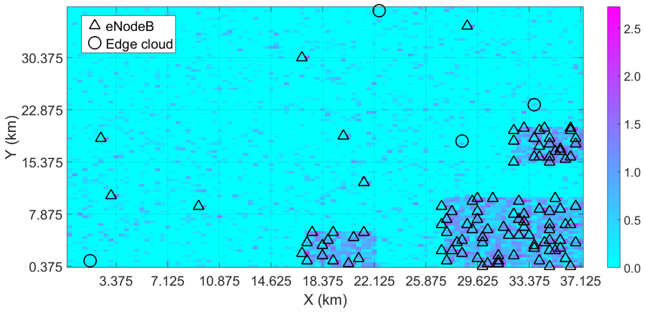


Figure 3.6: Scenario used to study the aggregated workload generation process for a population density of 500 users per km^2.

1 ms, 10 ms, 100 ms, 1 s, 10 s, and 100 s. More precisely, the analysis consists of depicting a sequence of simple plots of the packet counts (i.e., number of packets per time unit) for different choices of time units or time scales. From these representations we concluded that the DP traffic showed self-similarity (*i.e.*, it is statistically indistinguishable on different time scales). The same phenomenon was not observed for CP traces.

3.6.3.1 Aggregated UP Workload Arrival Process Characterization

The above-mentioned results motivates the use of self-similar stochastic processes for modeling the MBB UP traffic in a mobile network. Consequently, we adopted the arrival process model proposed in [105] to characterize the aggregated UP traffic arrival process.

Let A_t denote the cumulating UP traffic arrival process, i.e., the cumulative amount of traffic (in number of packets, say) arriving at the UP core gateway in the time interval $[0, t)$. The following model is considered for A_t [105]:

$$A_t = \lambda^{(DP)} \cdot t + \sqrt{\lambda \cdot \alpha^{(DP)}} \cdot Z_t, \qquad (3.11)$$

where Z_t is a normalized fractal Brownian motion (fBm) process with Hurst parameter $H^{(DP)} \in (1/2, 1)$, $\lambda > 0$ is the mean input rate, and $\alpha^{(DP)} > 0$ is a variance coefficient. An fBm process Z_t with Hurst parameter H is a Gaussian

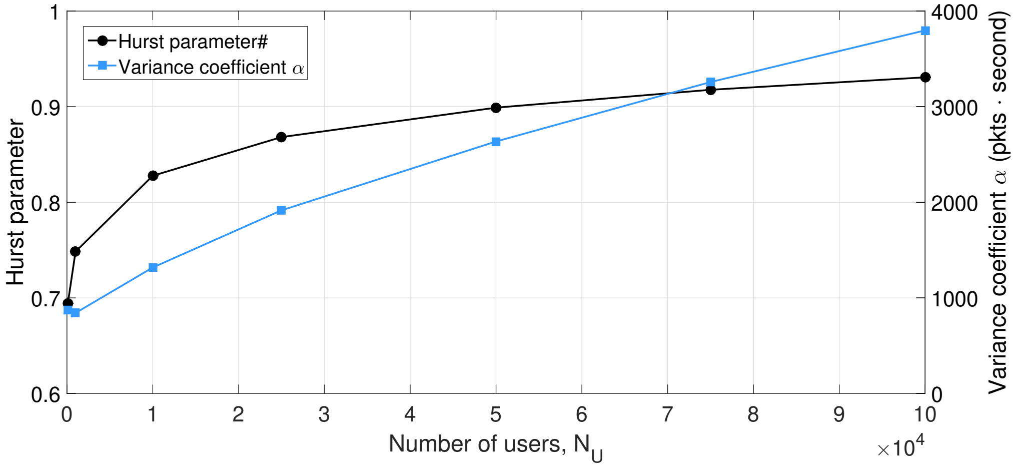


Figure 3.7: Variance coefficient α and Hurst parameter H measurements versus the number of users N_U for the aggregated DP arrival process.

process with mean equals zero and covariance function given by:

$$E[Z_t Z_s] = \frac{1}{2}\left(s^{2H} + t^{2H} - |t-s|^{2H}\right). \tag{3.12}$$

Then, the variance $VAR[Z_t]$ of an fBm process is given by:

$$VAR[Z_t] = E[Z_t^2] = t^{2H}. \tag{3.13}$$

From (3.13) it follows that the fBm process is self-similar (*i.e.*, for any constant $\beta > 0$, $\beta^{-H} Z_{\beta t}$ and Z_t have the same distribution for $t \geq 0$) as its covariance function is homogeneous of order $2H$ which is a statistical fractal property. The fBm process exhibits long-range dependence when $1/2 < H < 1$, *i.e.* the increments of the process ($Z_t - Z_s$ for $s \leq$) are positively correlated ($Corr(Z_{\beta t} - Z_{\beta s}, Z_{\beta v} - Z_{\beta u})$ does not converge to zero when $\beta \to +\infty$).

We measured $\lambda^{(DP)}$, $\alpha^{(DP)}$, and $H^{(DP)}$ parameters of the aforementioned traffic model for the aggregated UP workload generation process (see Fig. 3.7). To estimate the mean rate $\lambda^{(DP)}$, we simply counted the number of packets collected in the trace and divided it by the simulated measurement period. We obtained that each UE generates around 5.1121 packets per second in average.

This measurement was repeated considering different number of users N_U and we got the same result for the mean data rate per user. Consequently, $\lambda^{(DP)} = 5.1121 \cdot N_U$ for the compound traffic model included in Table 3.6.

To measure $\alpha^{(DP)}$ and $H^{(DP)}$, the same procedure as in [105] was followed. That is, we performed a linear regression from the logarithms of the sample variances of the increments of A_t for the 6 different time scales considered (*e.g.*, 1 ms, 10 ms, 100 ms, 1 s, 10 s, and 100 s). The variance of A_t can be computed as:

$$\begin{aligned} VAR[A_t] = E[(A_t - E[A_t])^2] = E[((A_t - \lambda^{(DP)}t)^2)] = \\ E[(\sqrt{\lambda^{(DP)} \cdot \alpha^{(DP)}} \cdot Z_t)^2] = \lambda^{(DP)} \cdot \alpha^{(DP)} \cdot E[Z_t^2] = \\ \lambda^{(DP)} \cdot \alpha^{(DP)} \cdot t^{2H^{(DP)}}. \end{aligned} \quad (3.14)$$

Then,

$$log\left(\alpha^{(DP)}\right) = -2 \cdot H^{(DP)} log(t-s) + log\left(\frac{VAR[A_{t-s}]}{\lambda^{(DP)}}\right). \quad (3.15)$$

Assuming that A_t defined in (6.26) has stationary increments [105] (*i.e.*, $A_t - A_s$ has the same distribution as A_{t-s} for $s \leq t$), $\alpha^{(DP)}$ and $H^{(DP)}$ can be estimated by means of linear regression. The measurement of $\alpha^{(DP)}$ and $H^{(DP)}$ was repeated for different number of users (see Fig. 3.7). Observe that the measured values for $H^{(DP)}$ versus N_U range from 0.7 to 0.93, which confirms the long-range dependence of the Data Plane (DP) traffic as $H^{(DP)} > 0.5$.

The $H^{(DP)}$ versus N_U curve was fitted by a logarithmic function within the range of N_U studied. Whereas $\alpha^{(DP)}$ versus N_U curve was fitted by a quadratic function. More precisely, for a given number of users $N_U \in (10^2, 10^5)$, we can estimate the main parameters of the fBm process that models the aggregated data traffic for the compound traffic model of Table 3.6 as follows:

$$\lambda^{(DP)} = 5.1121 \cdot N_U \, (packets/second) \quad (3.16)$$

$$\alpha^{(DP)} = -1.4 \cdot 10^{-7} N_U^2 + 0.043 N_U + 8.6 \cdot 10^2 \, (packets \cdot second) \quad (3.17)$$

$$H^{(DP)} = 0.035 \cdot ln(N_U) + 0.52 \quad (3.18)$$

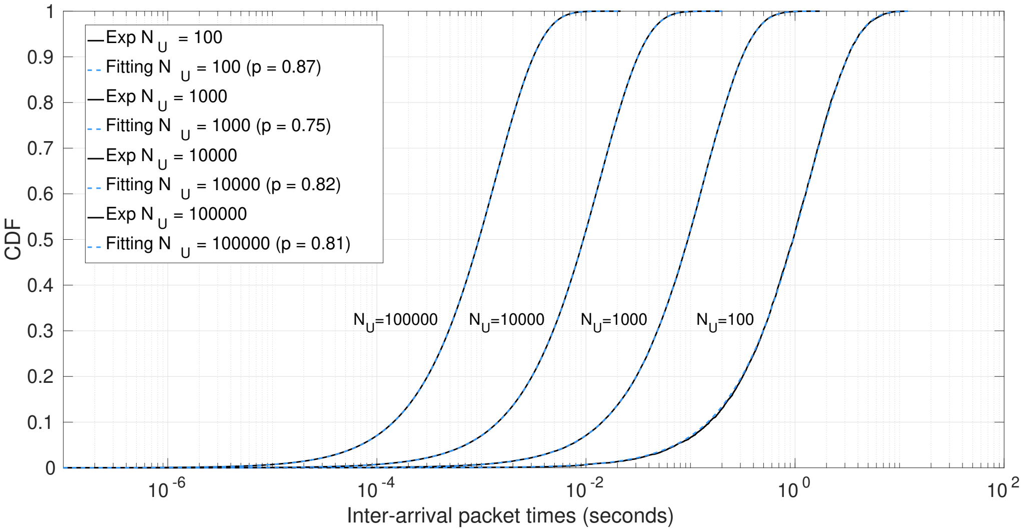


Figure 3.8: CDF of the signaling procedures inter-generation times in the mobile network.

where $ln(N_U)$ is the natural logarithm of N_U.

3.6.3.2 Aggregated Signaling Arrival Process Characterization

For the simulation scenario considered and the mean $\bar{t}_{PIG}^{(CP)}$ and standard deviation $\sigma_{PIG}^{(CP)}$ of the signaling procedures inter-generation times $t_{PIG}^{(CP)}$ were measured. The signaling procedures inter-generation time is defined as the the time period between the triggering of two consecutive signaling procedures (SR, S1R, HO, or TAU) in the mobile network. It was observed that $\bar{t}_{PIG}^{(CP)} \approx \sigma_{PIG}^{(CP)}$. This result suggests that the generation process of LTE signaling procedures can be considered as Poissonian distributed.

The empirical cumulative distribution of $t_{PIG}^{(CP)}$ was measured from the simulation results considering different number of users N_U and fitted by an exponential distribution (see Fig. 3.8). As it is shown in Fig. 3.8, the experimental curves (labeled as “Exp”) and the fitted curve (labeled as “Fitting”) are overlapped.

Additionally, a Kolmogorov-Smirnov test was performed to check whether the $t_{PIG}^{(CP)}$ samples comes from an exponential distribution. The test failed to reject the null hypothesis at the 1% significance level. The same experiment was conducted

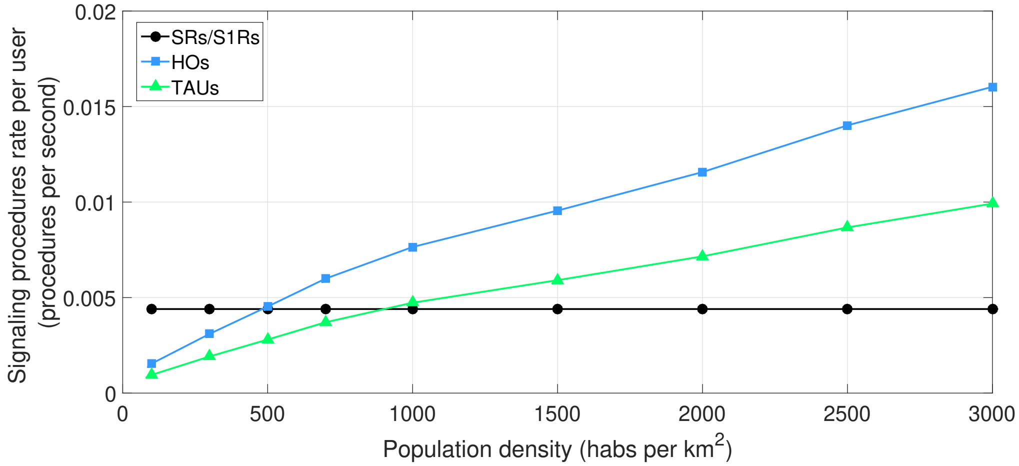


Figure 3.9: Generation rate per UE for the different LTE signaling procedures versus the population density for the considered scenario.

for different values of N_U and the same result was obtained. Specifically, we swept N_U from 100 to 100000. Consequently, the LTE CP workload generation process, under the assumption that it follows the abstraction described in Section 3.3, can be considered as Poissonian distributed. Then, as a result we conclude that the aggregated signaling arrival process at CP is fully characterized by the signaling generation rate.

Finally, following the same procedure to measure the aggregated UP traffic generation rate $\lambda^{(DP)}$, the signaling generation rates for each control procedure considered (*e.g.*, SR, S1R, HO, and TAU) were estimated for different population densities (see Fig. 3.9). It is observed that, unlike the SRs and S1Rs generation rates per user, the HOs and TAUs rates per user depend on the population densities. This fact is due to the increase in the RAN deployment density (*i.e.*, number of eNBs per km^2) when the population density increases.

Let λ_{SR}, λ_{S1R}, λ_{HO}, and λ_{TAU} be respectively the aggregated generation rate of the SR, S1R, HO, TAU procedures (*i.e.*,, the number of signaling procedures of a given type triggered in the network per unit time). From the curves shown in Fig. 3.9, it follows that the aggregated generation rates per procedure type considering the compound traffic model of Table 3.6 can be estimated as

$$\lambda_{SR} = \lambda_{S1R} = 0.0044 \cdot N_U \tag{3.19}$$

$$\lambda_{HO} = 4.2466 \cdot 10^{-6} \cdot N_U^2/(w \cdot h) + 0.003272 \cdot N_U \tag{3.20}$$

$$\lambda_{TAU} = 2.6281 \cdot 10^{-6} \cdot N_U^2/(w \cdot h) + 0.002025 \cdot N_U \tag{3.21}$$

where w and h respectively are the width and height of the coverage area of the whole RAN. And the aggregated signaling procedure generation rate $\lambda^{(CP)}$ can be computed as

$$\lambda^{(CP)} = \lambda_{SR} + \lambda_{S1R} + \lambda_{HO} + \lambda_{TAU} \tag{3.22}$$

3.7 Conclusions

Understanding the characteristics of the foreseen traffic demands for both the CP and the UP is of utmost importance to design and optimize future 5G mobile networks. The workload estimation is required as input for sizing (or dimensioning) the network. The real workload in a given scenario can be estimated using on-line measurements in operational networks. However, traffic modeling techniques are required during the network design and planning phases.

For this goal, in this chapter we have firstly reviewed the workload in current mobile networks. We also identified the main mMTC traffic features. The main conclusions drawn from this review are the following:

i) The most representative MBB services are web browsing, video streaming, video calling, social networking, and instant messaging in current mobile scenarios, accounting for for more than 70% of peak aggregate traffic in America mobile access networks [80]. Despite the differences among these services from the end user point of view, a common abstract model can capture the data and signaling traffic these services generate.

ii) Regarding the signaling workload, the most common procedures are those related with session and bearers management (*e.g.*, attach, detach, TAU, SR, S1R, X2-based HO, and S1-based HO). The SR, S1R, and X2-based HO are the signaling procedures that most frequently occur in LTE networks [97], whereas attach and S1-based HO procedures are the heaviest in terms of processing demands in the network entities [106].

iii) Unlike MBB traffic, MTC traffic is highly homogeneous and coordinated on small timescales. MMPPs are well suited for source traffic modeling of MTCs [91–93], offering accurate results and linear time complexity with the number of MTC devices.

As a contribution, we have provided an abstract model for both the data and signaling traffic generation for MBB services. Based on the abstract model, we have derived analytical expressions to estimate the signaling rates for different control procedures in MBB networks. The estimated signaling rates has been validated by means of simulation. These expressions shed light on the relationship between the signaling and data traffic workloads.

We have also defined compound traffic models that resembles the predicted demands for future mobile networks. The compound traffic model is built by considering traffic models derived in the literature from real traces, traffic forecast, and 5G use cases defined by different research projects [79].

Finally, we have characterized stochastically the aggregated generation workload processes for both the CP and the UP using the defined compound traffic model. The results show that the aggregated signaling generation is a Poisson process and the data traffic exhibits self-similarity and long range dependence features.

Chapter 4

Performance Evaluation of a Three-Tiered vMME

The Virtual Network Function (VNF) decomposition refers to the deployment of a VNF as separate components, so-called Virtual Network Function Components (VNFCs), each of which provides part of the VNF functionality. The linking of the VNFCs is specified in the VNF Descriptor (VNFD). VNF decomposition is of paramount importance for exploiting all the advantages Network Functions Virtualization (NFV) offers.

At the cost of increasing the complexity of NFV orchestration which are the subset of functions of the NFV ecosystem that are responsible for network services life cycle management, this approach might entail some advantages such as better utilization of the computational resources, and higher robustness of the VNF against system failures. It also eases the embedding of the VNF as each VNFC instance is more lightweight as it provides only part of the functionality of the VNF (Finding room for a 2 × Core flavor on a host is much easier than with one requiring 8 × Cores [107]).

VNFs with a three tiered or 1:3 mapping architecture [46, 108], adopted typically in Web services, is an example of VNF's decomposition. In this approach, each VNF is decomposed into multiple VNFCs of the following three types: Front-End (FE), stateless Worker (W), and State DataBase (SDB). This VNF decomposition achieves higher scalability and availability of the VNF, and it also reduces the complexity of VNF scaling [46, 108]. However, on the other hand it potentially increases the VNF response time, as every packet has to pass through several nodes. That is the main reason why this kind of VNF decomposition has been considered mainly to virtualize the Control Plane (CP) Evolved Packet Core (EPC) nodes [75, 76, 108–111] and the Internet Protocol (IP) Multimedia Subsystem entities (IMS) [112,113], where the delay constraints are less stringent than in User Plane (UP).

The softwarized 5G core network architecture proposed in Chapter 2 envisages the CP entities following a three-tier design. The present chapter studies the suitability of such architecture for the core network entities of the CP. It focuses specifically on the performance and scalability evaluation of a virtualized Mobility Management Entity (vMME) with a three-tiers design.

The rest of the chapter is organized as follows. Section 4.1 describes briefly the architecture and general operation of a three-tiered architecture. Section 4.2 highlights the processing tasks carried out by MME for the different signaling procedures. Section 4.3 includes the abstract model and main assumptions taken into account for the performance evaluation of the three-tiered vMME. Sections 4.4 and 4.5 respectively provide the analysis and assessment of the performance evaluation of the three-tiered vMME. Lastly, Section 4.6 presents the main conclusions of this chapter.

4.1 Architecture and Operation of Three-Tiered VNFs

As already mentioned, the functionality of a VNF with a three-tiered design is split into three different VNFCs: the FE, the W, and the shared SDB.

The FE is the communication interface with the outside world. It balances the load among the Ws. The worker implements the logic (*i.e.*, messages processing) of the VNF, and the DataBase (DB) contains all the state information making the Ws stateless.

Fig. 4.1 shows the architecture and general operation of a three-tiered VNF. Note that each VNFC might have several instances. The FE can only scale up/-down or vertically (*i.e.*, it can only increase or decrease the capacity of a VNFC instance by adding or releasing resources) without affecting the configuration of other VNFs. However, the W and the DB can also scale in/out or horizontally

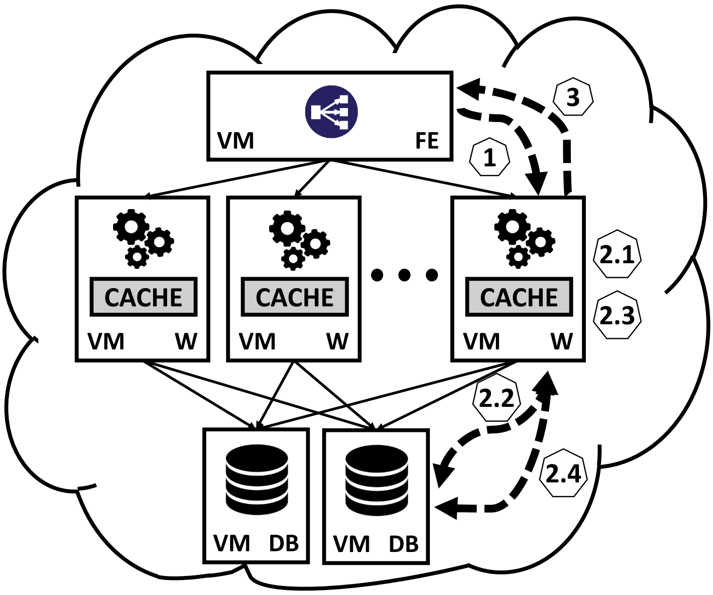


Figure 4.1: Architecture and operation of a three-tiered VNF.

(*i.e.*, they can increase or reduce the capacity of a VNFC by creating or removing instances or replicas of the VNFC).

As the FE provides the external interfaces of the VNF, all packets enter the VNF at the FE. Then, the FE sends every packet to the corresponding W according to its load-balancing scheme (in Fig. 4.1 labeled as "1").

Once the packet arrives at the W, the W parses the packet and checks whether the required data for processing it are stored in its cache memory (labeled as "2.1" in the same figure). This cache memory could be implemented inside the RAM allocated to the Virtual Machine (VM) where the W is running on.

If a cache mismatch occurs, then the W forwards a query to the DB to retrieve the data from it (labeled as "2.2" in Fig. 4.1). Note that this data retrieval pauses the packet processing at the W, during which the W might process other packets. When the DB gathers the necessary state variables, it sends them encapsulated in a packet back to the W.

The W can then finalize the packet processing (labeled as "2.3"). After processing finishes, it might be necessary to update some data in the DB (labeled as "2.4"). Then, the W generates a response packet and forwards it to the FE

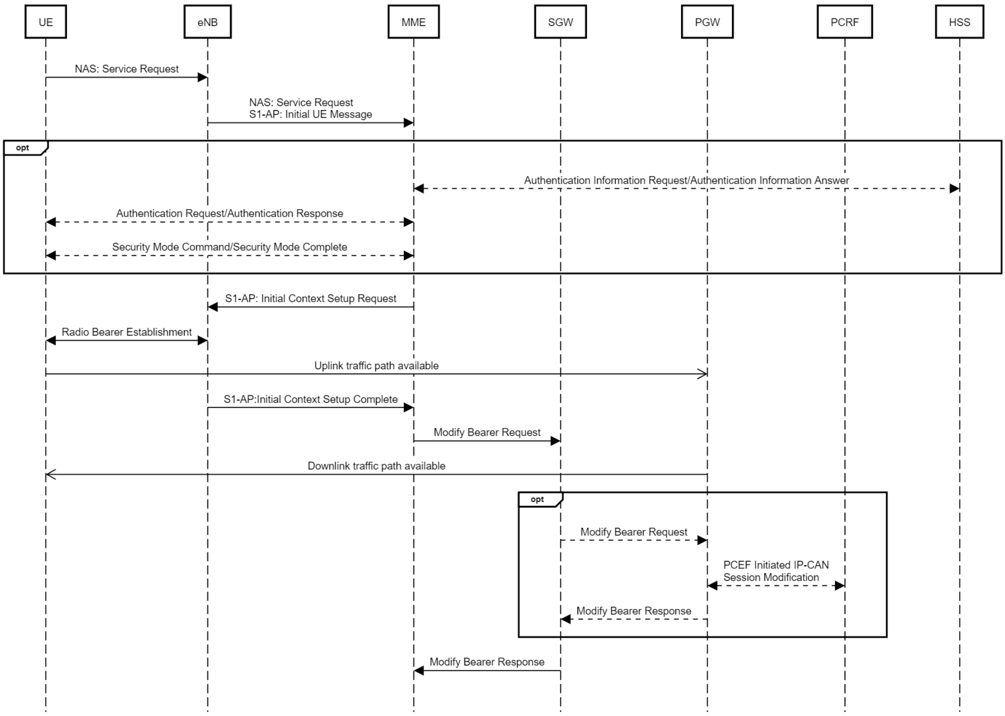

UE
eNB
MME
SGW
PGW
PCRF
HSS
NAS: Service Request
NAS: Service Request
S1-AP: Initial UE Message
opt
Authentication Information Request/Authentication Information Answer
Authentication Request/Authentication Response
Security Mode Command/Security Mode Complete
S1-AP: Initial Context Setup Request
Radio Bearer Establishment
Uplink traffic path available
S1-AP:Initial Context Setup Complete
Modify Bearer Request
Downlink traffic path available
opt
Modify Bearer Request
PCEF Initiated IP-CAN
Session Modification
Modify Bearer Response
Modify Bearer Response

Figure 4.2: UE triggered SR procedure [7].

(labeled as “3” in Fig. 4.1). Last, the packet exits the VNF.

4.2 MME Processing Tasks

As described in Section 3.2, the Long-Term Evolution (LTE) standard defines several signaling procedures that allow the control plane to manage the User Equipment (UE) mobility and the data flow between the UE and Packet Data Network Gateway (P-GW). From all of them, we only concentrate on the ones that generate most signaling load (*e.g.*, Service Request (SR), S1-Release (S1R), and X2-based Handover (HO)) [97]. The following subsections describe the processing carried out by the MME during the considered signaling procedures.

4.2.1 Service Request (SR)

When a UE does not have available resources and new traffic is generated, either from this UE or from the network to this UE, the UE performs a Service Request (SR) procedure. We shall focus on the UE-triggered SR because it occurs more frequently considering the Mobile Broadband (MBB) services scenario (typically the user starts the communication with a remote server). During this procedure the Mobility Management Entity (MME) receives three different messages (see Fig. 4.2): a `Service Request` (SR_1), an `Initial Context Setup Response` (SR_2), and a `Modify Bearer Response` (SR_3).

The Non-Access Stratum (NAS) `Service Request` message (SR_1) is encapsulated in the S1-AP `Intial UE Message`. Upon receiving the `Initial UE Message`, the MME allocates MME S1-AP UE ID, and establishes an S1 signaling connection between the evolved NodeB (eNB) and itself [96].

To process the SR_1 the MME first has to carry out UE integrity check and message decrypting. If the check passes, the MME does not need to authenticate the UE again. Otherwise, the MME performs authentication procedures for the UE (see Fig. 4.2). Then, it generates identifiers for the bearers to be established. Additionally, it stores and retrieves parameters and variables related to the UE context. Some of them are included in the subsequent `Initial Context Setup Request` message, which is sent to the eNB to request the establishment of a Data Radio Bearer (DRB) and a downlink S1 bearer.

During the processing of the `Initial Context Setup Response` message (SR_2), the MME also retrieves information of the UE context and includes this information in the subsequent `Modify Bearer Request` message. The processing of the `Modify Bearer Response` (SR_3) is lightweight as this message is only a confirmation from the Serving Gateway (S-GW) that the bearer modification was completed.

4.2.2 S1-Release (S1R)

The S1R procedure is triggered by user inactivity. Its purpose is to release data radio bearers and Downlink (DL) S1 bearer in the data plane, and radio and S1 signaling connections in the control plane for a UE. During the S1R, the MME processes three messages (see Fig. 4.3): the `UE Context Release`

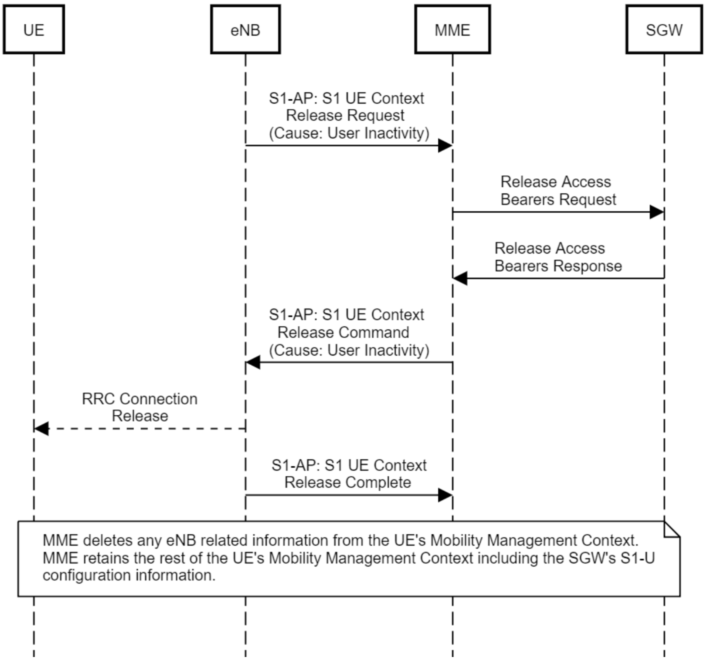

UE
eNB
MME
SGW
S1-AP: S1 UE Context Release Request (Cause: User Inactivity)
Release Access Bearers Request
Release Access Bearers Response
S1-AP: S1 UE Context Release Command (Cause: User Inactivity)
RRC Connection Release
S1-AP: S1 UE Context Release Complete
MME deletes any eNB related information from the UE's Mobility Management Context. MME retains the rest of the UE's Mobility Management Context including the SGW's S1-U configuration information.

Figure 4.3: S1R procedure [7], [8].

`Request` ($S1R_1$), the `Release Access Bearers Response` ($S1R_2$), and the `UE Context Release Complete` ($S1R_3$).

To process both the `UE Context Release Request` message ($S1R_1$) and the `Release Access Bearers Request` ($S1R_2$), the MME needs to retrieve information of the UE context and include this information in the subsequent messages. The processing of the `UE Context Release Complete` message ($S1R_3$) mainly implies the deletion of the bearer's context information by the MME (see Fig. 4.3).

4.2.3 X2-based Handover (HO)

The MME participates in the X2-based HO during the completion phase. Its purpose is to switch the bearers' end point from the source to the target eNB. The MME receives two messages during this phase: a `Path Switch Request` message (HO_1) and a `Modify Bearer Response` (HO_2).

The HO_1 message is sent by the Target eNB (TeNB) to notify that the UE

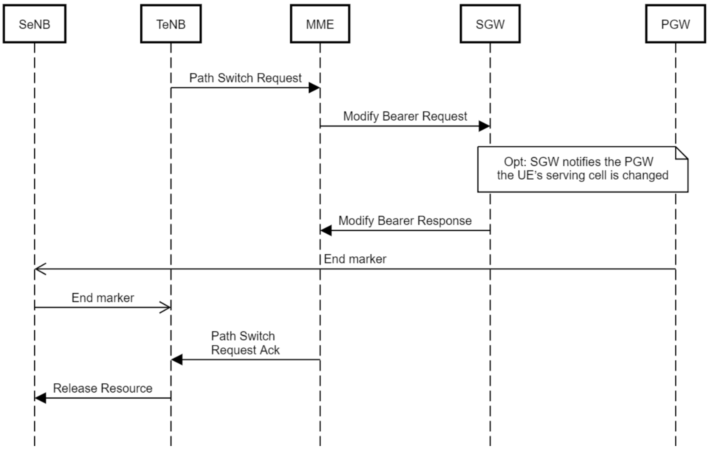


Figure 4.4: X2-based HO procedure - Completion Phase [7], [9].

serving cell is switched. After receiving this message, the MME informs the S-GW that the DL S1 bearer has been switched, and asks to switch the bearer path accordingly by sending a `Modify Bearer Request` message. The HR_2 is sent by the S-GW to confirm the bearer modification. Next, the MME notifies the target eNB that the new path has been established with a `Path Switch Request Ack` message.

To process both the `Path Switch Request` message (HO_1) and the `Modify Bearer Response` (HO_2), the MME also needs to retrieve information of the UE context and include this information in the subsequent messages. To process the `Path Switch Request` message, the MME also needs to store new information such as the IDs of the new serving cell and new tracking area.

4.3 System Model

Let us assume an LTE network with a three-tiered vMME, which runs in a cloud computing facility (see Fig. 4.5). The rest of the LTE network entities (*e.g.*, eNB, S-GW, P-GW, Home Subscriber Service (HSS), and Policy and Charging Rules Function (PCRF)) are deployed as Physical Network Functions (PNFs), *i.e.*, each of them is implemented via a tightly coupled software and hardware system [42].

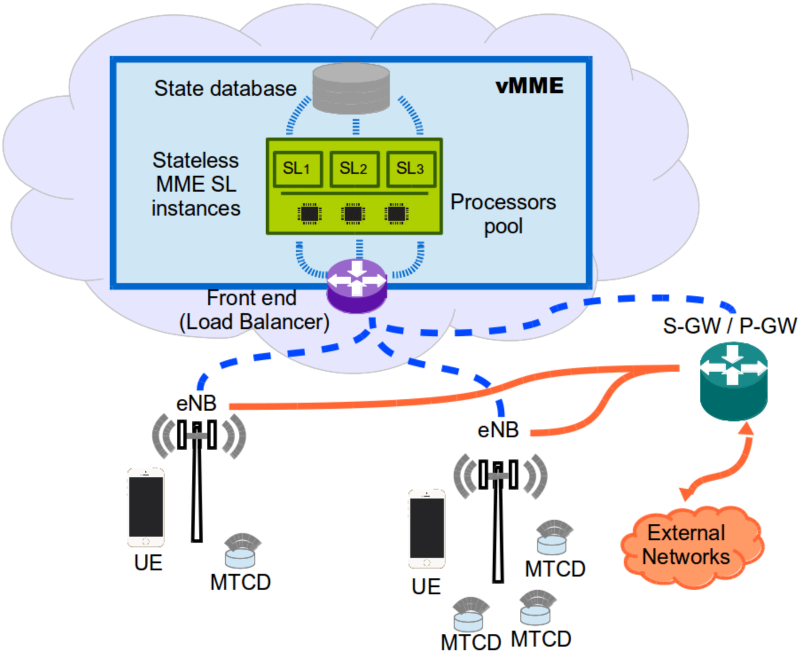


Figure 4.5: Overall system model.

The LTE network provides connectivity to N_U UEs and N_S sensors to external networks. The users move freely in the network coverage area and attach to different eNBs, whereas the sensors are placed at fixed locations. The UEs run the network applications, which generate or consume network traffic, as described in Section 3.3. The sensors send small data packets to centralized servers infrequently. The activity of the UEs and sensors, and the UEs' mobility trigger the signaling procedures. The sensors use the same signaling procedures as the UEs for the session and bearers management.

The eNBs provide radio connectivity to the UEs and sensors to the EPC. Each eNB contains inactivity timers with an expiration time of t_{IT}. The eNBs detect the UEs and sensors inactivity, *i.e.*, the UE or sensor does not perform any data communication over a period of length t_{IT}, and then release network resources.

As described in Section 2.1, the MME is the main control entity of the LTE networks. It is in charge of maintaining the mobility state of the UE, bearer management, and user authentication and authorization, among other functions. To support this functionality, LTE standard defines several signaling procedures (refer to Section 3.2), which imply an exchange of signaling messages between the MME and other LTE logical nodes (e.g., eNB, S-GW and HSS).

When the MME receives one control message, it processes the message (see Section 4.2), and then it might send a new message to another logical entity such as eNB, S-GW or HSS. Furthermore, the destination entity might send back a response message to the MME.

Let $\bar{t}_{IM}$ be the average time elapsed between the vMME sends a control message to another entity and the response message arrives at the vMME from that entity, where applicable. This time models the network delays and processing delay of the entity interacting with the MME.

The vMME is decomposed according to the three-tiered architecture presented in Section 4.1. Regarding the operation of the three-tiered vMME, we shall suppose there is no cache memory at the W instances to store the users and sensors contexts and the transactions states. Thus, the W instance first queries the corresponding user or sensor context and transaction state to the DB tier when a control message arrives.

In the same way, the W instance updates the user or sensor context and transaction state when it finishes processing a message. The user or sensor context consists of a set of information elements associated with the user or sensor that can be categorized into ID, location, security, and Evolved Packet System (EPS) Session/Bearer information [9]. For instance, when a W instance finishes processing the HO_2 message, it will update the eNB UE S1AP ID, E-UTRAN Cell Global Identifier, and S1 Tunnel Endpoint Identifier for DL in the user context stored in the DB tier.

The operation described above differs from a vMME implementation based on Elastic Core Architecture [75], and it allows fully stateless W instances. Consequently, different messages of the same signaling procedure and user can be processed by different W instances. Consequently, the number of W instances, denoted as m_W, can grow without affecting on in-session users. Finally, we shall assume that every W instance provides the same service process.

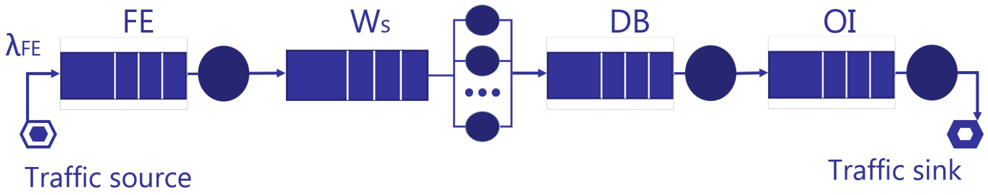


Figure 4.6: Feedforward open queuing network modeling a vMME with a three-tiers architecture.

4.4 Performance Analysis of Three-Tiered VNFs

4.4.1 Queuing Model

We will use a feedforward Jackson's network to model the vMME with a three-tiers design described in Section 4.3 (see Fig. 4.6). A Jackson's network is defined as an open network of M/M/m queuing nodes where external arrival processes are Poissonian and the transitions of the packets between nodes are probabilistic (probabilistic routing) [114].

The network of queues resembles a typical cloud processing chain, where the main bottlenecks considered are the processing capacities of the FE, the Ws, and the DB, and the bit rate of the Output Interface (OI). The OI refers to the Data Center (DC) interface to the external world which provides entry and exit point for all traffic from outside the DC towards the switching infrastructure for application and storage services [115].

The FE and the DB tiers, and the OI are modeled as M/M/1 queues whose service rates are denoted respectively as μ_{FE}, μ_{DB}, and μ_{OI}. The pool of Ws is modeled as a M/M/m queue, where each server attending the queue represents a W instance with service rate μ_W.

The aggregated signaling messages arrival process is the external arrival process, which is Poissonian (as shown in Chapter 3) with mean arrival rate λ_{MME}. All the control messages arrive at the FE tier and are served by the four queues in tandem. Last, all the control messages leave the network at the OI node.

4.4.2 Mean Response Time

For a Jackson's network with K queues, Jackson's Theorem states that when equilibrium exists ($\lambda_k < \mu_k \cdot m_k, \quad \forall \quad k \in [1, K]$, where m_k and μ_k are respectively the number of servers at queue k and the service rate offered by each server at queue k), the network has a product-form solution. That is, the probability of the overall system state $\boldsymbol{n} = (n_1, ..., n_K)$ will be given as

$$P(\boldsymbol{n}) = P(n_1, ..., n_K) = \prod_{k=1}^{K} \pi_k(n_k) \tag{4.1}$$

where $\pi_k(n_k)$ is the steady-state probability that there are n_k packets in the node k, which is found by considering the M/M/m queue at node k in isolation [114]. Note that a M/M/m queue is fully characterized by its total average arrival rate λ_k and its mean service time $s_k = 1/\mu_k$.

Then, the average number of packets in the system N can be computed as

$$N = \sum_{k=1}^{K} N_k = \sum_{k=1}^{K} \sum_{n=0}^{\infty} n \cdot \pi_k(n) \tag{4.2}$$

where N_k is the average number of packets in queue k.

The mean response time of the network T, *i.e.*, the mean time spent in the network by a packet, can be directly found from expression (4.2) by applying the Little's law. Then, it holds that

$$T = \frac{N}{\lambda} = \sum_{k=1}^{K} \sum_{n=0}^{\infty} \frac{n \cdot \pi_k(n)}{\lambda} = \sum_{k=1}^{K} V_k \frac{N_k}{\lambda_k} = \sum_{k=1}^{K} V_k T_k \tag{4.3}$$

where T_k is the mean response time of the queue k, λ is the external arrival rate entering the network, and V_k is the visit ratio. The visit ratio is the number of times that a packet visit a given queue during its lifetime in the network. The above equation shows that the mean response time of the network of queues is the sum of the total times that the packet will spend in each of the K queues. Below is the derivation of the mean response time of an M/M/m queue.

Each M/M/m queue of the network can be described through a birth-death process as shown in Fig. 4.7. A birth-death process is a special type of Markov

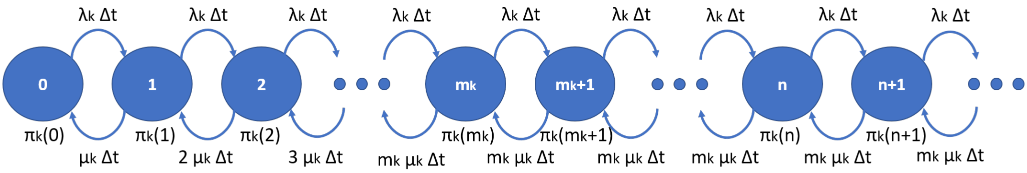


Figure 4.7: State transition diagram for the birth-death process of an M/M/m queue.

Chain where the state transitions at each step can only happen between adjacent states. The steady-state probabilities $\pi_k(n)$ of the M/M/m queue k are given by

$$\pi_k(n) = \begin{cases} \pi_k(0) \cdot \frac{\rho_k^n}{n!} & \text{if } n \leq m_k \\ \pi_k(0) \cdot \frac{\rho_k^n}{m_k!\, m_k^{n-m_k}} & \text{if } n > m_k \end{cases} \tag{4.4}$$

where $\pi_k(0)$ is the probability of the node k being empty and $\rho_k := \lambda_k/\mu_k$. Applying the normalization condition, *i.e.*, $\sum_{n=0}^{\infty} \pi_k(n) = 1$, we get

$$\pi_k(0) = \left(\sum_{n=0}^{m_k-1} \frac{\rho_k^n}{n!} + \frac{m_k \rho_k^n}{m_k!\,(m_k - \rho_k)} \right)^{-1} \tag{4.5}$$

An interesting performance measure for the M/M/m queue is the probability of queuing, or equivalently, the probability that a packet has to wait for service. This probability is given as

$$E_c(m_k, \rho_k) = \sum_{n=m_k}^{\infty} \pi_k(n) = \frac{\left(\frac{(m_k \cdot \rho_k)^{m_k}}{m_k!}\right) \cdot \left(\frac{1}{1-\rho_k}\right)}{\sum_{n=0}^{m_k-1} \frac{(m_k \cdot \rho_k)^n}{n!} + \left(\frac{(m_k \cdot \rho_k)^{m_k}}{m_k!}\right) \cdot \left(\frac{1}{1-\rho_k}\right)} \tag{4.6}$$

The above equation is referred to as Erlang-C formula, in honor of the mathematician Agner Krarup Erlang who first analyzed the M/M/m queue to model the call loss in a telephone exchange with m outgoing lines [116].

Finally, the mean response time of an M/M/m queue T_k is given as:

$$T_k = \sum_{n=0}^{\infty} \frac{n \cdot \pi_k(n)}{\lambda_k} = \sum_{n=m+1}^{\infty} \frac{n \cdot \pi_k(n)}{\lambda_k} + \frac{1}{\mu_k} = \frac{E_c(m_k, \rho_k)}{m_k \cdot \mu_k - \lambda_k} + \frac{1}{\mu_k} \tag{4.7}$$

The mean response time of the vMME T_{vMME}, according to the queuing model shown in Fig. 4.6, can be obtained using (4.3) under stability conditions ($\lambda_{MME} < \mu_{FE}$, $\lambda_{MME} < \mu_W \cdot m_W$, $\lambda_{MME} < \mu_{DB}$, $\lambda_{MME} < \mu_{OI}$) as

$$\overline{T}_{vMME} = \overline{T}_{FE} + \overline{T}_{SL} + \overline{T}_{DB} + \overline{T}_{OI} \tag{4.8}$$

Thus, by substitution of the corresponding queue parameters in (4.7) and considering that the FE, the DB, and the OI queues have only one server, the mean response time of each stage is given by

$$\overline{T}_{FE} = \frac{1}{\mu_{FE} - \lambda_{FE}} \tag{4.9}$$

$$\overline{T}_{W} = \frac{C(m_W, \rho_W)}{m_W \cdot \mu_W - \lambda_W} + \frac{1}{\mu_W} \tag{4.10}$$

$$\overline{T}_{DB} = \frac{1}{\mu_{DB} - \lambda_{DB}} \tag{4.11}$$

$$\overline{T}_{OI} = \frac{1}{\mu_{OI} - \lambda_{OI}} \tag{4.12}$$

Note that for the sake of clarity, we have used the labels FE, W, DB, and OI to differentiate the features and performance measures of each vMME stage instead of integer indexes (*i.e.*, $k \in \{1, 2, 3, 4\}$), but this does not affect the analysis. Observe also that $\lambda_{vMME} = \lambda_{FE} = \lambda_W = \lambda_{DB} = \lambda_{OI}$, since the queuing model of the vMME is the tandem of the four stages.

4.4.3 Worker Tier Dimensioning

The estimation of the the mean response time of the three-tiered vMME provided in the previous subsection can be used for the dimensioning of the worker tier. That is, to determine the minimum number of W instances m_W required for a given signaling arrival rate λ_{vMME} so that the vMME mean response time is kept under a threshold $\overline{T}_{max}$. Assuming that $\mu_{DB} > \lambda_{vMME}$, $\mu_{FE} > \lambda_{vMME}$, and $\mu_{OI} > \lambda_{vMME}$, the problem can be formulated as:

$$m_W = \min\{M : \overline{T}_{vMME}(\lambda_{vMME}, M) \leq \overline{T}_{max}, M \in \mathbb{N}\} \tag{4.13}$$

This problem could be solved with a brute-force algorithm that iterates until the vMME mean response time $\overline{T}_{vMME}$ is below the threshold $\overline{T}_{max}$. Starting at $m_W = \lceil \lambda_{vMME}/\mu_W \rceil$, m_W increases by one and $\overline{T}_{vMME}$ is recomputed using (4.8)-(4.12) at each iteration. The algorithm stops when $\overline{T}_{vMME} \leq \overline{T}_{max}$.

4.4.4 Scalability

In order to complete the study of the vMME as a distributed system, in this section we analyze its scalability. To that end, we adopt the scalability metric defined in [117]. According to this metric, a distributed system is scalable if the productivity is maintained as the system scale changes. The productivity is defined as the value delivered by the system per second over the cost incurred per second at scale factor z:

$$F(z) = \frac{\lambda(z) \cdot f(z)}{C(z)} \tag{4.14}$$

where $\lambda(z)$ denotes the average throughput in responses per second attained at scale z, $C(z)$ is the running costs of the system at scale z, and $f(z)$ is the value function that provides the value of each response, calculated from its Quality of Service (QoS) at scale z.

Here we will use the value function $f(k)$ defined in [117], which calculates the average response time $\overline{T}(z)$ compared to a target value $\widehat{T}$:

$$f(z) = \frac{1}{(1 + \frac{\overline{T}(z)}{\widehat{T}})} \tag{4.15}$$

The scalability metric is defined as the ratio of the productivity figures of the system at two different scales z_1 and z_2 [117]:

$$\psi(z_1, z_2) = F(z_2)/F(z_1) \tag{4.16}$$

If we substitute (4.15) for $f(z)$ in the above equation, we get

$$\psi(z_1, z_2) = \frac{\lambda(z_2) \cdot C(z_1) \cdot (\overline{T}(z_1) + \widehat{T})}{\lambda(z_1) \cdot C(z_2) \cdot (\overline{T}(z_2) + \widehat{T})} \tag{4.17}$$

Typically, the productivity of the system at a given fixed scale z_1, $F(z_1)$, is taken as a reference point and the scalability metric becomes a function of one single variable denoted by $\psi(z)$, whose argument might be understood as the configuration of the system for which we want to assess its scalability. In this case, $\psi(z)$ is interpreted as follows.

- If $\psi(z) = 1$, the system perfectly scales with z.
- If $\psi(z) > 1$, then the system scales positively with z.
- If $\psi(z) < \gamma$, the system does not scale.

Here, we set $\gamma = 0.8$ as in [117].

A strategy for scaling up/out (or down/in) the system is defined by the scaling factor z and several scaling variables (*e.g.*, number of processors, memory, disk, and network capacity allocated to the system) which depend on z. Here, we set as the scaling variable $m_W = z$. Therefore, the reference scale factor z_1 corresponds to the system with one W instance. Additionally, there are also adjustable variables (*e.g.*, allocation of processes to processors, the choice of communication protocols, and so on) known as scaling enablers. To maximize the productivity for any given z [117],the scaling enablers have to be tuned.

In our case, the scaling enablers are configured to achieve the maximum throughput, while the system response time is below a threshold.

To estimate the running costs of the cloud-based vMME realistically, we will employ the on-demand Amazon EC2 Service billing model [118].

Let $C_{ci}(m)$, $C_b(m)$, and $C_{db}(m)$ respectively denote the per instance computing cost, the load-balancer service cost, and the DB accessing cost. Then, the total cost $C(m)$ is

$$C(m_W) = C_b(m_W) + m_W \cdot C_{ci}(m_W) + C_{db}(m_W) \tag{4.18}$$

The cost of each element usually includes a rental fee, a storage charge, and a per transaction or throughput price. The computing cost per instance $C_{ci}(m_W)$ includes a per unit time billing costs depending on the type of processor $C_{ci_{type}}(m)$, the cost of the outgoing traffic sent to Internet $C_{ci_{thro}}(m)$ per unit

Table 4.1: Parameters configuration.

RAN topology [59]	
eNBs layout	Regular Grid 387 m x 552 m
eNB coverage area	138 m x 129 m
Number of eNBs	12
Inactivity timer value (t_{IT})	10 seconds
UE mobility	
Mobility model	Fluid-flow model
Speed	Uniform distribution (0, 4.2) m/s
EPC delays	
One-way delay (eNB → vMME)	7.5 ms
Two-ways delay (vMME ⇌ [eNB | S-GW])	15 ms
T_{max}	1 ms
Service rates	
FE service rate (μ_{FE})	120000 packets per second [119]
DB service rate (μ_{DB})	100000 transactions per second [120]
OI service rate (μ_{OI})	5000000 packets per second

time, and the per computing instance storage cost $C_{ci_{stor}}(m)$:

$$C_{ci}(m) = C_{ci_{type}}(m) + C_{ci_{stor}}(m) + C_{ci_{thro}}(m) \tag{4.19}$$

The database accessing cost $C_{db}(m)$ includes a rental fee per unit time $C_{db_{type}}(m)$, the cost per data capacity $C_{db_{stor}}(m)$, and a fee per transactions per unit time $C_{db_{trans}}(m)$.

$$C_{db}(m) = C_{db_{type}}(m) + C_{db_{stor}}(m) + C_{db_{thro}}(m) \tag{4.20}$$

The considered cloud service provides a load balancer service. Its cost $C_b(m)$ is charged by activation time $C_{b_{type}}(m)$ and served throughput $C_{b_{thro}}(m)$.

$$C_b(m) = C_{b_{type}}(m) + C_{b_{thro}}(m) \tag{4.21}$$

4.5 Performance Evaluation of a vMME with a Three-tiered Design

4.5.1 Experimental Setup

In order to validate the queuing model for estimating the mean response time of a three-tiered vMME, we used two software tools: the LTE workload generator described in Section 3.6.1 and a queuing simulator of a three-tiered vMME. The main configuration parameters are summarized in Table 4.1.

For the LTE workload generator, the considered simulation scenario relies on the dense urban information society use case defined in the Mobile and wireless communications Enablers for 2020 Information Society (METIS) project [59]. More precisely, the Evolved-Universal Terrestrial Radio Access Network (E-UTRAN) comprises 12 eNBs distributed regularly in a 4 × 3 grid over a rectangular area of size $387\,m \times 552\,m$. The coverage area for each eNB is rectangular with dimensions of $138\,m\,x\,129\,m$. The users move across the area following a fluid-flow mobility model. The user speed is uniformly distributed between 0 and $4.2\,m/s$. All users have an independent and constant Uplink (UL) and DL data rate of $300\,Mbps$ [59]. The most relevant information (e.g., timestamp, user/device id, and procedure type) of every LTE signaling procedure triggered during the simulation is recorded and dumped to a trace file.

The vMME simulator was developed using Matlab Simulink-Simevents [121]. It simulates the operation of the vMME tiers (*e.g.*, FE, W, and DB) and their messages exchange, as described in Section 4.3. The signaling traces generated by the LTE workload generator are used as an input of the simulator to emulate the signaling procedures arrival process at the vMME. The processing of each tier is simulated as a First-Come, First-Served (FCFS) queue attended by one or several servers. The servers have a deterministic service time. The load balancer has a service rate of 120000 packets per second [119]. The database service rate has been obtained by assuming the Amazon Aurora database [118] (deployed in the Amazon Cloud). It serves 100000 transactions per second [120]. The output interface is a 10G Ethernet that serves up to 5000000 packets per second (*i.e.*, assuming an average packet size of 250 Bytes for the control messages generated by the vMME).

To characterize the service times of the vMME W tier instances, the code of the main functions invoked by an MME to process the different control messages were implemented in C. Although that implementation were not fully compliant with the Third Generation Partnership Project (3GPP) LTE standards, it broadly executes the main tasks. The number of CPU instructions executed by an MME for processing every signaling message were measured from that code by using profiling tools, see Table 4.2. Then, the mean processing times were estimated for the EC2 m3.xlarge virtual instance of the Amazon EC2 service [118]. The average computing capacity of this type of instance is $11.38 \cdot 10^9$ float operations per second [122].

Table 4.2: Number of instructions executed by an MME and the corresponding processing times in *m3.xlarge* instance for the different signaling messages.

Type of message	Number of instructions	Processing Time (μs)
SR_1	1.45e+06	$s_{SR_1} = 127.4$
SR_2	1.07e+06	$s_{SR_2} = 94.0$
SR_3	1.06e+06	$s_{SR1_3} = 93.2$
$S1R_1$	1.07e+06	$s_{S1R_1} = 94.0$
$S1R_2$	1.07e+06	$s_{S1R_2} = 94.0$
$S1R_3$	1.06e+06	$s_{S1R_3} = 93.2$
HO_1	1.07e+06	$s_{HO_1} = 94.0$
HO_2	1.07e+06	$s_{HO_2} = 94.0$

The service times of the W instances for processing the different signaling messages were configured to the values included in Table 4.2. Notice that the mean service time of a W instance might be estimated as:

$$\begin{aligned}\widehat{s}_W = & \frac{\lambda_{SR}}{\lambda_{vMME}} \cdot (s_{SR_1} + s_{SR_2} + s_{SR_3}) \\ & + \frac{\lambda_{S1R}}{\lambda_{vMME}} \cdot (s_{S1R_1} + s_{S1R_2} + s_{S1R_3}) \\ & + \frac{\lambda_{HO}}{\lambda_{vMME}} \cdot (s_{HO_1} + s_{HO_2})\end{aligned} \tag{4.22}$$

where $\lambda_{SR}/\lambda_{vMME}$, $\lambda_{S1R}/\lambda_{vMME}$, and $\lambda_{HO}/\lambda_{vMME}$ are respectively the frequency of occurrence of the SR, S1R, and HO procedures.

The one-way delay between any eNB and the vMME, and the time between the vMME sends a control message to other entity and the response from that entity arrives at the vMME were implemented as constant delays in the simulator. The former was set to 7.5 ms, whereas the latter was fixed to 15 ms [123, 124].

4.5.2 vMME Capacity

Here, we evaluate the vMME capacity, defined as the maximum number of users and Machine-Type Communications (MTC) devices supported by the vMME for a given setup. In our study we report results obtained both after simulation and analytically as well. We will also leverage these results to validate the analytical

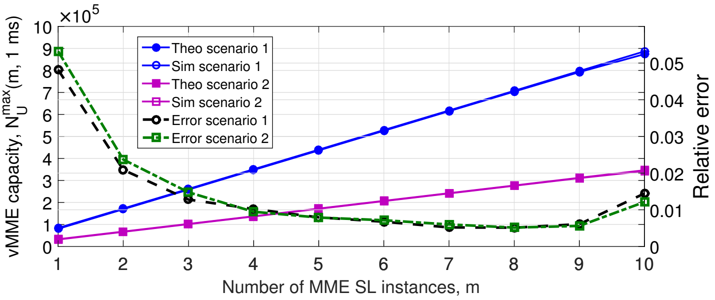


Figure 4.8: Capacity of the vMME *versus* the number of W instances.

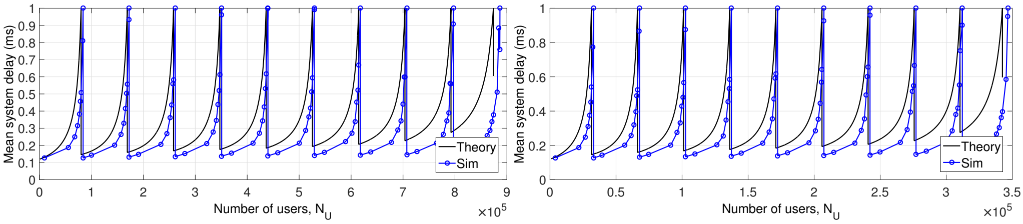
model presented in Section 4.4. For this purpose, we consider the following two scenarios:

- *Scenario 1*: with 1 MTC device per each UE,
- *Scenario 2*: with 3 MTC devices per each UE.

The 3GPP LTE standards define a delay budget to perform the signaling procedures [123, 125]. It implies that LTE CP entities have a delay budget for processing the different control messages. Here, we consider a processing delay budget of $\overline{T}_{max} = 1\ ms$ for the three-tiered vMME, which is half that for MME in a typical LTE network [123].

Figure 4.8 shows the vMME capacity $N_U^{max}(m_W, \overline{T}_{max})$ *versus* the number of W instances m_W for $\overline{T}_{max} = 1\ ms$. Notice that $N_U^{max}(m_W, \overline{T}_{max})$ denotes the maximum number of users supported by the vMME for a given scenario, m_W, and $\overline{T}_{max}$. A vMME capacity of $N_U^{max}(m_W, \overline{T}_{max})$ actually means that the vMME can withstand the signaling workload generated by $N_U^{max}(m_W, \overline{T}_{max})$ users and $N_U^{max}(m_W, \overline{T}_{max})$ MTC devices for *Scenario 1*, and $N_U^{max}(m_W, \overline{T}_{max})$ users and $3 \cdot N_U^{max}(m_W, \overline{T}_{max})$ MTC for *Scenario 2*. In Figure 4.8 the theoretical vMME capacity curve were obtained by using (4.8)-(4.13). The relative error between the theoretical and experimental curves roughly ranges from 0.5% to 5.5%. It demonstrates that the analytical model is useful for dimensioning purposes despite its implicit simplifications.

We also assess the vMME capacity for different target response times, $\overline{T}_{max}$, and user seeds. Results show that the vMME capacity does not differ significantly



(a) One sensor per each UE (Scenario 1). (b) Three sensors per each UE (Scenario 2).

Figure 4.9: Mean response time of the vMME versus the number of users.

in the range of the considered target response times, $\overline{T}_{max} \in [0.5, 3]$ ms. More precisely, from $\overline{T}_{max} = 0.5\,ms$ to $\overline{T}_{max} = 3\,ms$, the vMME capacity decreases by 0.94% (in *Scenario 1*) and 1.25% (in *Scenario 2*).

To study the influence of the user speed on the vMME capacity, we only consider the *Scenario 1* due to its greater suitability, as the MTC devices were supposed without mobility. In this case, results show that doubling the user speed drops the capacity of the vMME by 6.26%.

Figure 4.9 depicts the mean response time of the vMME as a function of the number of users for each scenario. As a general trend, given a number of W instances, the delay grows with the number of users. There is a point where the number of W instances cannot withstand the signaling arrival rate and the response time of the vMME shoots up. Whenever it holds that $\overline{T}_{vMME} = \overline{T}_{max}$, a new W instance is created to cope with the CP workload. This procedure explains the spiky patter shown in Fig. 4.9.

Notably, simulation and theoretical results show a similar shape, though the response times obtained by simulation are shorter. This is due to the assumptions underlying the theoretical model. For instance, the theoretical model considers exponential service times and a feedforward network. The root-mean-square error between simulation and analytical results is around 0.35 ms for both scenarios. The error increases with m_W because of the delay contribution of the DB instance in the analytic model has a significant impact earlier than in the simulation case.

Table 4.3: Cloud service configuration and cost calculation.

Cost	Configuration	Calculation
$C_{ci_{type}}(k)$	*m3.xlarge* instance rental (0.266\$/hour)	0.266/3600
$C_{ci_{stor}}(k)$	Local storage per month (10GB), and optimized data access (0.025\$/hour)	$10 \cdot 0.10 + 0.025/3600$
$C_{ci_{thro}}(k)$	The outgoing data rate from the DC	0.000(\$)/GB First GB/month 0.090(\$)/GB Up to 10 TB/month 0.085(\$)/GB Next 40 TB/month 0.070(\$)/GB Next 100 TB/month 0.050(\$)/GB Next 350 TB/month
$C_{db_{type}}(k)$	Aurora $db.r3.8xlarge$ instance (4.64\$/hour)	4.64/3600
$C_{db_{stor}}(k)$	0.1\$ per GB/month, for a total database size of $N_U \cdot 1KB$.	$(0.1 \cdot N_U \cdot 1024 \cdot \lambda_{vMME}/1e9)/2628000$
$C_{db_{thro}}(k)$	0.2\$ per million transactions/month	$0.2 \cdot \lambda_{vMME}/1e6$
$C_{b_{type}}(k)$	Service fee of 0.025\$/month	0.025/2628000
$C_{b_{thro}}(k)$	0.008\$ per GB serviced, supposing $O_{size} = 200$ Bytes	$\lambda_{vMME} \cdot 0.008 \cdot 200/1e9$

4.5.3 vMME Scalability

This section addresses the vMME scalability assessment. As we explained in Section 4.4, the scalability metric considered in this chapter depends on the running costs of the system. Here, we adopt the on-demand plan of the Amazon EC2 Service as billing model. Although there are alternative cheaper pricing plans (*e.g.*, reservation plan) [126, 127], the on-demand plan allows the customer to dynamically provision resources when required to cope with unexpected workload. Typically, the pricing in on-demand plans is charged by pay-per-use basis [126]. Table 4.3 details the pricing and setup considered.

Figure 4.10 shows the running costs of the system (measured in \$/s) as a function of the number of users. It includes three scenarios for different UEs to MTC devices ratios. The running cost of the vMME is piecewise linear function with number of users in the system. The overhead costs of deploying new W instances introduce discontinuities between consecutive segments of the function increasing the ordinate origin of the successive linear piece. Thus, the running cost exhibits a superlinear growth, hindering the system scalability.

We evaluate the scalability of the system by using (4.17), see Fig. 4.11. The vMME scales positively regarding the number of W instances for $m_W < 10$. However, beyond that point, the vMME is not perfectly scalable, *i.e.*, $\psi(k) < 1$. This is because of the DB tier utilization reaches about 100% of its capacity. At that point, it is required to scale the DB tier.

Nevertheless, recall from Fig. 4.8 that the vMME can serve roughly 900000

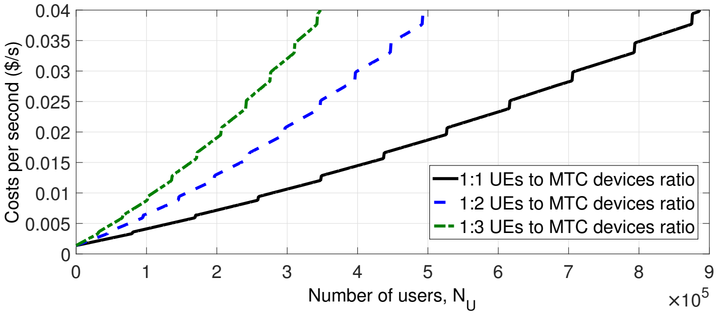


Figure 4.10: Cost per second versus number of users.

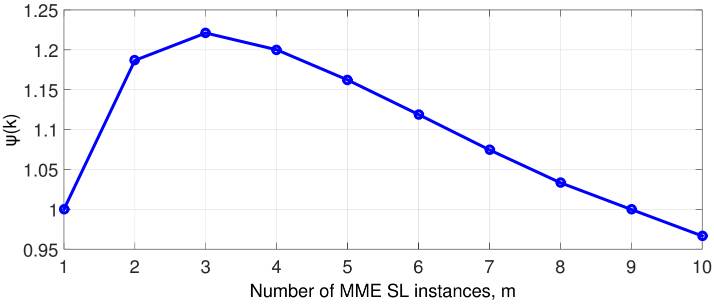


Figure 4.11: Scalability $\Psi(k)$ versus number of vMME SL instances.

UEs and 900000 MTC devices for *Scenario 1* and more than 325000 UEs and $3 \cdot 325000$ MTC devices for *Scenario 2*. This is the equivalent to approximately processing capacity 37000 LTE signaling procedures per second with a mean response time below $1\,ms$. The capacity of the vMME obtained is in the same order of magnitude as non-virtualized MME solutions [128].

4.6 Conclusions

The VNF decomposition refers to the deployment of a VNF as separate components, so-called VNFCs, each of which providing part of the VNF functionality. The linking of the VNFCs is specified in the VNFD. This approach can yield important benefits such as a better utilization of the computational resources, a higher robustness or to ease the resource allocation of the VNF.

A VNF might be decomposed into the following three VNFCs: FE, W, and DB. The FE acts as external interface of the VNF and balances the load among the W instances. The W implements the logic of the VNF carrying out the intensive processing of the incoming packets. And the DB stores the state information of the VNF making the W instances stateless. This decomposition resembles the typical architecture of the web services and brings some benefits such as a higher flexibility and availability of the VNF, and a reduction in the complexity of the VNF scaling. However, this architecture also increases the VNF response time, as every packet has to pass through several nodes.

In this chapter, we have analyzed and evaluated the capacity and scalability of a vMME with the aforementioned three-tiered design. According to the scalability metric considered [117], a system is scalable if the productivity keeps pace with costs. We define the productivity as the value delivered by the system per second. The productivity of the system is determined by assessing the performance of the scaled system and might depend on the QoS metrics (*e.g.*, throughput, jitter, delay, and/or packet loss probability) and/or other metrics such as the availability of the system. In our study we have considered the productivity as a function of the throughput and mean response time of the vMME. The capacity of the vMME is defined as the maximum number of users and MTC devices supported by the vMME for a given configuration, while the mean response time of the vMME is below a threshold ($1\,ms$ in the considered experimental setup).

To evaluate the mean response time of the vMME, we have modeled it as feedforward Jackson network. We validated this model by simulation. Despite the adopted simplifications, results have shown that it is useful to perform the dimensioning of the vMME. Specifically, we have obtained a relative error between the theoretical and simulation results for the vMME capacity below 5.5%.

The reported results, generated by considering a real cloud service configuration, suggest that the vMME is scalable for signaling workloads of up to 37000 procedures per second. This limit stems from the DB instance utilization reaches its maximum. To continue the scalability evaluation beyond that point, a strategy to scale the DB would have to be considered.

The results showed in this chapter also prove the feasibility of the virtualization of the CP logical entities and functionalities, and their decomposition following three-tiered architectures, which were inspired by web services, in cellular networks.

Chapter 5

Performance Modeling of Softwarized Networks

Computer networks have to fulfill a set of Quality of Service (QoS) or performance requirements which are imposed by the types of services they support. The QoS requisites are specified in the form of a Service Level Agreement (SLA) which is a document that describes the relationship between the provider and the customer [129]. Typical QoS metrics to measure the performance of a computer network include throughput, availability, packet loss probability, delay, and jitter.

Performance modeling is a cheap, agile and widely used technique for assessing the QoS of computer networks. It involves the abstraction of the features and properties of the computer networks, focusing exclusively on those that are of interest of study [10]. Its main objective is usually to obtain a set of performance metrics such as response times or packet loss probability in the steady-state of the system.

Figure 5.1 summarizes the performance modeling process. First, a system model is built based on the system specifications and/or behavior. The system model provides an abstraction of the system including those features that have the greatest impact on the system performance metrics of interest. Second, the analytical or simulation model is developed from the system model.

On the one hand, analytical models describe the system as a set of mathematical equations that relate performance metrics with the input parameters (*e.g.*, traffic demand, link capacities, processing times, time-to-failure, time-to-recovery,

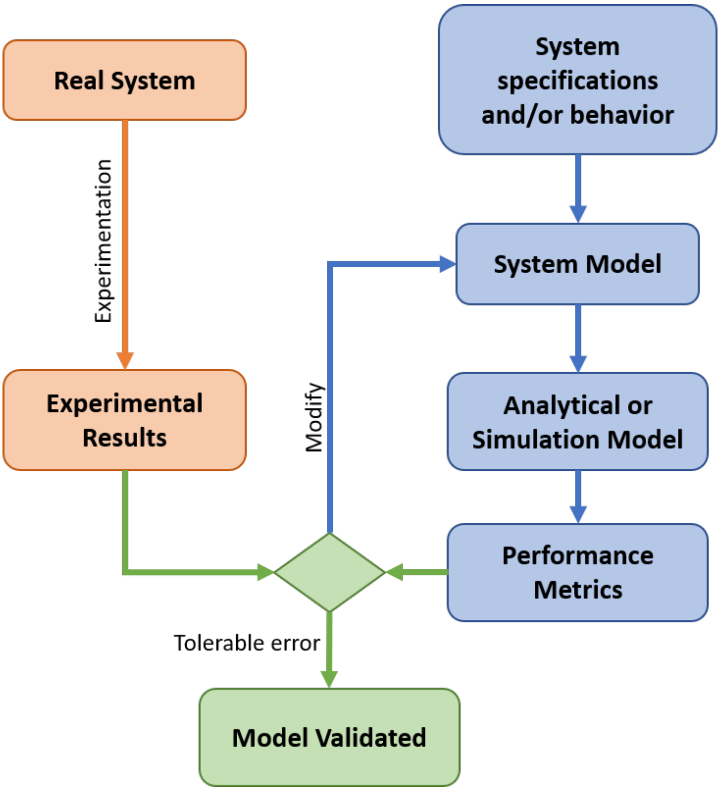


Figure 5.1: Performance modeling process [10].

etc). The most common theoretical framework for performance modeling of computer networks is Queuing Theory (QT) [130–133] which mainly provides mean performance metrics of the network. There are also newer approaches such as Network Calculus which is a theoretical framework that relies on alternative algebras (*e.g.*, min-plus and max-plus) and inequalities for analyzing performance guarantees in computer networks [134–136].

On the other hand, simulation models emulate the real system operation through use of computer software. Computer networks are most often simulated with discrete-event models, where the system state changes are caused by events, and these changes are considered instantaneous [10]. The simulation models offer higher level of flexibility and accuracy than analytical models at the cost of a longer development time and a greater computational complexity.

Third, the performance metrics predicted by the model are compared to experimental results (*i.e.*, the actual performance metrics offered by the real system). If the error is below a given threshold, taking into account the purpose or the application of the model, then it is validated. Otherwise, the system model has to be reviewed and refined and the subsequent process has to be repeated.

In the context of Softwarized Networks (SoftNets), performance modeling has the following two key applications, besides the traditional ones (*e.g.*, network design optimization, network tuning, bottlenecks identification, capacity planning, request policing, etc), which allow the automation of the deployment and scaling of network services:

- Dynamic Resource Provisioning (DRP), which enables a system to adapt its computational resources autonomously depending on the current workload so that some performance requirements are met. In [137] and [138], the authors show the application of QT-based models to that end. Additionally, we will also address this issue in Chapter 7.

- Network embedding (*i.e.*, how to map Virtual Network Function Component (VNFC) instances to physical infrastructures), during which the system must verify whether some given computational resources assignment will cater the particular SLA end-user demands. In [139], the authors provides an example of this application of performance modeling.

This chapter is intended to tackle the performance modeling of network services and chains of Virtual Network Functions (VNFs). It includes a performance model proposal based on QT networks for any composition of VNFs. The rest of the chapter is structured as follows. Section 5.1 provides some basic background about queuing networks. Section 5.2 briefly reviews the related literature. Section 5.3 describes the system model. Section 5.4 details the QT-based performance model for chains of VNFs. In Section 5.5 we particularize the performance model to a specific three-tiered virtualized Mobility Management Entity (vMME) use case. Section 5.6 explains experimental procedures, including the description of the experimental setup, the parameters estimation, and the conducted experiments. Section 5.7 provides simulation results to verify the correctness of the proposed model and to compare it with other baseline approaches in terms of

Table 5.1: Main notation for QNs.

Input Parameters	
λ_{0k}	Mean external arrival rate at node k
p_{ij}	Transition probability from node i to node j
μ_k	Mean service rate per server at node k
m_k	Number of servers of node k
V_k	Average number of visits a job makes to the node k during its lifetime in the QN
Output Performance Metrics	
λ_{QN}	Throughput of the QN
λ_k	Throughput of the node k
$\overline{N}$	Average number of jobs in the QN
$\overline{N}_k$	Average number of jobs in the node k
$\overline{T}$	Mean response time of the QN
$\overline{T}_k$	Mean response time of the node k.

computational time complexity and estimation error. Section 5.8 provides experimental results for model validation and includes a subsection for measured input parameters of the model. Finally, Section 5.9 summarizes the conclusions.

5.1 Fundamentals of Queuing Networks

Queuing Networks (QNs) are models that consists of multiple nodes, each with one or several servers. In these models, jobs arrive at any node of the QN to be served. Once a job is served at a node, it might either move to another node or leave the QN. The arrival and service processes at any node are typically described as stochastic processes.

QNs can be broadly categorized into open and closed networks. Open networks are those in which jobs arrive from outside to one or more nodes and after some time they leave the network. Conversely, closed networks have always a constant number of jobs circulating in it with no new job arrivals to the QN or job departures from the QN.

There are also multi-class QNs that have a set of jobs classes, where each class has its own service process at each node and specifications about the transitions between nodes. On the contrary, single-class QNs have only one class of job.

5.1.1 Open Jackson Networks

An open Jackson network is an arbitrary QN where its nodes, which are usually referred to as Jackson Servers, are M/M/m, its external arrival processes are Poissonian, and the routing of jobs in it is probabilistic.

Probabilistic routing means that a job served at node i is next moved to node j with probability p_{ij} or exits the network with probability $1 - \sum_{j=1}^{K} p_{ij}$. The routing decision for each job is random and independent of the routing decision for any other job at the node.

Under stability conditions, λ_k might be found by solving the following flow balance equations:

$$\lambda_k = \lambda_{0k} + \sum_{j=1}^{K} \lambda_j \cdot p_{jk} \tag{5.1}$$

For a Jackson network, *Jackson's Theorem* [140] states that when the stability condition is fulfilled at each queue ($\lambda_k < \mu_k \cdot m_k$), the joint probability of the QN states is given as

$$P(n_1, n_2, \ldots, n_K) = \prod_{k=1}^{K} p_k(n_k) \tag{5.2}$$

where $p_k(n_k)$ is the probability that there are n_k jobs in node k.

The most important implications of the Jackson's Theorem are the following [132]:

- The nodes of the QN can be considered in isolation even for QNs with feedback.
- The state of each node behaves as if it is independent of the states of the rest of nodes.
- The aggregated arrival process to any node behaves as if it is Poisson.

The mean performance metrics of the Jackson network can be computed as follows:

$$\lambda_{QN} = \sum_{k=1}^{K} \lambda_{0k} \tag{5.3}$$

$$\overline{T} = \sum_{k=1}^{K} V_k \cdot \overline{T}_k = \sum_{k=1}^{K} V_k \cdot \left(\frac{E_c(m_k, \rho_k)}{m_k \cdot \mu_k - \lambda_k} + \frac{1}{\mu_k} \right) \tag{5.4}$$

$$\overline{N} = \sum_{k=1}^{K} \overline{N}_k = \overline{T} \cdot \lambda_{QN} \tag{5.5}$$

5.1.2 Closed Gordon-Newell Networks

A Gordon-Newell network is a closed network whose nodes have First-Come, First-Served (FCFS) queuing discipline and exponentially distributed service times, and the routing of the jobs in the QN is probabilistic [141]. These QNs are also referred to as closed Jackson networks.

Note that in a closed QN, the N jobs continually circulate in the network moving from one node j to another node k with probability p_{jk}. Then, the routing matrix $P = [p_{jk}]$ is stochastic, *i.e.*, the row sums are all equal to one ($\sum_{k=1}^{K} p_{jk} = 1$) [133]. Moreover, here we will consider irreducible closed networks, *i.e.*, a job may reach one node to any other node in a finite number of steps with positive probability [133]. For these networks, P is irreducible (*i.e.*, the rank of $(I_{K\times K} - P)$ is $K-1$, where $I_{K\times K}$ is the identity matrix of order K).

Under equilibrium conditions, the flow balance equations of a closed network are given by

$$\lambda_k = \sum_{j=1}^{K} \lambda_j \cdot p_{jk} \tag{5.6}$$

Note that the above equations are not independent, since there are no external arrivals to the network. This system of linear equations is underdetermined with one degree of freedom. Then, we can find $\lambda_k \, \forall \, k \in [1, K]$ up to a multiplicative constant a, *i.e.*, $\lambda_k = a \cdot \lambda_k^*$, where $\lambda_k^* \, \forall \, k \in [1, K]$ is particular solution of (5.6).

For all states $n_1, ..., n_K$ such that $n_1 + ... + n_K = N$, the joint probability distribution of the network nodes states is given by the following product-form expression in a Gordon-Newell network.

$$P(n_1, n_2, \ldots, n_K) = \frac{1}{G(N)} \left[\prod_{k=1}^{K} \frac{u_k^{(n_k)}}{\zeta_k(n_k)} \right] \tag{5.7}$$

where the normalization constant $G(N)$, the relative utilization u_k, and the marginal probability of node k $P_k(n_k) = u_k^{(n_k)}/\zeta_k(n_k)$ are given by

$$G(N) = \sum_{n_1+...+n_K=M} \left(\prod_{k=1}^{K} \frac{u_k^{(n_k)}}{\zeta_k(n_k)} \right) \tag{5.8}$$

$$u_k = \frac{\lambda_k^*}{min(n_k \cdot \mu_k, m_k \cdot \mu_k)} \tag{5.9}$$

$$\zeta_k(n_k) = \begin{cases} n_k! & n_k \leq m_k \\ m_k! \cdot m_k^{(n_k-m_k)} & n_k > m_k \end{cases} \tag{5.10}$$

To find the solution of Gordon-Newell QNs, the primary problem is to compute the normalization constant $G(N)$. The greater the number of nodes in the QN or jobs, the more difficult the direct computation of this constant becomes. This problem can be circumvented in one of two ways. If we are only interested in computing the mean performance metrics of the QN, then we can compute them using the Mean Value Analysis (MVA) algorithm [142]. We explain this algorithm in the next section for multiple servers nodes and single traffic class. The other way is to use the convolution algorithm [132,133,143], which iteratively and efficiently computes the normalization constant $G(N-n)$ for $n = 0, ..., N-1$. The convolution algorithm is not covered in this queuing networks review as we will only focus on the mean performance metrics of the networks and MVA algorithm is enough for this purpose.

5.1.2.1 Mean Value Analysis

The MVA algorithm directly computes the mean performance measures of the closed network without computing the normalization constant or evaluating the state probabilities [132]. The algorithm is based on the following principle [142]: in a closed QN with a product-form solution, when a job arrives to a node k, it sees the same average number jobs in the node $\overline{N}_{k,N}^{(job)}$ as an outside observer will see if the QN had one less job $\overline{N}_{k,N-1}$. That is

$$\overline{N}_{k,N}^{(job)} = \overline{N}_{k,N-1} \tag{5.11}$$

Based on the above result, the MVA algorithm starts with zero jobs in the QN and it iterates until N jobs have been added to the QN. Only one job is added to the QN at each iteration, and the average performance metrics are computed from the results obtained in the previous iteration (for more details see Algorithm 1).

Algorithm 1 MVA Algorithm for Multi-Server Nodes and Single Traffic Class.

```
Input: $N$, and $\mu_k$ and $V_k \, \forall \, 1 \leq k \leq K$
Output: $\overline{T}_k$, $\overline{N}_k$, $\lambda_k$, and $\lambda_{QN}$
Initialization $p_k(0,0) = 0$, $p_k(j,0) = 0$, and $\overline{N}_{k,0} = 0$ for $k = 1, ..., K$, $j = 1, ..., m_k - 1$
for each $n \in [1, N] \cap \mathbb{N}$ do
    if Node $k$ is an infinite server then // Mean delay per node
        $\overline{T}_{k,n} = \frac{1}{\mu_k}$;
    else
        if Node $k$ is an FCFS, an LCFS, or a PS queue then
            $\overline{T}_{k,n} = \frac{\overline{N}_{k,n-1}+1}{\mu_k}$; // Mean delay per node
        else
            if Node $k$ is a multi-server FCFS then
                $Y_k = \sum_{j=1}^{m_k-1} (m_k - 1) \cdot p_k(j-1, n-1)$;
                $\overline{T}_{k,n} = \frac{\overline{N}_{k,n-1}+1+Y_k}{\mu_k}$;
            end if
        end if
    end if
end for
$\lambda_{QN} = \frac{n}{\sum_{k=1}^{K} \overline{T}_{k,n} \cdot V_k}$; // Throughput of the QN
for each $n \in [1, N] \cap \mathbb{N}$ do
    $\overline{N}_{k,n} = V_k \cdot \lambda_{QN} \cdot \overline{T}_{k,n}$; // Average number of jobs per node
    $p_k(j,n) = 1 - \sum_{i=1}^{K} p_k(i,n)$ for $j = 0$ and $p_k(j,n) = \frac{\lambda_{QN} \cdot p_k(j-1,n-1)}{\mu_k}$ for $j = 1, ..., N$;
end for
```

The MVA algorithm can be extended for Baskett-Chandy-Muntz-Palacios (BCMP) networks [144]. BCMP networks have product-form solutions and are an extension to a Jackson network. In a BCMP network, each node belongs to one of the following four types [145]:

- FCFS node whose services times follows a negative exponential distribution. The service rate of each node k, μ_k, may depend on the number of jobs in it.

- The infinite server, which is a service node with an infinite number of servers whose service time distributions have rational Laplace Transforms. Then a job immediately receives service upon entry to the node. Distributions with a rational Laplace Transform can be represented by a network of exponential stages [146].

- A single server node with Processor Sharing (PS) discipline, *i.e.*, when there are n jobs in the node k each is receiving service at a rate of $1/n \cdot \mu_k$. The service time distributions have rational Laplace transforms.

- A single server node with preemptive-resume Last-Come-First-Served (LCFS) discipline and service time distributions with rational Laplace transforms.

The MVA algorithm included in this section (Algorithm 1) is valid for single class QNs where each individual node may be multi-server FCFS, LCFS, PS, or infinite servers and has service time that is exponentially distributed.

5.1.2.2 Modeling of User Sessions in Closed Queuing Networks

Let us assume a population of N active users issuing requests to a system. Each request pass through several nodes before the system sends back a response to the corresponding user. Each request utilizes one single resource of the system a at time.

The above-described situation can be modeled as a closed queuing network as exemplified in Fig. 5.2. Specifically, an infinite server of mean service time $\mu = 1/Z$ is typically used to capture the behavior of the N active users issuing requests to the system, where Z is commonly referred to as user think time. The user think time is the mean time elapsed between two consecutive requests issued by a single user. This approach will be used in this chapter to compare the proposed performance model, which considers an open queuing network to model a composition of interconnected VNFs, with those ones proposed in the literature and rely on closed queuing networks.

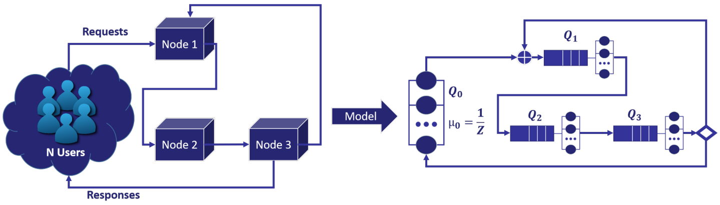


Figure 5.2: Modeling of user sessions in a closed network. For simplicity, only one queue is used to capture the behavior of each node of the system, though more complex models could be considered for the system.

5.2 Related Works

This section briefly reviews models proposed in the literature to assess the performance of softwarized networks. This review also includes some models for multi-tier Internet applications. There are several analytic models proposals tailored for multi-tier Internet services. However, in general terms, they do not take into account the particularities of the chains of VNFs.

In [147] Urgaonkar *et.al.* propose and validate experimentally a closed queuing network tailored to model these services. This seminal work has served as the basis for many other works. To compute the mean response time of a multi-tier application, they use the iterative algorithm MVA. As shown in [111], MVA execution takes a long time to solve the queuing network for scenarios characterized by a large number of active user sessions (*e.g.*, cellular network scenarios). In addition, the model assumes that a packet flow utilizes the resource of only one tier instance at a given time [147]. Then, for instance, it cannot be applied to model chains of VNFs processing video flows.

In [148] Bi *et.al.* address the dynamic resource provisioning problem for multi-tier applications. The authors assume an M/M/m queue to model the first tier and M/M/1 queues for the rest of the tiers. Since it is based on Jackson's network, this model may offer limited accuracy in some scenarios [111, 149].

Some works have tackled the modeling of a single VNFC instance. In [150], Gebert *et.al.* present an analytical model of a VNFC instance running on commercial-off-the-shelf (COTS) hardware. The authors consider the interrupt moderation techniques in their model. Their model is based on a generalization of the clocked approach and is evaluated by means of discrete-time analysis. The model is validated experimentally. However, their experimental setup does not include the virtualization layer.

In [149] Faraci *et.al.* propose Markov model of an Software Defined Networking (SDN)/Network Functions Virtualization (NFV) node consisting of a *Flow Distributor*, a *processor*, and different *Network Interface Cards*. Numerical results are provided for different input parameters. However, no validation is performed.

In [151] Duan copes with the composite network-cloud service provisioning assuming Service-Oriented Architecture (SOA) for both network virtualization and cloud computing. The author models the composite network-cloud service provisioning as a queuing system where each entity is modeled as latency-rate server and employs deterministic network calculus theory to derive the worst-case performance. Numerical examples are provided, but no validation is carried out.

The modeling of software-defined networks is addressed in [152] and [153]. Both works employ deterministic network calculus theory to model SDN switches and its interactions with the SDN controller.

Previously in Chapter 4, we proposed a model based on an open Jackson's network to evaluate the performance and scalability of a vMME with a three-tier architecture. The performance model for chain of VNFs included in the present chapter enhances that model by extending its applicability domain, increasing its flexibility to capture more complex operations, and using a more accurate methodology of analysis.

5.3 System Model

Let us assume a chain of VNFs (see Fig. 5.3), where each VNF might consist of multiple VNFCs working together. Each VNFC provides a well-defined part of the VNF functionality. In turn, each VNFC might have several instances and each VNFC instance is placed on a single Virtual Machine (VM) on which it runs.

In this work we do not address the containerization which is an OS-level virtualization method. We consider that two different VMs might offer distinct performance, even if they host instances of the same VNFC. The rationale of this is because the VMs have different number of CPU cores allocated or due to

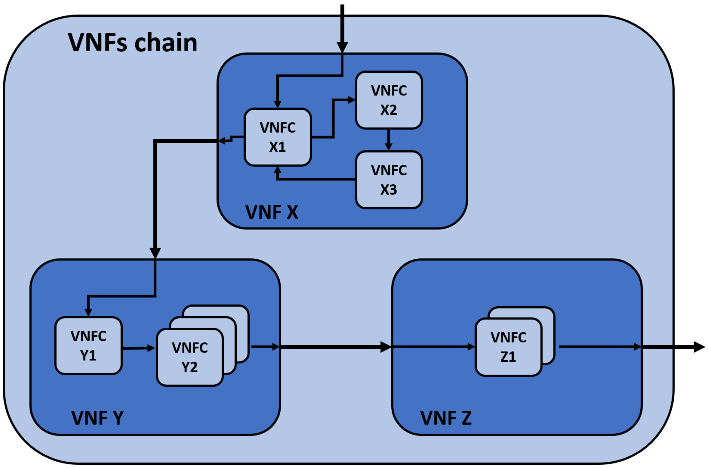


Figure 5.3: Chain of VNFs that is composed of the VNFs *X*, *Y*, and *Z*. VNFs *X*, *Y*, and *Z* have respectively 3 (*e.g.*, *X1*, *X2*, and *X3*), 2 (*e.g.*, *Y1*, and *Y2*), and 1 (*e.g.*, *Z1*) VNFCs. VNFCs *Y2* and *Z1* have respectively 3 and 2 instances.

the heterogeneity of hardware in the data center.

Here, without loss of generality, we shall consider that all the VNFs of the chain are running in the same data center. This data center comprises several server machines interconnected through a physical switch (see Fig. 5.4).

Each server hosts a hypervisor and one or several VMs. For each VM, the hypervisor emulates a virtual Network Interface Card (vNIC) by using an associated back-end driver [154]. The physical Network Interface Card (pNIC) of the server machine is connected to each vNIC through a virtual bridge [154]. The virtual bridge sends a packet to a specific VM by forwarding it to the corresponding back-end driver. The reception of a packet in the back-end driver generates a software interruption that the guest Operating System (OS) handles when the VM is executed. On the VM side, this interruption triggers the load and execution of a service routine to process the packet header and store the packet in a queue located in the user space of the RAM [150]. The packets are stored in the queue of the RAM until the application requests them for processing. The transmission of packets is conducted on the opposite path in a similar way.

Each VNFC instance serves the packets stored in the queue of the RAM

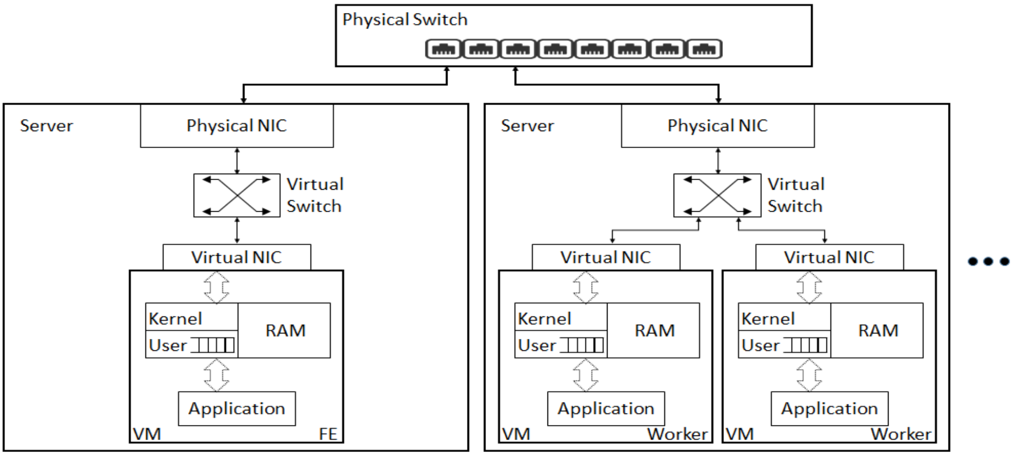


Figure 5.4: Top of Rack switch interconnecting two PMs. Each PM might host several VMs which are in turn interconnected through a virtual switch.

following a FCFS discipline. The service rate, μ_k, of the VNFC instance k is given by its computational resource acting as the bottleneck (*e.g.*, CPU or I/O).

Here we will focus on VNFCs executing CPU-intensive tasks. Additionally, we will suppose that each VM has dedicated physical CPU cores. Then, any VNFC instance running in the VM does not experiment dynamic changes in performance at runtime [155].

We shall also assume the physical network supporting the connection between any couple of PMs (*i.e.*, Top of Rack switch and physical links) has enough capacity to comfortably support the network I/O workload demands of all hosted VMs. In addition, each hypervisor has allocated enough computational capacity to comfortably withstand the operations associated with the transmission and delivery of network traffic from/to its VMs.

Under this assumption, the mean inter-VMs link delay, d_{ki}, between the vNICs of any two VMs k and i is roughly constant.

For the case of two VMs hosted on different physical servers, the inter-VMs link delay consists of:

- The back-end driver processing time and the packet transmission to the pNIC through the virtual bridge at the source physical server and the opposite path at the destination physical server [154].

- The transmission and propagation delays for the link between the source PM and the physical switch, and the link connecting the physical switch with the destination PM.
- The processing delay of the physical switch.

Note that in this case the inter-VMs link delay of two VMs hosted on the same PM only includes the latencies described in the first bullet point above.

5.4 Analytical Model for Chains of VNFs

This section explains the queuing model for a chain of VNFs and the Queuing Network Analyzer (QNA) method. QNA is the methodology of analysis considered to derive the performance metrics from the model.

5.4.1 Queuing Model

Let us consider a chain of VNFs with L VNFs. Each VNF $l \in [1, L]$ is composed of J_l different VNFCs. Moreover, every VNFC $j \in [1, J = \sum_{l=1}^{L} J_l]$ of the chain might have multiple instances (horizontal scaling).

Let K_j denote the number of instances per each VNFC j. To model this system we will employ an open network of $K = \sum_{j=1}^{J} K_j$ G/G/m queues $Q_1, Q_2, \cdots, Q_K$ (see Fig. 5.5), where each queue represents a VNFC instance running on a VM. Figure 5.5 shows the queuing model associated with the chain of VNFs depicted in Fig. 5.3.

The inter-VMs link delay between the VNFC instances i and k are modeled as infinite servers, *i.e.*, its response time is independent of the workload, with service rate $z_{ik} = 1/d_{ik}$. For the sake of simplicity, these infinite servers are not depicted in Fig. 5.5.

Every queue, which operates under FCFS discipline, might have m_k servers which represent different physical CPU cores processing messages in parallel. All the servers of the same queue have an identical and generalized service process, which is also characterized by its mean μ_k (service rate) and its Squared Coefficient of Variation (SCV) c_{sk}^2. However, servers belonging to different queues

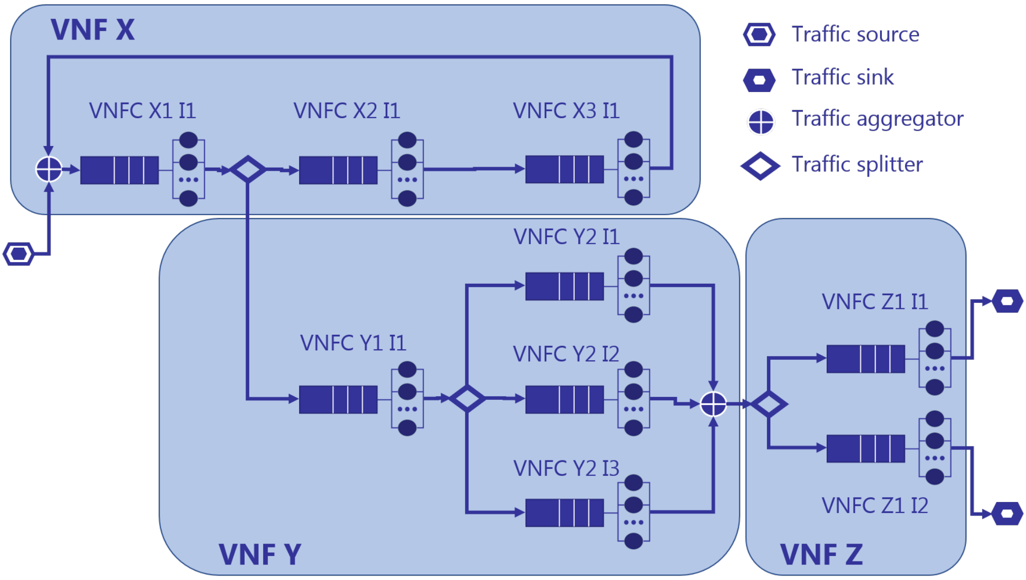


Figure 5.5: Queuing model for the chain of VNFs shown in Fig. 5.3.

may have distinct service processes, even if they represent instances of the same VNFC. This feature is useful to model the heterogeneity of the physical hardware, underlying the provisioned VMs, inherent to non-uniform infrastructures like computational clouds [155].

Regarding the external arrival process to each queue Q_k, it is assumed to be a generalized inter-arrival process, which is characterized by its mean λ_{0k} and its SCV, calculated as $c_{0k}^2 = variance/(mean)^2$.

Furthermore, every queue has a parameter ν_k associated with it, which is a multiplicative factor for the flow leaving Q_k that models the creation or combination of packets at the nodes. This means that if the total arrival rate to queue Q_k is λ_k, then the output rate of this queue would be $\nu_k \lambda_k$. The parameter ν_k can be used, for instance, to model packet dropping in a virtualized firewall or the video encoding rate modification in a video transcoder.

For the transitions between queues, we assume probabilistic routing where the packet leaving Q_k is either next moved to queue Q_i with probability p_{ki} or exits the network with probability $p_{0k} = 1 - \sum_{i=1}^{K} p_{ki}$.

Table 5.2: Model input parameters.

Notation	Description
λ_{0k}	Mean external arrival rate at queue Q_k.
c^2_{0k}	SCV of the external arrival process at queue Q_k.
m_k	Number of servers at queue Q_k.
μ_k	Average service rate at queue Q_k.
c^2_{sk}	SCV of the service process at queue Q_k.
$K^{(j)}$	Number of instances of the jth stage.
$P = [p_{ik}]$	Routing probability matrix.
ν_k	Multiplicative factor for the flow leaving Q_k.
d_{ik}	Link delay between queues Q_i and Q_k.

We also consider the routing decision is made independently for each packet leaving queue Q_k. The transition probabilities p_{ki} are gathered in the routing matrix denoted as $P = [p_{ki}]$. This approach allows to define any arbitrary feedback between VNFC instances, and to model caching effects and different load-balancing strategies at any VNFC.

5.4.2 System Response Time

To compute the system response time, we will use the QNA method which is an approximate analytical technique for solving open networks of G/G/m queues [156]. The QNA method uses two parameters, the mean and the SCV, to characterize the arrival and service time processes for every queue. This is the reason why this kind of method is sometimes referred to as two-moment method. Table 5.2 summarizes the input parameters of the QNA method.

The QNA method resembles the methodology of analysis for open Jackson networks.

- First, the mean and SCV of the inter-arrival times at every queue are computed.
- Second, the different queues are analyzed in isolation as standard G/G/m queues.
- Finally, the global performance metrics are computed by assuming the queues are stochastically independent, even though the queuing network

might not have a product form solution.

As a consequence, QNA method can be seen as a generalization of the open Jackson network of M/M/m queues to an open Jackson network of GI/G/m queues. In fact, QNA is consistent with the Jackson network theory, *i.e.*, if all the external arrival and service processes are Poisson, then QNA is exact [156].

As we will show in Section 5.8.2, although the QNA method is approximate, it performs well to estimate the global mean response time of a VNF with multiple VNFCs. In the following subsections, we will describe the main steps of the QNA method in detail.

5.4.2.1 Internal Flows Parameters Computation

The first step of the QNA method is to compute the mean and the SCV of the arrival process to each queue.

Let λ_k denote the total arrival rate to queue Q_k. As in the case of open Jackson networks, we can compute λ_k, $\forall$ $\{k \in \mathbb{N} | 1 \leq k \leq K\}$ by solving the following set of linear flow balance equations

$$\lambda_k = \lambda_{0k} + \sum_{i=1}^{K} \lambda_i \nu_i p_{ik} \tag{5.12}$$

The most interesting aspect of the QNA method is that it estimates the SCV of the aggregated arrival process c_{ak}^2 to each queue Q_k from a set of linear equations. To do this, the QNA method approximates c_{ak}^2 as a convex combination of the asymptotic value of the SCV $(c_{ak}^2)_A$ and the SCV of an exponential distribution ($c_{exp}^2 = 1$), *i.e.*, $c_{ak}^2 = \alpha_k (c_{ak}^2)_A + (1 - \alpha_k)$.

The asymptotic value can be found as $(c_{ak}^2)_A = \sum_{i=1}^{K} q_{ik} c_{ik}^2$, where q_{ik} is the proportion of arrivals to Q_k that came from Q_i. That is, $q_{ik} = (\lambda_i \cdot \nu_i \cdot p_{ik})/\lambda_k$. And α_k is a function of the server utilization $\rho_k = \lambda_k/(\mu_k \cdot m_k)$ and the arrival rates. This approximation yields the following set of linear equations, which may be solved to get c_{ak}^2, $\forall$ $\{k \in \mathbb{N} | 1 \leq k \leq K\}$:

$$c_{ak}^2 = a_k + \sum_{i=1}^{K} c_{ai}^2 b_{ik}, \qquad 1 \leq k \leq K \tag{5.13}$$

$$a_k = 1 + \omega_k \left\{ (q_{0k} c_{0k}^2 - 1) + \sum_{i=1}^{K} q_{ik} [(1 - p_{ik}) + \nu_i p_{ik} \rho_i^2 x_i] \right\} \tag{5.14}$$

$$b_{ik} = \omega_k q_{ik} p_{ik} \nu_i (1 - \rho_i^2) \tag{5.15}$$

$$x_i = 1 + m_i^{-0.5} (max\{c_{si}^2, 0.2\} - 1) \tag{5.16}$$

$$\omega_k = \big(1 + 4(1 - \rho_k)^2 (\gamma_k - 1)\big)^{-1} \tag{5.17}$$

$$\gamma_k = \left(\sum_{i=0}^{K} q_{ik}^2 \right)^{-1} \tag{5.18}$$

The above equations prevent us from monitor the arrival process at each VNFC instance. Instead it is only required to know the first and second order moments of the external arrival processes, thus saving computational capacity for monitoring purposes.

5.4.2.2 Response Time Computation per Queue

Once we have found λ_k and c_{ak}^2 for all internal flows, we can compute the performance parameters for each queue, which are analyzed in isolation (*i.e.*, considering that the queues are independent of each other).

Let W_k be the mean waiting time at queue Q_k. Then, the mean response time at queue Q_k is given by $T_k = W_k + 1/\mu_k$.

If Q_k is a GI/G/1 queue (Q_k has only one server), W_k can be approximated as:

$$W_k = \frac{\rho_k \cdot (c_{ak}^2 + c_{sk}^2) \cdot \beta}{2 \cdot \mu_k (1 - \rho_k)} \tag{5.19}$$

with

$$\beta = \begin{cases} exp(-\frac{2 \cdot (1-\rho_i) \cdot (1-c_{ai}^2)^2}{3 \cdot \rho_i \cdot (c_{ai}^2 + c_{si}^2)}) & c_{ai}^2 < 1 \\ \beta = 1 & c_{ai}^2 \geq 1 \end{cases} \tag{5.20}$$

If, by contrast, Q_k is a GI/G/m queue, W_k can be estimated as:

$$W_k = 0.5 \cdot \left(c_{ai}^2 + c_{si}^2\right) \cdot W_k^{M/M/m} \tag{5.21}$$

where $W_k^{M/M/m}$ is the mean waiting time for a M/M/m queue, which can be computed as:

$$W_k^{M/M/m} = \frac{E_C(m_k, \frac{\lambda_k}{\mu_k})}{m_k \mu_k - \lambda_k} \tag{5.22}$$

and $E_C(m, \rho)$ represents the Erlang's C formula which has the following expression:

$$E_C(m, \rho) = \frac{\left(\frac{(m \cdot \rho)^m}{m!}\right) \cdot \left(\frac{1}{1-\rho}\right)}{\sum_{k=0}^{m-1} \frac{(m \cdot \rho)^k}{k!} + \left(\frac{(m \cdot \rho)^m}{m!}\right) \cdot \left(\frac{1}{1-\rho}\right)} \tag{5.23}$$

5.4.2.3 Global Response Time Computation

For the overall mean response time of the chain of VNFs, T, we can distinguish two delay contributions, *e.g.*, the overall mean sojourn time of the VNFCs instances, T_{VNFCs}, and the overall mean sojourn time of the inter-VMs links, T_{IVMLs}.

T_{VNFCs} and T_{IVMLs} are respectively the mean total time any packet spends in VNFCs instances and inter-VMs links during its lifetime in the VNF. Then:

$$T = T_{VNFCs} + T_{IVMLs} \tag{5.24}$$

$$T_{VNFCs} = \sum_{k=1}^{K} (W_k + \frac{1}{\mu_k}) \cdot V_k \tag{5.25}$$

$$T_{IVMLs} = \sum_{k=1}^{K} \sum_{i=1}^{K} d_{ki} \cdot p_{ki} \cdot V_k \tag{5.26}$$

Where V_k denotes the visit ratio for VNFC instance k (Q_k) which is defined as the average number visits to node Q_k by a packet during its lifetime in the network. That is

$$V_k = \lambda_k / (\sum_{k=1}^{K} \lambda_{0k}) \tag{5.27}$$

5.5 Case Study: a Three-Tier vMME

The Mobility Management Entity (MME) is the main control entity of the Long-Term Evolution (LTE)/Evolved Packet Core (EPC) architecture. It interacts with the evolved NodeB (eNB), Serving Gateway (S-GW), and Home Subscriber

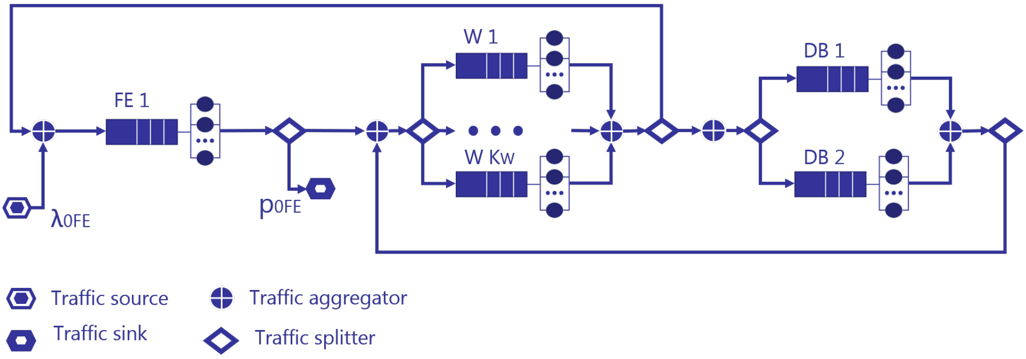


Figure 5.6: Queuing model for the vMME with a three-tier design shown in Fig. 4.1.

Service (HSS) within the EPC to realize functions such as Non-Access Stratum (NAS) signaling, user authentication and authorization, mobility management (e.g. paging, user tracking), and bearer management, among many others [108].

In this section, we will particularize the model for chains of VNFs to a vMME with a three-tier architecture, as described in Section 4.1. Here, we will consider the same operation for the three-tiered vMME as that one assumed in [75] and [76]. In a nutshell, the Worker (W) will retrieve the User Equipment (UE) context from DataBase (DB) when the initial message of any signaling procedure arrives, and it will save the updated UE context into the DB when the W finishes processing the last message of any signaling procedure [75, 76]. Consequently, a W instance must have enough memory to store all the required state data (*e.g.*, UE context) of all the control procedures it is serving at a given time. This operation reduces the load of the DB compared to the vMME operation described in Section 4.3.

5.5.1 Queuing Model for the vMME

Figure 5.6 depicts the queuing model associated with the three-tiered vMME shown in Fig. 4.1. Observe that, in contrast to the queuing model for the vMME developed in Chapter 4, this model includes the feedback between tiers, thus enabling the model to capture the actual flow of the control messages through the tiers. Moreover, this model considers a queue per tier instance and arbitrary service and external arrival processes.

5.5.1.1 Signaling Workload for the vMME

As described in Chapter 3, the UEs run applications that generate or consume network traffic. This UE activity along with the UE mobility trigger the LTE network control procedures. These signaling procedures allow the Control Plane (CP) to manage the UE mobility and the data flow between the UE and Packet Data Network Gateway (P-GW). Each of these control procedures yields several signaling messages to be processed by the vMME.

Here, we will only consider the most frequent LTE signaling procedures, *e.g.*, Service Request (SR), S1-Release (S1R), and X2-based Handover (HO) [97]. Let λ_{sp} be the mean generation rate of the signaling procedure $sp \in SP = \{SR, S1R, HO\}$ in the LTE network. Then, we can compute the frequency of occurrence of the signaling procedure sp as

$$f_{sp} = \frac{\lambda_{sp}}{\sum_{p \in SP} \lambda_p} \tag{5.28}$$

Similarly, the average number of control packets to be processed by the vMME per signaling procedure N_{pp} can be computed by

$$N_{pp} = \sum_{sp \in SP} f_{sp} \cdot n_{sp} = 3 \cdot f_{SR} + 3 \cdot f_{S1R} + 2 \cdot f_{HR} \tag{5.29}$$

where n_{SP} is the number of packets to be processed by the vMME for the control procedure sp. Specifically, $n_{SR} = n_{S1R} = 3$ and $n_{HR} = 2$.

The external packet arrival rate at the vMME is given as

$$\lambda_{vMME} = N_{pp} \cdot \sum_{sp \in SP} \lambda_{sp} \tag{5.30}$$

Finally, as shown in Section 3.6.3.2, the aggregated signaling generation process in an LTE network is Poisson. Then, the SCV of the inter-arrival times of control messages to the vMME equals one.

5.5.1.2 Transition Probabilities

Let K_{FE}, K_W, and K_{DB} respectively denote the number of Front-End (FE), W, and DB instances ($K = K_{FE}+K_W+K_{DB}$). To derive the transition probabilities of the queuing model, we will assume perfect load balancing for all tiers [157]. That is, each FE, W, and DB instance respectively processes $1/K_{FE}$, $1/K_W$, and $1/K_{DB}$ fraction of the total workload of the tier they belong to.

Considering the operation described in [75], there are two DB accesses per control procedure. Therefore, the visit ratio per packet at each DB and W instance will respectively be $V_{DB} = 1/K_{DB} \cdot 2/N_{pp}$ and $V_W = 1/K_W \cdot (1 + 2/N_{pp})$.

The FE maintains 3GPP standardized interfaces towards other entities of the network (*e.g.*, eNBs, HSS, and S-GW), thus all control messages from outside arrive at the FE tier. For the same reason, the vMME sends all response messages generated by W instances out on FE tier. Then, the visit ratio at each FE instance is given by $V_{FE} = 2/K_{FE}$, *i.e.*, each packet visits the FE tier two times. Furthermore, a packet served at any FE instance leaves the vMME (queuing network) with probability $p_{0FE} = 0.5$.

Consequently, the transition probabilities between the VNFC instances of the vMME are given by:

$$p_{FE \to W} = \frac{1}{K_W} \cdot \frac{1}{2} \tag{5.31}$$

$$p_{W \to FE} = \frac{1}{K_{FE}} \cdot \frac{1}{(1 + \frac{2}{N_{pp}})} \tag{5.32}$$

$$p_{W \to DB} = \frac{1}{K_{DB}} \cdot \frac{\frac{2}{N_{pp}}}{(1 + \frac{2}{N_{pp}})} \tag{5.33}$$

$$p_{DB \to W} = \frac{1}{K_W} \tag{5.34}$$

5.6 Experimental Procedures

This section presents the experimental setup, the procedures used to measure the input parameters for the model, and a description of the experiments carried out to validate the performance model for chains of VNFs.

5.6.1 Experimental Setup

The experimental setup includes three software tools:

i) a traffic source,

ii) an LTE network emulator, and

iii) a vMME with a three-tier design.

All of these tools were implemented in C/C++.

The traffic source generates LTE procedure calls according to the compound traffic model and the scenario considered in [108]. It only emulates the triggering of the most frequent LTE signaling procedures (*e.g.*, SR, S1R, and X2-based HO) [97]. The minimal inter-departure time supported by the traffic source is 5 μs.

The LTE network emulator reproduces the eNB and S-GW operation, *i.e.*, it processes and generates the signaling messages the eNB and S-GW would exchange with the vMME. It also emulates the latencies between the vMME and these LTE logical entities by introducing a constant delay to every incoming and outgoing packet. For all the experiments carried out, the two-way delay between the vMME and network emulator was set to 9 ms.

The three-tier vMME follows the behavior described in described in Section 5.5. Although the implementation is not fully 3GPP compliant, it performs similar operations. The database tier was implemented by using SQLite3 [158] entirely loaded in RAM.

Regarding the hosting environment, the experimental framework includes different kinds of physical servers. There are three servers with Intel(R) Core(TM) i7-6700K CPU at 4.00GHz with 4 cores, which are referred as *type I* servers. And one server with two Intel(R) Xeon(R) E5-2603 CPUs at 1.70GHz, with 6 cores each, which is referred as *type II* server. All the servers have a 10 Gbps Ethernet Network Interface Card (NIC), 32 GB of RAM memory, and run Ubuntu Server 16.0. All these servers are interconnected by means of an 8-port 10 Gbps Ethernet switch.

As virtualization environment, Kernel-based Virtual Machine (KVM) for Linux kernel was used. Each of the physical servers runs a KVM hypervisor [159].

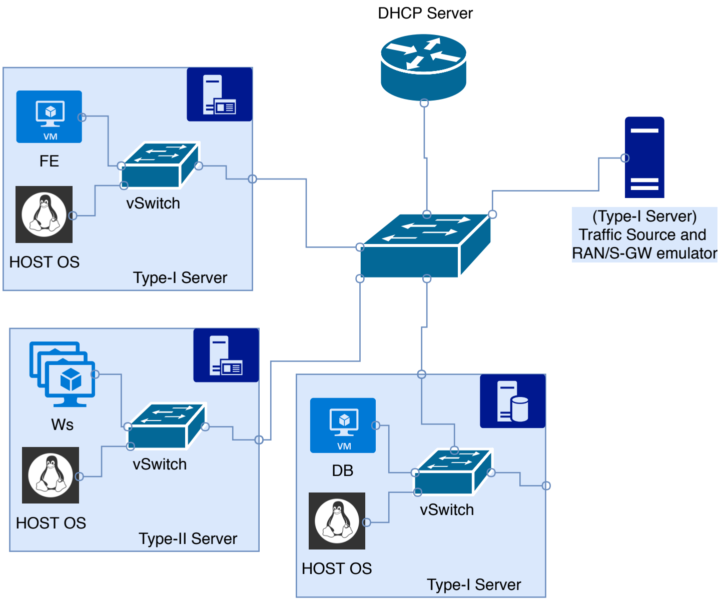


Figure 5.7: Experimental setup to validate the performance model for chains of VNFs.

For all KVM guests, the NICs were paravirtualized[1] with the Linux standard *virtio*, and bridged networking was used [159].

In the experiments, each VNFC instance of the vMME (*e.g.*, FE, W, and DB) runs on a different VM. The VMs hosting the FE and DB instances run on separated *type I* servers, whereas the VMs hosting the W instances run on the *type II* server. The traffic source and network emulator run on the other *type I* server.

In order to achieve real-time operation for the vMME, we set several CPU related configurations [160]. Specifically, the hyperthreading feature, the dynamic frequency scaling governor, and the processor C-States were disabled. In addition, CPU pinning was used and the affinity of the processes were configured in order to allocate one dedicated physical core to each VNFC instance [159]. Figure 5.8 shows a comparison between the service time obtained for a processing instance

[1] The paravirtualization is a technique where the physical hardware is not emulated, thus improving the virtualization performance. Instead, the guest OS is aware that it is virtualized and provides a particular driver for each hardware component to communicate with the back end driver on the hypervisor [154].

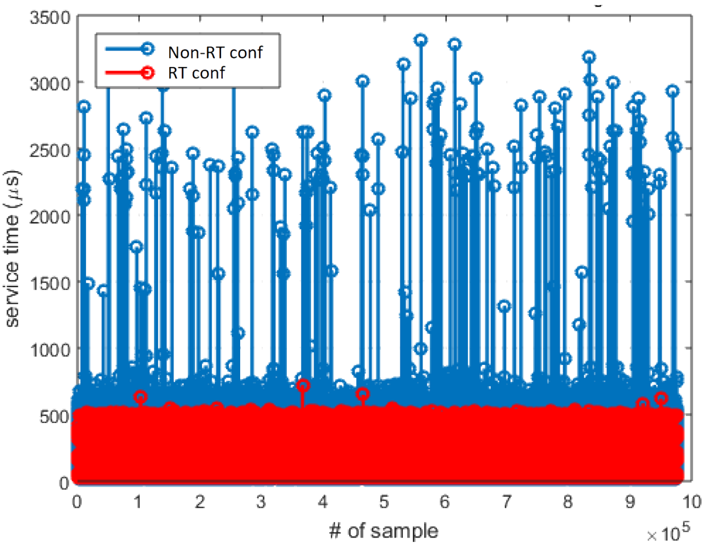


Figure 5.8: The service time samples of a W instance when the real-time configuration is enabled and disabled.

when the real-time configuration is enabled (labeled as "*RT conf*") and disabled (labeled as "*Non-RT conf*"). As it is observed, the variance and the maximum value of the service time increase when the real-time configuration is disabled. The adoption of NFV in mobile networks requires the real-time operation of their virtualized functional entities as they have to meet stringent maximum delay constraints 99.999% of the time.

The Linux kernel version *4.4.0-81-generic* default settings was used for all the networking buffers, *e.g.*, the receive and send socket buffers, *rmem_default* and *wmem_default*, was fixed at 212992 bytes; and the buffer reception at any interface, *netdev_max_backlog*, was fixed at 1000 packets. With this setting, a negligible probability of packet loss was observed in all the experiments.

5.6.2 Parameter Estimation

This section addresses the procedures used to estimate the input parameters listed in Table 5.2 by means of measurements.

5.6.2.1 External Arrival Process

It was estimated by recording the arrival times of the packets at the external interfaces of the VNF. Samples of the inter-arrival time, IAT, can be obtained as the difference between the arrival instants of two consecutive packets. Then, the first and second order moments, $E[IAT] = 1/\lambda_0$ and $VAR[IAT] = c_{a0}^2 \cdot E[IAT]^2$, of IAT can be estimated as the sample mean and variance.

5.6.2.2 Service Processes

The service process for each VNFC was characterized by taking measurements of the service time directly from the source code of the application. That is, by reading the system clock at the beginning and the end of the execution of the code which implements the packet processing. Samples of the service time, s_k, were taken for every processed packet at the VNFC instance k. Then, the first and second order moments of s_k, $E[s_k] = 1/\mu_k$ and $VAR[s_k] = c_{sk}^2 \cdot E[s_k]^2$, were estimated as the sample mean and variance.

In order to ensure that the above measurements are good approximations of the actual service time at any VNFC instance, the following measurement process was tried. With the VNFC instance fully overloaded (*i.e.*, the queue is never empty), the departure times of the outgoing packets were monitored and recorded. Then actual samples of the service time can be obtained as the difference between the departure instants of two consecutive outgoing packets. This estimation allows us to consider the VNFC as a black box, *i.e.*, the source code is not required.

We carried out an experiment where the service time process of a VNFC instance were measured by using both aforementioned techniques. The values measured for the mean and SCV of the service time were 155.08 μs and 1.06, respectively, by using the first methodology, and 157.29 μs and 1.03, respectively, with the second one. Hence, we can estimate the service time by taking measurements directly from the application.

5.6.2.3 Transition Probabilities

As shown in Section 5.5.1.2, the probability transition matrix (or equivalently the visit ratios) for the three-tiered vMME depends on the VNF internal operation and the percentages of each type of LTE control procedure. Since we know the VNF internal operation beforehand, we only needed to monitor the frequency of occurrence of each considered signaling procedure sp, f_{sp}, at the FE.

In a more general scenario, the transition probability matrix can be estimated by using counters at each VNFC instance to monitor the number of incoming packets and the outgoing packets towards other VNFC instances.

5.6.2.4 Mean inter-VM Link Delays

In the experimental setup there was no any mechanism to synchronize the clock of the different physical servers. Then, in order to estimate the mean inter-VM link delays, d_{ki}, between the VNFC instances k and i, an echo service was employed. Let us assume we want to measure d_{ki}, and the echo server is running in the same VM as the VNFC instance i. At the VNFC instance k, the departure time of the query message, $Q_k^{(out)}$, and the arrival time of the response message, $R_k^{(in)}$, were recorded. At the VNFC instance i, the arrival and departure instants of the query and response messages, $Q_i^{(in)}$ and $R_i^{(out)}$, were collected. Then, assuming symmetric inter-VM links between k and i, *i.e.*, $d_{ki} = d_{ik}$, the mean inter-VM link delay, d_{ki}, between k and i might be estimated as the sample average $(1/2) \cdot ((R_k^{(in)} - Q_k^{(out)} - (R_i^{(out)} - Q_i^{(in)})))$.

5.6.3 Experiments

Five scenarios respectively with 1, 2, 3, 4, and 5 worker instances were considered. These scenarios are referred to as *S1*, *S2*, *S3*, *S4*, and *S5*, respectively. There was only one DB and FE instance for all of them. Several signaling workload points were evaluated for each of them. Specifically, we assessed 10, 11, 13, 16, and 19 workload points for *S1*, *S2*, *S3*, *S4*, and *S5*, respectively. The maximum signaling workload evaluated for *S5* was 17000 control packets per second. Each experiment, *i.e.*, a signaling workload point for a given scenario, was repeated 5 times. As stop condition for all the experiments, the vMME processed 200000 signaling procedures.

The measurement tools employed in all the experiments were network sniffers monitoring the incoming and outgoing traffic at the vNIC of the VM hosting each VNFC instance. To measure the vMME response time, we recorded the

Table 5.3: Main Parameters Configuration for Simulations

Signaling Rates per UE	
λ_{SR} (Poisson)	0.0045 procedures/second
λ_{S1R} (Poisson)	0.0045 procedures/second
λ_{HO} (Poisson)	0.0012 procedures/second
EPC Delays	
One-way delay (eNB → vMME)	4.5 ms
Two-way delay (vMME ⇌ [eNB | S-GW])	9 ms
Service Rates	
FE service rate (μ_{FE})	120000 packets per second
DB service rate (μ_{DB})	100000 transactions per second
W CPU power (r_{cpu})	$11.38 \cdot GFLOPS$
$(NI^{(H)}$ $H \in \{SR_1, SR_2, SR_3, SRR_1, SRR_2, SRR_3, HR_1, HR_2\}$ [108]	(1.45, 1.07, 1.06, 1.07, 1.07, 1.06, 1.07, 1.07) $\cdot 10^6 Instructions$

arrival time of each control message and the departure time of its corresponding response at the FE instance.

5.7 QNA Method Implementation Verification

The correctness of the QNA method implementation was verified by means of simulations as previous step to the experimental validation of the proposed performance model for chains of VNFs. In addition, QNA method was compared in terms of accuracy and computational complexity with the baseline approaches for analyzing queuing networks (*e.g.*, Jackson Networks and MVA).

The simulation setup employed is similar to that one described in Section 4.5.1, but considering the operation described in Section 5.5 for the three-tiered vMME. Table 5.3 includes the configuration of the main parameters used in the simulations.

In order to compare the QNA method with the baseline approaches for analyzing queuing networks, the Queueing Network Package for GNU Octave [161] was used. This package includes functions to solve queuing networks by using Jackson's networks assumptions (e.g., qnos function) and MVA algorithm (e.g., qncsmva function).

5.7.1 Execution Time of the QNA Methods

Table 5.4 includes the execution time taken for Jackson networks methodology, QNA method, MVA algorithm, and simulation approach to estimate the mean

Table 5.4: Execution times for solving the queuing network that models a vMME.

N_U	K	Jackson	QNA	MVA	Simulation
100000	3	6.667 ms	8.000 ms	9.740 s	224.280 s
500000	4	6.000 ms	9.340 ms	48.691 s	241.100 s
1000000	5	5.333 ms	10.667 ms	97.162 s	252.880 s
2000000	8	7.000 ms	21.335 ms	194.884 s	251.660 s

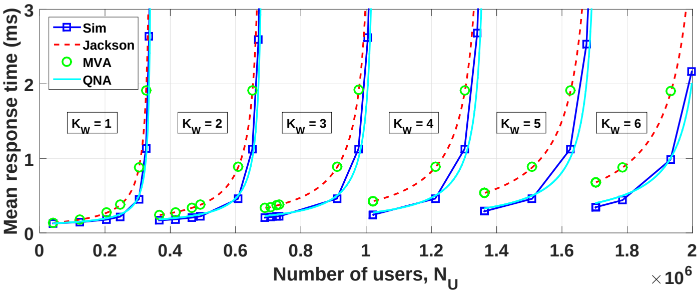


Figure 5.9: Mean response time of the three-tiered vMME estimated by using different approaches for analyzing networks of queues and measured by simulations.

performance metrics of the three-tiered vMME. Each measurement was repeated three times. The results included in Table 5.4 is the sample average of those three measurements performed for each approach.

As it can be observed, QNA method has a time complexity that depends on the number of queues in the network K, $O(K)$. Whereas the MVA algorithm [157] exhibits a time complexity that is a linear function of the number of active user sessions N_U and the number of queues in the network K, $O(K \cdot N_U)$.

Since N_U may be high in cellular scenarios, the MVA algorithm might even take longer than the simulation of the scenario. For all the simulations, the system has to process $4 \cdot 10^6$ packets as the stop condition. We checked that this achieves simulation convergence. The chosen stop condition explains that the simulation execution time does not depend strongly on N_U.

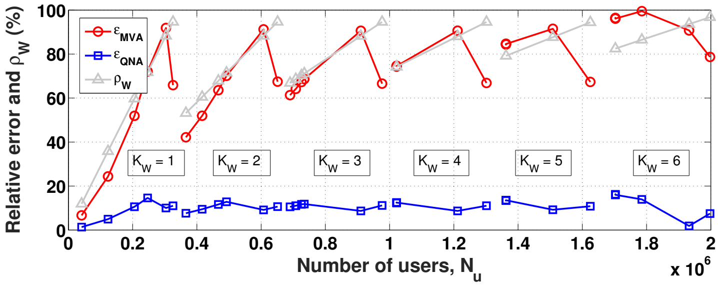


Figure 5.10: Comparison between the estimation error of the different methodologies for analyzing networks of queues.

5.7.2 QNA Method Accuracy

Figure 5.9 depicts the mean response time of the three-tiered vMME versus the number of users N_U estimated by using QNA method, MVA algorithm, Jackson Network methodology, and simulation. When T equals the fixed target mean response time, $T_{target} = 3ms$, a new W instance with a single CPU core is added. For $N_U = 2 \cdot 10^6$, the FE utilization $\rho_{FE} = 0.95$ and from this point on, the FE should be scaled to achieve $T \leq T_{target}$. Noteworthy, Jackson and MVA methods yield similar results for the case studied.

From the data showed in Fig. 5.9, the relative error for the different theoretical methodologies considered was computed as $\epsilon = \frac{|T_{sim} - T_{theo}|}{T_{sim}}$, where T_{sim} and T_{theo} are respectively the mean response time obtained by simulation and estimated by using the corresponding methodology.

Simulation results show that QNA method outperforms the baseline methodologies considered for analyzing network of queues in terms of accuracy for the use case studied. Specifically, QNA method offers an estimation error approximately equals 10%, whereas the estimation error of the baseline methodologies ranges roughly from 60% to 90% (see Fig. 5.10).

5.8 Model Validation

In this section, we will validate the analytical model for the three-tiered vMME use case. For this purpose, the mean response time of the vMME predicted by the analytical model is compared to that one obtained from the experimental testbed. In addition, we will compare these results with those provided by a Jackson's network model. Next, we will commence the section including the measured values for the input parameters of the model.

5.8.1 Measured Input Parameters for the Model

The service time measured in the application of each VNFC in the experimental testbed is depicted in Fig. 5.11. Results show that, except in the W case, the application service time has a low variability. This is due to each VM settled to a specific core and the rest of the system's processes to another. Additionally, the W service time presents a ladder shape. This is caused by the different processing tasks carried out by the W application to each packet as described in [111].

From the sample mean and variance of the application service time collected from the experimental testbed, the service rate μ and the SCV c_s^2 were estimated (see Table 5.5). These values are provided with its 95% confidence interval. The results show the FE application has the highest service rate, whereas the W has the lowest service rate of all considered VNFCs. This is the motivation behind the horizontal scaling of the W VNFC.

Additionally, we have measured the mean inter-VMs link delay d_{ki} between different VNFC instances (see Table 5.5) from the testbed up to a rate of 17000 packets per second. The measurements have yielded a nearly constant mean delay within the evaluated range.

Finally, we estimated the transition probabilities between VNFCs using (5.29), (5.31), (5.32), (5.33), and (5.34) (see Table 5.5). As shown in Section 5.5.1, they only depend on the VNF internal operation and the frequency of occurrence for each type of control procedure. For all the experiments, $f_{SR} = f_{S1R} \approx 0.44$ and $f_{HO} \approx 0.12$. Consequently, the visit ratio of each VNFC instance are $V_{FE} = 2$, $V_W = (1/K_W) \cdot 1.69$, and $V_{DB} = 0.69$.

5.8.2 Analytical Model Evaluation

In order to validate the parameters of the arrival processes for each VNFC instance, first, the relative error between the estimation of the SCVs of the internal arrival processes c_{ak}^2 was estimated by using (5.13) and the measured SCVs as explained in Section 5.6.2. A relative error sample was computed for each tested

Table 5.5: Measured input parameters for the model.

Service Processes	
FE service rate (μ_{FE})	115126 packets per second
FE service time SCV (c^2_{sFE})	0.0225 ± 0.0088
W service rate (μ_W)	6716 packets per second
W service time SCV (c^2_{sW})	0.6457 ± 0.0016
DB service rate (μ_{DB})	23874 transactions per second
DB service time SCV (c^2_{sDB})	0.0280 ± 0.0001
Transition Probabilities	
$p_{FE \rightarrow W}$	$\frac{1}{K_W} \cdot 0.5$
$p_{W \rightarrow FE}$	0.59
$p_{W \rightarrow DB}$	0.41
$p_{DB \rightarrow W}$	$\frac{1}{K_W}$
Mean Inter-VM Link Delays	
$d_{FE \rightarrow W} = d_{W \rightarrow FE}$	29.54 ± 0.22 μs
$d_{W \rightarrow DB} = d_{DB \rightarrow W}$	31.33 ± 0.38 μs

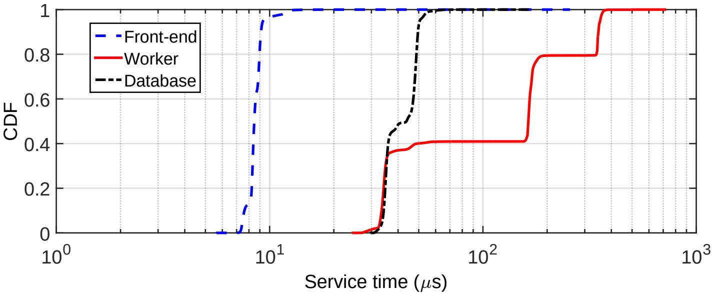


Figure 5.11: Service time process for each VNFC.

external arrival rate and each scenario (from 1 to 5 workers), and minimum, maximum, average and standard deviation values were calculated with these samples, see Table 5.6. As shown, the average error is approximately 26%, 24% and 8.5% for the FE, the Ws and the DB, respectively. It was observed that, for each scenario, the estimation error decreases with the load.

Fig. 5.12 shows the overall mean response time of the vMME, T, obtained experimentally (labeled as 'Exp') and computed using the model (labeled as

Table 5.6: Characterization of the relative error for the estimation of the SCVs of the internal arrival processes.

VNFC	min	max	avg	sdt
FE	2.29%	62.30%	26.20%	12.50%
W	0.03%	63.34%	24.10%	15.74%
DB	0.16%	40.88%	8.55%	9.36%

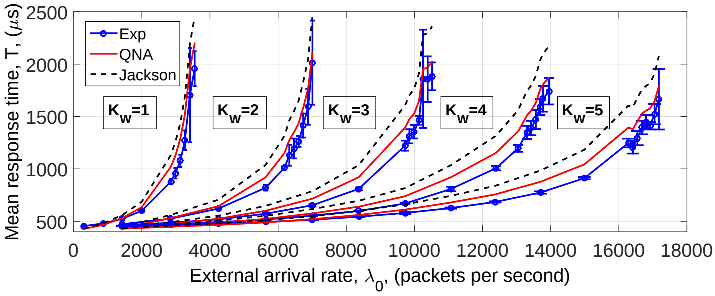


Figure 5.12: Overall mean system response time.

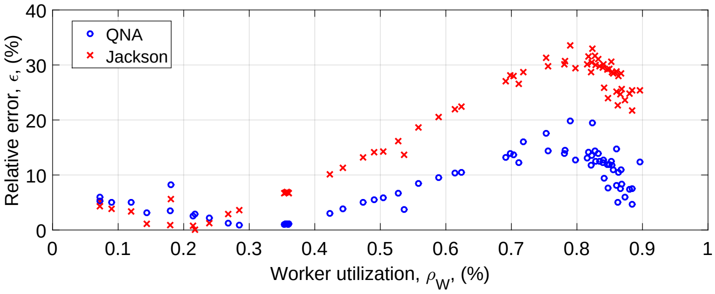


Figure 5.13: Model validation.

'QNA') and the method for analyzing Jackson's networks (labeled as 'Jackson'). This figure combines the results from the 5 executed scenarios, *i.e.* using from 1 to 5 workers. Additionally, each load point is executed 5 times and the mean value and the 95% confidence intervals are included. As shown, the QNA model closely follows the empirical curve.

Similarly, Fig. 5.13 presents a scatter plot of the relative error for the different analytical models considered. This error is calculated as $\epsilon = |T_{exp} - T_{theo}|/T_{exp}$, where T_{exp} and T_{theo} are the mean response time obtained experimentally and computed by using the corresponding model, respectively. As shown, the QNA model outperforms Jackson's approach for medium and high loads, achieving less than half of error. For low loads, both methods produce an error lower than 10%.

5.9 Conclusions

Performance modeling is an agile technique for evaluating the QoS of computer networks. In the context of SoftNets, it has two key applications: DRP and Network Embedding. These applications enable the automation of the deployment and scaling of chains of VNFs.

This chapter addresses the performance modeling of chains of VNFs. More precisely, an open QT network has been used as modeling technique. The model developed is sufficiently general to capture the complex behavior of VNFs chains. To solve the resulting queuing network, the QNA method [156] has been adopted. QNA method is an approximate technique to derive the performance metrics of a network of G/G/m queues.

The proposed performance model has been validated experimentally for an LTE vMME with a three-tier design use case. We have seen that the transition probabilities of the three-tiered vMME depend on the frequency of occurrence of each LTE control procedure and the vMME operation considered. The experimental setup employed, which emulates a virtualized Data Center (DC), and the procedures used to measure the input parameters for the model (e.g., external arrival process, service process, transition probabilities, and mean inter-VM link delays) have been detailed.

In the validation process, an experimental evaluation of the overall mean response time of the three-tiered vMME for different workloads and number of W instances has been conducted. The results obtained have been compared to those ones predicted by the performance model in terms of estimation error. The validation results have shown that, for medium and high workloads, QNA methodology achieves less than half of error compared to Jackson approach. For low workloads, both methods produce an error lower than 10%.

Chapter 6

Planning of the Virtualized EPC

Network Softwarization (NetSoft) paradigm is an overall approach for designing, implementing, deploying, managing, maintaining network equipment and/or network components by software programming [28]. Under the NetSoft approach, isolated, fully automated, programmable, flexible, and service-customized networks known as network slices can be deployed on top of a common physical infrastructure [11, 30, 31]. This is referred to as Network Slicing, an enabling technology that will allow the mobile operators to cover different scenarios and use cases with diverse (and potentially inompatible) requirements [32] (see Fig. 6.1).

The adoption of network slicing in 5G mobile networks requires solutions for planning the slices which should be optimized for the different use cases. Before instantiate a network slice, its template must be created and verified. It includes the allocation of all shared/dedicated resources to the particular network slice instance [162]. This process broadly involves defining the necessary functionalities of the network slice for each use case, the dimensioning of the required resources, and the resources allocation in a given infrastructure. Furthermore, the resource dimensioning and allocation process has to be done in a manner that ensures the Quality of Service (QoS) requirements for each use case.

A mobile network comprises mainly four domains (see Fig. 6.2):

i) Radio Access Network (RAN) which is responsible for all radio-related

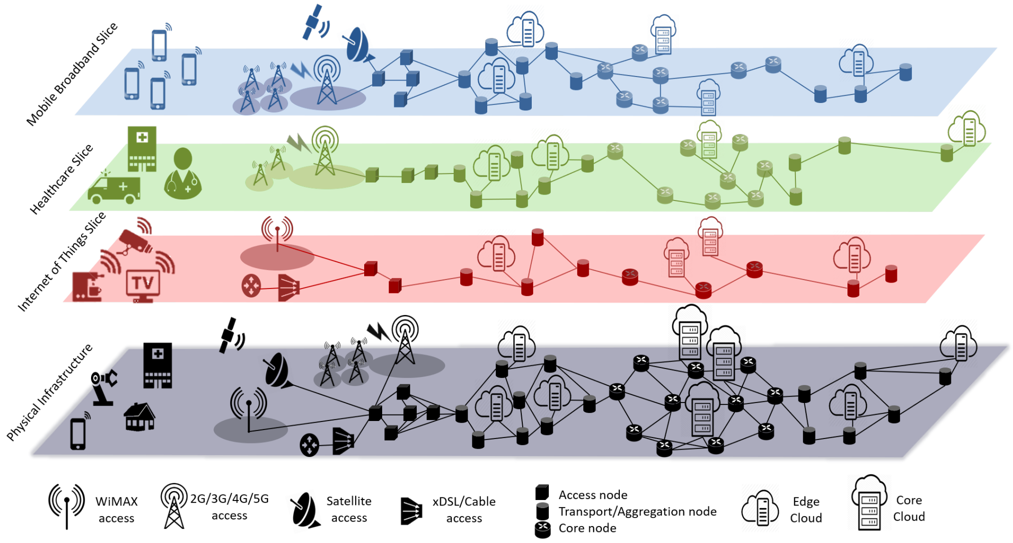


Figure 6.1: Network slicing concept: several network slices tailored for different use cases running on a common substrate infrastructure [11].

functionalities (e.g., radioresources scheduling, MIMO schemes, and coding, among others).

ii) Backhaul Network (BN) or transport network that provides connectivity among the base stations sites and the Core Network (CN).

iii) CN which is responsible for non-radio functions (e.g., authentication, charging, and bearer setup, among others)

iv) SGi-LAN refers to the set of all available service functions which can be used to establish different Service Function Chains for different services. It is presently used by mobile service providers to differentiate their services to their subscribers and reflect the business model of mobile operators [163]. It also provides connectivity among the internal application platforms such as IP Multimedia Subsystem (IMS) and the CN.

Each domain should be optimized for the different use cases and furnished with the corresponding network services. The four domains, together with their proper

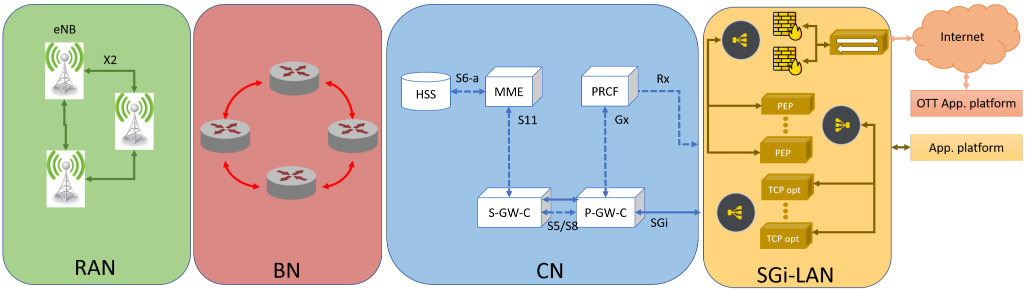


Figure 6.2: Domains of a mobile network.

network services, tailored for a particular use case, conforms a mobile network slice.

Additionally, although NetSoft paradigm enables operators to dynamically adapt the resources allocated to each network slice and services [164], the on-demand plans offered by cloud providers are more expensive than reservation plans. More precisely, resources can be purchased as a reservation for up to 70% off the on-demand price. Then, the resource dimensioning and allocation processes during the network slices planning is important for operators to save money.

In this chapter, we will focus on the planning of the CN slice segment of an Long-Term Evolution (LTE) network for the enhanced Mobile Broadband (eMBB) service group. To that end, we will address the joint optimization problem of resources dimensioning of the virtualized Evolved Packet Core (vEPC) and their allocation among a set of candidates Edge Clouds (ECs).

The remainder of the chapter is organized as follows. Section 6.1 briefly reviews the related literature. Section 6.2 describes the system model. Section 6.3 includes the formulation of the joint optimization problem of resource dimensioning and allocation for the vEPC. In Sections 6.4 and 6.5, respectively the modeling and analysis to estimate the performance of the Control Plane (CP) and Data Plane (DP) is developed. Next, in Section 6.6 we introduce a heuristic to perform the planning of the vEPC. Section 6.7 details the experimental setup. Section 6.8 provides numerical and simulation results that show the proper operation of the heuristic method to carry out the vEPC planning. Finally, Section 6.9 summarizes the main conclusions.

6.1 Related Works

This section briefly reviews the related literature. In particular, we will focus on performance models and embedding algorithms (*i.e.*, how to map Virtual Network Function Component (VNFC) instances to physical infrastructures) for the vEPC.

6.1.1 Modeling of the vEPC

Analytical models constitute an agile way to predict the performance of a system in advance. There are several proposals in the literature in tackling the analytic modeling of the vEPC [106, 108, 111, 137, 165]. Invariably, these woks employ Queuing Theory (QT).

In [106], Rajan *et al.* model the Evolved Packet Core (EPC) as a D/D/m node. They conclude that simply replacing existing EPC elements with virtualized equivalents can have severe performance bottlenecks and that vEPC elements need to be carefully designed. Prados *et al.* [108] analyze the performance of a virtualized Mobility Management Entity (vMME) with a three-tier architecture, inspired by web services, by using a Jackson's network, *i.e.*, a network of M/M/m queues. In that work, each queue represents a tier or VNFC of the vMME. Additionally, in this work we validate the model by simulation and show that it provides fairly good results for computational resources dimensioning. In [111], we enhance the previous model by extending its applicability domain to any chain of Virtual Network Functions (VNFs), increasing its flexibility, and using a more accurate technique of analysis. Specifically, each VNFC instance is modeled as a G/G/m queue. The resulting network of queues is solved by using the approximated technique proposed by Whitt *et al.* in [156] for the Queuing Network Analyzer (QNA), hereinafter referred to as QNA method. In this case, we particularize the model to a three-tiered vMME and compared QNA method with other standard techniques for analyzing network of queues (*e.g.*, Jackson method and Mean Value Analysis (MVA)). The paper shows that QNA method outperforms Jackson and MVA techniques in terms of the system response time estimation error.

Tanabe *et al.* [165] propose a bi-class (*e.g.*, Machine-to-Machine (M2M) and Mobile Broadband (MBB) communications) queuing model for the vEPC. The

CP and DP of the vEPC are respectively modeled as M/M/m/m and M/D/1 nodes. This model constitutes the core of the vEPC-ORA method which aim at optimizing the resource assignment for the CP and DP of the vEPC.

Finally, in [137], Ren *et al.* propose a Dynamic Resource Provisioning (DRP) algorithm for the vEPC considering the capacity of legacy network equipment already deployed. To evaluate the performance of their solution, they model each vEPC element as a M/M/m/K queue and assume that the VNF instantiation time is exponentially distributed.

The above works only model part of the EPC and/or do not capture the interactions between its elements. In the present work, this gap is covered. We will consider the main elements of the LTE network CP (*e.g.*, User Equipment (UE), evolved NodeB (eNB), Mobility Management Entity (MME), Serving Gateway (SGW), PDN (Packet Data Network) Gateway (PGW), Home Subscriber Service (HSS), Policy and Charging Rules Function (PCRF)) and their interactions. In this way it is possible to predict the performance of the whole LTE CP from the external arrival process (*i.e.*, aggregated generation signaling process).

In [137,165], the resources dimensioning of the vEPC is addressed. Nevertheless, these works address the dimensioning of each component in an isolated way. Then, it is necessary to define a processing delay budget for each entity to be dimensioned in advance. The holistic model for an LTE network detailed in Section 6.4 overcomes this limitation enabling the resources dimensioning algorithm to consider an overall processing delay budget for the whole EPC. This leads to an optimal dimensioning of the resources, *i.e.*, resources saving or maximal resources utilization.

For the DP, we leverage the results obtained in Chapter 3 for the analysis of the LTE data traffic traces to derive its performance metrics. Specifically, the vEPC DP is modeled as single queue fed by a fractal Brownian motion (fBm) process. To the best our knowledge, there is no previous work that employs stochastic network calculus results for analyzing the performance of the vEPC.

6.1.2 Algorithms for the vEPC Embedding

There is a rich and contributive literature on proposing algorithms to embed the whole vEPC or partially some of its entities in a physical infrastructure [139, 166–173].

Taleb *et al.* [166] propose a heuristic algorithm for virtualized S-GWs (vS-GWs) embedding. The algorithm tries to minimize the frequency of mobility gateway relocations while ensuring that a maximum capacity for each vS-GW, which handles the traffic load of a serving area, is not exceeded. This work is extended in [169] where some additional objectives and restrictions are considered. Regarding the objectives, the path betweenUEs and Packet Data Network Gateway (P-GW) is minimized, and the overall network resource utilization is optimized. Concerning the restrictions, this work was a pioneer in considering some relevant 3GPP constraints.

In [167], Bagaa *et al.* address the embedding of the virtualized P-GW (vP-GW). This work formulates the embedding problem as a multi-objective non-linear optimization problem which minimizes the costs for network operators, maximizes the network performance and balance the load equally among the vP-GW instances. To solve the problem, three heuristic algorithms are proposed that achieve near-optimal solutions.

In [168], Basta *et al.* investigates different approaches to deploy the core gateways (*e.g.*, Serving Gateway (S-GW) and P-GW) in the Data Centers (DCs). Specifically, they consider a fully and partially virtualization approach for the gateways. The former consists in moving the CP and DP functionalities of each gateway to a DC. The latter decouples CP and DP functionalities by using Software Defined Networking (SDN) paradigm and only the CP part is hosted within a DC. Finally, the authors formulates a problem to choose the virtualization approach and DC location to embed the gateways that minimize the total transport network load.

Martini *et al.* [170] formulate the problem of choosing the VNF instances provided by a distributed set of DCs to serve a given service chain request. The objective is to minimize the overall latency of the chain. This optimization problem can be formulated as a Resource Constrained Shortest Path problem.

Baumgartner *et al.* [139, 171] formulate the joint optimization problem of the virtual mobile core network topology, which is assumed to be unknown beforehand, composition and embedding. The formulation guarantees a maximum End-to-End (E2E) latency and taking into account the processing, queuing and propagation delays.

In [172], Laghrissi *et al.* address the embedding of the virtualized core gateways (*e.g.*, vS-GW and vP-GW) which is formulated as a Constraint Satisfaction Problem and solved by using the forward checking algorithm. This work considers and explores several optimization objectives for the vS-GW and vP-GW embedding.

Finally, Dietrich *et al.* [173] formulate a mixed-integer linear program for the vS-GW and vMME embedding. To reduce its time complexity, they transform it into a linear program by employing relaxation and rounding techniques. Their proposal mitigates the load imbalance in today's mobile networks, which leads to improvements in terms of request acceptance and resource utilization.

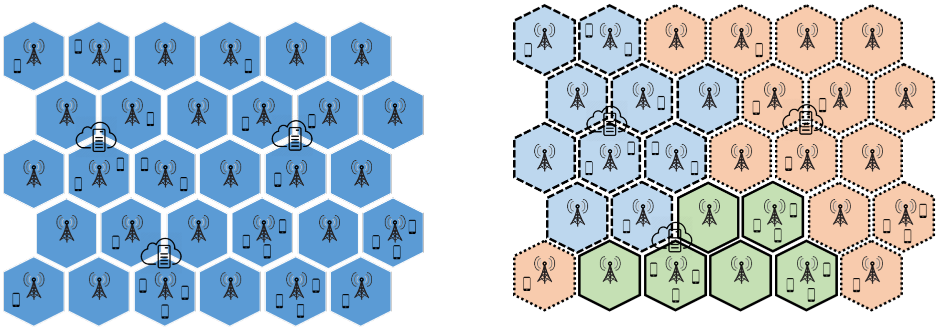
The resources dimensioning and embedding are treated throughout the literature as separate problems. These two stages of the resources allocation are closely related and to perform them in a coordinated way brings some benefits. For instance, as we will show there is a trade-off between the workload balance among a set of candidates DCs (propagation delays) and the resources utilization (processing delays) when an overall delay budget to be met is partitioned among these two stages. In this chapter, we formulate the joint optimization problem for planning the vEPC to address this trade-off.

6.2 System Model

This section proposes the abstract model for the considered LTE network architecture, where the EPC slices tailored for the eMBB use cases will be deployed. Additionally, we will review the performance requirements specified by the Third Generation Partnership Project (3GPP) for the EPC.

6.2.1 System Architecture

Let us assume an E-UTRAN already deployed and consisting of I eNBs, which provides connectivity to a set of J UEs to LTE EPC (see Fig. 6.3a). Each UE



(a) E-UTRAN deployment and ECs sites.

(b) Workload partitioning at a granularity of eNB among the candidate ECs.

Figure 6.3: E-UTRAN deployment and the workload distribution among the ECs sites.

i is attached to the eNB j. Let u_{ji} be a binary variable indicating whether the UE j is attached to the eNB i ($u_{ji} = 1$) or not ($u_{ji} = 0$).

We will assume the coverage map of this E-UTRAN as a rectangular area A with height h and width w. Within A, there are already deployed K ECs (see Fig. 6.3a).

Let $\mathbf{r_i^{(eNB)}} = (x_i^{(eNB)}, y_i^{(eNB)}) \ \forall \quad i \in \mathbb{N} \cap \{1, .., I\}$, $\mathbf{r_j^{(UE)}} = (x_j^{(UE)}, y_j^{(UE)})$ $\forall \quad j \in \mathbb{N} \cap \{1, .., J\}$, and $\mathbf{r_k^{(EC)}} = (x_k^{(EC)}, y_k^{(EC)}) \ \forall \quad k \in \mathbb{N} \cap \{1, .., K\}$ denote two dimensional vectors representing respectively the positions of the eNBs, UEs, and ECs within A.

The MME, S-GW, and P-GW of the EPC will be virtualized as a single VNF, referred to as vEPC, and deployed on the candidates ECs. Other EPC entities such as the HSS and the PCRF might be located outside of the ECs.

The signaling and data workload generation process of each UE is detailed in Section 3.3. The aggregated workload generated by the J UEs attached to the E-UTRAN is distributed among the K candidates ECs. This workload distribution is performed at a granularity of eNB (see Fig. 6.3b), *i.e.*, each eNB i is assigned to a candidate EC k. Let v_{ik} be a binary variable indicating whether the eNB i is assigned to the EC k ($v_{ik} = 1$) or not ($v_{ik} = 0$). To serve its corresponding workload, a full vEPC is instantiated on each EC.

The LTE network architecture assumed in this chapter is depicted in Fig.

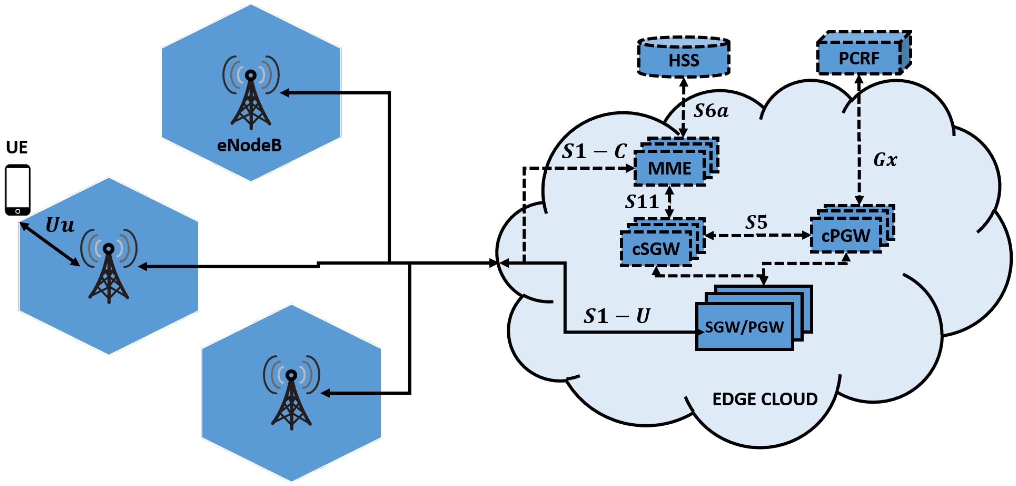


Figure 6.4: Assumed LTE network architecture.

6.4. The CP and the DP of the vEPC are fully decoupled. Each CP entity (*e.g.*, the MME, and the control functionalities of the S-GW and P-GW -cSGW and cPGW-) is implemented separately as a single VNFC. The DP functionalities of the S-GW and P-GW are integrated on a single VNFC. All of the VNFCs of the vEPC execute CPU-intensive tasks. Each VNFC might have multiple instances. Each VNFC instance runs on an isolated virtualization container like a Virtual Machine (VM).

Let $m_l^{(c)}$ denote the number of dedicated physical CPU cores allocated to the instance l of the VNFC $c \in C = \{MME, cSGW, cPGW, DPGW\}$. Since the number of CPU cores of a physical server is finite and they are shared among several VMs, we will consider that $m_l^{(c)}$ is limited to m_{max}, *i.e.*, $m_l^{(c)} \leq m_{max}$.

6.2.2 Performance Requirements

The LTE network has to meet a set of performance requirements in terms of latency and packet loss probability [125].

For the CP, the 3GPP defines a bound on the mean CP latency $\overline{T}_{budget}^{(CP)}$ as the performance requirement, *i.e.*, the average elapsed time to move an UE from IDLE state to ACTIVE state [125]. Here, we will translate this specification as the required average time to carry out a Service Request (SR) procedure.

Moreover, we will consider the worst-case scenario for the SR procedure. That is the UE authentication, Non-Access Stratum (NAS) security setup, and the Evolved Packet System (EPS) session modification steps will be carried out during the SR (see Fig. 4.2).

Let $\overline{T}_e$ and $\overline{T}_{if}$ denote respectively the mean response times of the CP entity $e \in E = \{UE, eNB, MME, cSGW, cPGW, HSS, PCRF\}$ and the LTE interface $if \in IF = \{Uu, S1-C, S11, S6a, S5, Gx\}$. The mean time required to carry out an SR, noted as $\overline{T}^{(SR)}$ in the worst-case scenario can be computed as:

$$\begin{aligned}\overline{T}^{(SR)} &= 5 \cdot \overline{T}_{UE} + 8 \cdot \overline{T}_{eNB} + 5 \cdot \overline{T}_{MME} + 2 \cdot \overline{T}_{cSGW} \\ &+ 2 \cdot \overline{T}_{cPGW} + \overline{T}_{HSS} + \overline{T}_{PCRF} + 8 \cdot \overline{T}_{Uu} + 7 \cdot \overline{T}_{S1-C} \\ &+ 2 \cdot \overline{T}_{S11} + 2 \cdot \overline{T}_{S6a} + 2 \cdot \overline{T}_{S5} + 2 \cdot \overline{T}_{Gx}\end{aligned} \tag{6.1}$$

The above equation means that during an SR call flow, in the worst case scenario, the UE, eNB, MME, cSGW, cPGW, HSS, and PCRF entities have to process respectively 5, 8, 5, 2, 2, 1, and 1 control messages. And 8, 7, 2, 2, 2, and 2 control messages have to traverse respectively the LTE Uu, S1-C, S11, S6a, S5, and Gx interfaces [54]. Then, the CP delay requirement can be expressed as $\overline{T}^{(SR)} \leq \overline{T}^{(CP)}_{budget}$.

For the DP, the performance requirements considered are the maximum DP delay budget $T^{(DP)}_{budget}$ and the packet loss probability at the vEPC $P^{(EPC)}_{budget}$. We consider the $T^{(DP)}_{budget}$ as the maximum time it takes for a packet to travel from the SGi interface at the SGW/PGW VNFC to the UE application. The $P^{(EPC)}_{budget}$ is the maximum allowable packet loss at the DPGW VNFC receive buffer.

Let $T^{(DP)}_{max}$ and $P^{(EPC)}$ respectively denote the actual maximum delay of the DP and the packet lost probability of the EPC. We can compute $T^{(DP)}_{max}$ as:

$$T^{(DP)}_{max} = T^{(max)}_{UE} + T^{(max)}_{eNB} + T^{(max)}_{DPGW} + T^{(max)}_{Uu} + T^{(max)}_{S1-U} \tag{6.2}$$

where $T^{(max)}_{UE}$, $T^{(max)}_{eNB}$, and $T^{(max)}_{DPGW}$ are respectively the actual maximum DP packet processing delay at UE, eNB, and DPGW. And $T^{(max)}_{Uu}$ and $T^{(max)}_{S1-U}$ are respectively the actual maximum delays for the DP radio and backhaul interfaces.

Then, the DP requirements can be expressed as $T^{(DP)}_{max} \leq T^{(DP)}_{budget}$ and $P^{(EPC)} \leq P^{(EPC)}_{budget}$.

6.3 Problem Formulation

In this section, we formulate the joint optimization problem to distribute the aggregated workload generated by the UEs of the E-UTRAN among the candidates ECs and to perform the dimensioning of the required resources for each vEPC instance. Our optimization objectives are three-fold:

i) to distribute the workload as equally as possible among the candidates ECs;

ii) to minimize the propagation delays or, equivalently, to use the closest instance of the vEPC to serve the UEs; and

iii) to minimize the required computational resources.

Taking these objectives into account and based on the defined system model, the resources dimensioning and allocation of the vEPCs can be formulated as follows:

Objectives :

$$minimize \left(\sum_{k=1}^{K} \left(\sum_{i=1}^{I} \sum_{j=1}^{J} v_{ik} u_{ij} - \frac{J}{K} \right) \right) \tag{6.3}$$

$$minimize \left(\sum_{k=1}^{K} \sum_{i=1}^{I} v_{ik} \cdot d_{ik} \right) \tag{6.4}$$

$$minimize \left(\sum_{k=1}^{K} \sum_{C} \sum_{l} m_l^{(C)} \right) \quad m_l^{(C)} \in \mathbb{N} \tag{6.5}$$

Where $d_{ik} = ||r_i^{(eNB)} - r_k^{(EC)}||$ is the euclidean distance between the eNB i and the EC k.

Constraints :

CP :

$$C1: \quad \overline{T}_k^{(SR)} \leq \overline{T}_{budget}^{(CP)} \tag{6.6}$$

DP :

$$C2: \quad max \left(T^{(DP)} \right) \leq \overline{T}_{budget}^{(DP)} \tag{6.7}$$

$$C3: \quad P^{(EPC)} \leq P_{budget}^{(EPC)} \tag{6.8}$$

Others

$$C4: \quad m_l^C \leq m_{max} \quad \forall \quad k \in [1, K] \cap \mathbb{N} \tag{6.9}$$

$$C5: \quad \sum_{k=1}^{K} \sum_{i=1}^{I} v_{ik} = I, \quad v_{ik} \in \{0, 1\} \tag{6.10}$$

Constraints 1, 2, and 3 (expressions (6.6), (6.7) and, (6.8)) guarantee that the QoS requirements are fulfilled. Specifically, Constraint 1 assures that the actual mean delay to carry out a SR for the vEPC k (*i.e.*, vEPC instance running on EC k) is lower or equal than the mean CP latency $\overline{T}_{budget}^{(CP)}$. Constraints 2 and 3 respectively ensure that the maximum DP delay budget and the packet loss probability at the EPC are met. Constraint 4 limits the maximum number of physical cores requested for a single VNFC instance. That is because the Physical Machines (PMs) have a maximum number of physical cores. Consequently, the number of physical cores we can request per VNFC instance is limited. Moreover, in general, the higher the number of physical cores requested for a VNFC instance the more difficult it may be to find a PM in the substrate infrastructure that can host the instance. Finally, constraint 5 warrants that all of the eNBs are assigned to a candidate EC k (or vEPC instance k).

6.4 LTE CP Modeling

Based on the model detailed in Chapter 5, the CP of the LTE is modeled as an open network of G/G/m queues (see Fig. 6.5), where each queue represents an instance of a VNFC of the vEPC to be dimensioned (*e.g.*, MME, CP functionality of the SGW (cSGW), and CP functionality of the PGW (cPGW)) [111].In Kendall's notation, a G/G/m queue is a queuing node with m servers, arbitrary arrival and service processes, First-Come, First-Served (FCFS) discipline, and infinite capacity and calling population.

Each queue has $m_l^{(c)}$ servers which represent different CPU instances processing messages in parallel. As stated in Section 6.2.1, $m_l^{(c)} \leq m_{max}$. The rest of the LTE CP entities are modeled as infinite servers, *i.e.*, its mean response time is constant and independent of its workload. It is assumed that those entities have enough capacity to withstand comfortably their workloads.

The traffic sources are located at the eNB and the UE, since the consid-

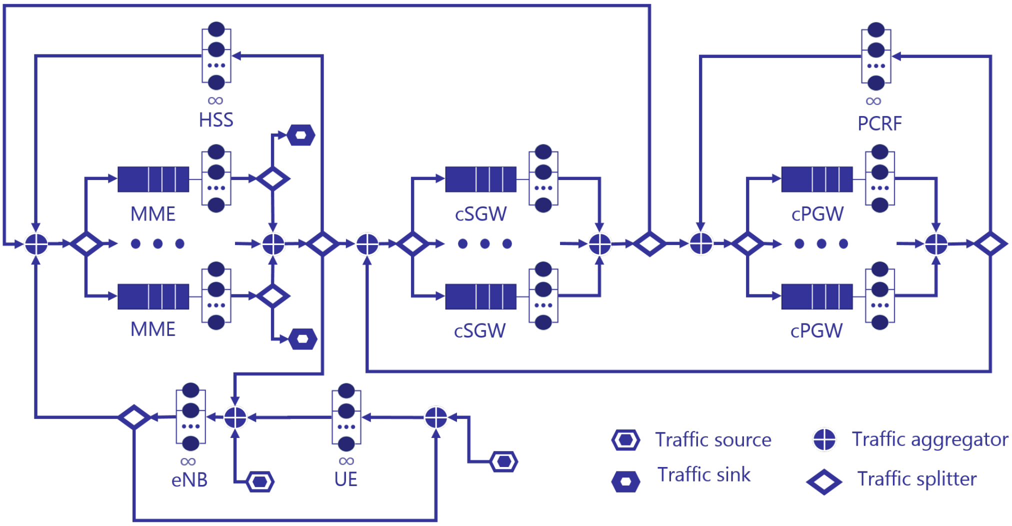


Figure 6.5: LTE CP queuing model.

ered LTE signaling procedures (*e.g.*, SR, S1-Release (S1R), Handover (HO), and Tracking Area Update (TAU)) are triggered by these entities. Specifically, the TAU and SR procedures are triggered by the UE, and the S1R and HO procedures are triggered by the eNB. For similar reasons, the traffic sinks are placed at the MME instances.

To solve the network of queues, we will employ the Queuing Network Analyzer (QNA) method [156]. This technique was applied and validated in Chapter 5 to estimate the mean response time of a VNF with several VNFCs [111]. Here, we will use QNA method to estimate the mean response times of the VNFCs of the vEPC CP to be dimensioned, *i.e.*, $\overline{T}_{MME}$, $\overline{T}_{cSGW}$, and $\overline{T}_{cPGW}$. These response times can be estimated using (5.12)-(5.22) and the following set of input parameters:

- The mean arrival rate λ_{0k} and the squared coefficient variation (SCV) c^2_{0k} of the external arrival processes at node k. Note that only the UE and the eNB have external arrival processes in the model of the LTE CP (see Fig. 6.5). Considering the abstract model described in Section 3.3 for the signaling generation process, we found that the aggregated signaling arrival processes are Poissonian (see Section 3.6). Consequently, $c^2_{0k} = 1, \quad \forall \quad k$.

- The mean service rate μ_k and the SCV of the service times c_{sk}^2 at each queue k, and c_{sk}^2. Section 5.6.2 includes the description of methodologies to measure these input parameters.

- The steady state transition probabilities matrix $P = [p_{ki}]$, where p_{ki} denotes the probability of a packet leaving the node k is next moved to the node i or leaves the network with probability $p_{0k} = 1 - \sum_i p_{ki}$. In the following subsection, we will derive analytical expressions to compute P for the queuing model of the LTE CP.

6.4.1 Transition Probabilities for the LTE CP Queuing Model

Let V_e denote the visit ratio of the CP entity $e \in E = \{UE, eNB, MME, cSGW, cPGW, HSS, PCRF\}$ which is defined as the average number of visits to entity e by a signaling procedure during its lifetime in the network. That is,

$$V_e = \frac{\lambda_e}{\sum_e \lambda_{0e}} = \frac{\lambda_e}{(\lambda_{0UE} + \lambda_{0eNB})} \tag{6.11}$$

Equivalently, V_e is equal to the average number of packets to be processed by the entity e per control procedure. Let $n_{sp}^{(e)}$ and λ_{sp} respectively be the number of packets to be processed by the entity e during the signaling procedure $sp \in SP = \{SR, S1R, HO, TAU\}$ and the average generation rate of the control procedure $sp \in SP$ in the network. Then,

$$V_e = \frac{\sum_{sp \in SP} \lambda_{sp} \cdot n_{sp}^{(sp)}}{\sum_{sp \in SP} \lambda_{sp}} \tag{6.12}$$

The visit ratios and the transition probabilities between entities are related through (5.12) (flow balance equations) as:

$$V_e = \frac{\lambda_{0e}}{\sum_{e \in E} \lambda_{0e}} + \sum_{es \in E} V_{es} \cdot p_e^{es} \tag{6.13}$$

where p_e^{es} denotes the transition probability from entity $es \in E$ to entity $e \in E$. Additionally, the sum of the transition probabilities for a given entity es are normalized to unity:

$$p_{0es} + \sum_{e \in E} p_e^{es} = 1 \tag{6.14}$$

Finally, assuming that the workload is distributed among the instances of the MME, cSGW, and cPGW according to their computational capacity, *i.e.*, $V_{e_l} = m_l^{(e)} / (\sum_l m_l^{(e)}) \cdot V_e$, and using (6.13) and (6.14), we can compute the transition probabilities for our LTE CP queuing model. They are given by the following expressions:

$$p_{UE}^{eNB} = \frac{V_{UE} - \frac{\lambda_0^{(UE)}}{\sum_{e \in E} \lambda_0^{(e)}}}{V_{eNB}} \tag{6.15}$$

$$p_{MME_l}^{eNB} = \frac{m_l^{(MME)}}{\sum_l m_l^{(MME)}} \cdot (1 - p_{UE}^{eNB}) \tag{6.16}$$

$$p_{eNB}^{MME_l} = \frac{V_{eNB} - \frac{\lambda_0^{(eNB)}}{\sum_{e \in E} \lambda_0^{(e)}} - V_{UE}}{V_{MME}} \tag{6.17}$$

$$p_{cSGW_l}^{MME_l} = \frac{m_l^{(cSGW)}}{\sum_l m_n^{(cSGW)}} \cdot \left(1 - p_{eNB}^{MME_l} - p_{eNB}^{MME_l} - \frac{1}{V_{MME}}\right) \tag{6.18}$$

$$p_{HSS}^{MME_l} = \frac{V_{HSS}}{V_{MME}} \tag{6.19}$$

$$p_{MME_l}^{cSGW_l} = \frac{m_l^{(MME)}}{\sum_m m_m^{(MME)}} \cdot \left(1 - \sum_l p_{cPGW_l}^{cSGW_l}\right) \tag{6.20}$$

$$p_{cPGW_l}^{cSGW_l} = \frac{m_l^{(cPGW)}}{\sum_m m_m^{(cPGW)}} \cdot \frac{(V_{PGW} - V_{PCRF})}{V_{SGW}} \tag{6.21}$$

$$p_{cSGW_l}^{cPGW_l} = \frac{m_l^{(cSGW)}}{\sum_m m_m^{(cSGW)}} \cdot \left(1 - \frac{V_{PCRF}}{V_{PGW}}\right) \tag{6.22}$$

$$p_{PCRF}^{PGW_l} = \frac{V_{PCRF}}{V_{PGW}} \tag{6.23}$$

$$p_{MME_l}^{HSS} = \frac{m_l^{(MME)}}{\sum_m m_m} \tag{6.24}$$

$$p_{cPGW_l}^{PCRF} = \frac{m_l^{(cPGW)}}{\sum_n m_n^{(cPGW)}} \tag{6.25}$$

Note that the transition probabilities depend on the average number of packets to be processed for each LTE CP entity per control procedure, which is equal to the visit ratio of the entity; the external arrival processes λ_{0UE} and λ_{0eNB}; and depend on the number of processing instances assigned to each instance l of the entity e, $m_l^{(e)}$.

6.5 LTE DP Modeling

For the considered architecture, the LTE DP consists of three network entities (*e.g.*, UE, eNB, and Data Plane Gateway (DPGW)), which are connected in tandem. Since we are focusing on the vEPC dimensioning, we will assume that the UE and eNB entities have constant maximum delays.

For the DPGW, we will model it as a single queue fed by a fBm process. This choice was motivated by the results described in Section 3.6.3 for the DP aggregated workload characterization.

To characterize the arrival process we adopt the model proposed in [105]. Let A_t denote the cumulating arrival process to the DPGW queue, *i.e.*, the cumulative amount of traffic (in number of packets) arriving at the DPGW in the time interval $[0, t)$. The following model is considered for A_t [105]:

$$A_t = \lambda \cdot t + \sqrt{\lambda \cdot \alpha} \cdot Z_t \tag{6.26}$$

where Z_t is a normalized fBm parameter with Hurst parameter $H \in [1/2, 1)$, $\lambda > 0$ is the mean input rate, and $\alpha > 0$ is a variance coefficient.

Under the above packet arrival model and considering a constant rate server with capacity C, the violation probability $\epsilon = P[B > b]$ of a backlog bound b can be approximated as [105], [136]:

$$\epsilon \approx exp\left(-\frac{(C-\lambda)^{2H}}{2 \cdot \kappa(H)^2 \cdot \lambda \cdot \alpha} b^{2-2H}\right) \tag{6.27}$$

where $\kappa(H) = H^H(1-H)^{1-H}$. The above equation give us an approximation for the probability of saturation of a buffer of size b packets or equivalently the packet loss probability at a queue fed with a fBm arrival process.

Finally, the maximum response time of a queuing node with buffer size b and constant rate server with capacity C can be computed as:

$$T^{(max)} = \frac{b+1}{C} \tag{6.28}$$

By using equations (6.27) and (6.28), we can perform the dimensioning of the required capacity of the DPGW.

6.6 PES: Planner for the EPC as a Service

This section includes a heuristic method, which is dubbed "Planner for the EPC as a Service" (PES), to find a sub-optimal solution of the problem formulated in Section 6.3. To achieve a method with low-complexity, the workload distribution among the candidates ECs and the resources dimensioning of the vEPC at each EC stages are decoupled. The whole heuristic method is shown in Algorithm 2, which proceeds as following.

Algorithm 2 PES

```
Input: eNBs positions $r_i^{(eNB)}$ along with the number of UEs they serve
    $\mathbf{N}_{eNB}^{UE}(i) = \sum_j u_{ji}$, and the QoS specs $T_{budget}^{(DP)}$, $P_{budget}^{(EPC)}$, and $\overline{T}_{budget}^{(CP)}$.
Output: eNBs assigment (i.e., $v_{ik}$), and total number of processing instances
    allocated to each vEPC entity per EC (e.g., $\mathbf{m}_{MME}$, $\mathbf{m}_{cSGW}$, $\mathbf{m}_{cPGW}$, and
    $\mathbf{m}_{DPGW}$).
1: $[\mathbf{N}_{EC}^{UE}, v_{ik}] \Leftarrow Partitioning(\mathbf{r}^{(eNB)}, \mathbf{N}_{eNB}^{UE})$
2: for each $k \in K$ do
3:     Compute the processing delay budgets for the vEPC CP and DP,
       $T_{proc-budget}^{(CP)}$ and $T_{proc-budget}^{(DP)}$, by using (6.29) and (6.30).
4:     For $N_U = \mathbf{N}_{EC}^{UE}(k)$, estimate the external arrival processes ($\lambda^{(CP)}$, $\lambda^{(DP)}$,
       $\alpha^{(DP)}$, and $H^{(DP)}$) by using (3.16), (3.17), (3.18), (3.19), (3.20), (3.21), and
       (3.22).
5:     $[\mathbf{m}_{MME}(k), \mathbf{m}_{cSGW}(k), \mathbf{m}_{cPGW}(k), \mathbf{m}_{DPGW}(k)] \Leftarrow Dimensioning(\lambda^{(CP)},$
       $\lambda^{(DP)}, \alpha^{(DP)}, H^{(DP)}, T_{proc-budget}^{(CP)}, T_{proc-budget}^{(DP)}, P_{budget}^{(EPC)})$
6: end for
```

First, the partitioning algorithm assigns each eNB to a candidate EC (line 1 of the Algorithm 2). The workload partitioning process is detailed in the Algorithm

3. The idea of the Algorithm 3 is to distribute the workload as equally as possible among the candidates ECs, while guaranteeing a maximum propagation delay for the BN $t^{(max)}_{prop-backhaul}$. To that end, the algorithm initializes the workload assigned to each EC k $\mathbf{N}^{UE}_{EC}(k)$, which is measured as the number of assigned UEs, to zero. Then, it iteratively finds the candidate EC k^* with the lowest workload allocated and its nearest eNB i^* not assigned yet. If the propagation delay limit between the EC k^* and the eNB i^* is not violated, then, the eNB i^* is attached to the EC k^* ($v_{i^*k^*} = 1$). Otherwise, the EC k^* is excluded from the set of candidates ECs K. The algorithm ends when all the eNBs are allocated. It should be noted that the algorithm requires $N_{eNB} + N_{EC}$ iterations to assign all the eNBs in the worst case.

Algorithm 3 E-UTRAN Partitioning Algorithm

```
Require: All eNBs of the set $I$ have to be assigned to an EC of the set $K$.
Input: eNBs positions $r^{(eNB)}_i$ along with the number of UEs they serve
    $\mathbf{N}^{UE}_{eNB}(i) = \sum_j u_{ji}$, the ECs positions $r^{(EC)}_k$, and the maximum propagation
    time for the backhaul network $t^{(max)}_{prop-backhaul}$.
Output: eNBs assigment, i.e., $v_{ik}$
Initialization $\mathbf{N}^{UE}_{EC} = \overrightarrow{0}$, $v_{ik} = 0$
while $I \neq \emptyset$ do
    $k^* = \arg\min_{k \in K}(\mathbf{N}^{UE}_{EC}(k))$
    $i^* = \arg\min_{i \in I} ||\mathbf{r}^{(EC)}_{k^*} - \mathbf{r}^{(eNB)}_i||$
    if $||\mathbf{r}^{(EC)}_{k^*} - \mathbf{r}^{(eNB)}_{i^*}|| \leq t^{(max)}_{prop-backhaul} \cdot c$ then
        $I \Leftarrow I \backslash i^*$, $v_{i^*k^*} = 1$
        $\mathbf{N}^{UE}_{EC}(k^*) \Leftarrow \mathbf{N}^{UE}_{EC}(k^*) + \mathbf{N}^{UE}_{eNB}(i^*)$
    else
        $K \Leftarrow K \backslash k^*$
    end if
end while
```

Once the eNBs assigment finish, the processing time budgets for the vEPC CP $T^{(CP)}_{proc-budget}$ and DP $T^{(DP)}_{proc-budget}$ can be computed (line 3 of the Algorithm 2). To that end, we can evaluate $\overline{T}^{(SR)}$ and $T^{(DP)}_{max}$ in (6.1) and (6.2), respectively, for $\overline{T}_{MME}$, $\overline{T}_{cSGW}$, $\overline{T}_{cPGW}$, and $T^{(max)}_{DPGW}$ equal to zero. That is, $\overline{T}^{(SR)}_0 = \overline{T}^{(SR)}(\overline{T}_{MME} = 0, \overline{T}_{cSGW} = 0, \overline{T}_{cPGW} = 0)$ and $T^{(DP)}_{max0} = T^{(DP)}_{max}(T^{(max)}_{DPGW} = 0)$.

Then,

$$T^{(CP)}_{proc-budget} = \overline{T}^{(CP)}_{budget} - \overline{T}^{(SR)}_{0} \tag{6.29}$$

$$T^{(DP)}_{proc-budget} = T^{(DP)}_{bugdet} - T^{(DP)}_{max0} \tag{6.30}$$

Now, once there is an estimation of the number of UEs to be served by each EC, we can also estimate the aggregated external arrival processes for both the LTE CP and DP, which are inputs to the resources dimensioning algorithm. Here, we will use the stochastic characterization of the aggregated LTE data traffic and signaling workload generation processes included in Section 3.6.3, where curve fittings are provided to estimate the main parameters to model them as a function of the number of users.

Algorithm 4 Dimensioning Algorithm

```
Input: Processing delay budgets for the vEPC CP $T^{(CP)}_{proc-budget}$ and DP
    $T^{(DP)}_{proc-budget}$; $P^{(EPC)}_{budget}$; External arrival processes characterization for CP and
    DP ($\lambda^{(CP)}$, $\lambda^{(DP)}$, $\alpha^{(DP)}$, and $H^{(DP)}$).
Output: number of physical cores allocated to each vEPC entity $m_{MME}$,
    $m_{cSGW}$, $m_{cPGW}$, and $m_{DPGW}$
{DATA PLANE:}
Solve (6.27) numerically for $b \leq T^{(DP)}_{proc-budget} \cdot C - 1$ and $\epsilon \approx P^{(EPC)}_{budget}$ to obtain
    the required DP processing capacity $C$. Then, $m_{DPGW} = \lceil C/\mu_{DPGW} \rceil$.
{CONTROL PLANE:}
Initialization $m_{MME} = \lceil \lambda_{MME}/\mu_{MME} \rceil$, $m_{cSGW} = \lceil \lambda_{cSGW}/\mu_{cSGW} \rceil$,
    $m_{cPGW} = \lceil \lambda_{cPGW}/\mu_{cPGW} \rceil$, $M_{CP} = m_{MME} + m_{cSGW} + m_{cPGW}$, $T^{(CP)}_{proc} =$
    $8 \cdot T_{MME}(m_{MME}) + 3 \cdot T_{cSGW}(m_{cSGW}) + 2 \cdot T_{cPGW}(m_{cPGW})$;
while $T^{(CP)}_{proc} > T^{(CP)}_{proc-budget}$ do
    $M_{CP} \Leftarrow M_{CP} + 1$
    for each $m \in \{m_{MME}, ..., M_{CP} - m_{cSGW} - m_{cPGW}\} \cap \mathbb{N}$ do
        for each $n \in \{m_{cSGW}, ..., M_{CP} - m_{MME} - m_{cPGW}\} \cap \mathbb{N}$ do
            $l = M_{CP} - m - n$
            $T_{aux} = 8 \cdot T_{MME}(m) + 3 \cdot T_{cSGW}(n) + 2 \cdot T_{cPGW}(l)$
            if $T^{(CP)}_{proc} > T_{aux}$ then
                $T^{(CP)}_{proc} \Leftarrow T_{aux}$, $m_{MME} \Leftarrow m$, $m_{cSGW} \Leftarrow n$, $m_{cPGW} \Leftarrow l$
            end if
        end for
    end for
end while
```

Finally, the resources dimensioning is carried out (see Algorithm 4). The dimensioning of the vEPC CP and DP is performed separately. Since we are considering only one VNFC for the vEPC DP, its dimensioning simply requires solving numerically (6.27). For the CP, the algorithm searches for the minimum number of processing instances to be allocated to the vEPC CP for a given EC so that a processing delay budget $T^{(CP)}_{proc-budget}$ be met. The algorithm iterates until the processing delay budget is fulfilled. At each iteration it increments by one the number of processing instances M_{CP} allocated to the vEPC CP. For a given M_{CP}, the algorithm explores different combinations to distribute these instances among the different VNFCs to be dimensioned (*e.g.*, MME, cSGW, cPGW), and chooses that one providing the lowest processing delay.

To achieve linear complexity, the search space is limited at each iteration (see line 12 of Algorithm 4). In the algorithm, $T_{mme}(m)$, $T_{cSGW}(n)$, and $T_{cPGW}(l)$ respectively denote the mean response times of the MME, cSGW, and cPGW for a given number of allocated processing instances m, n, and l. These mean response times are estimated by using the QNA method, as described in Chapter 5.

Note that although it is not explicitly included in Algorithm 4, for each processing instances allocation (m, n, l) it is necessary to re-estimate the internal flow parameters at each queue using (5.12)-(5.18), and the transition probability matrix using (6.15)-(6.25).

The number of instances or, equivalently, the number of virtualization containers for each vEPC entity at a given EC can be simply computed as: $\lceil m_{MME}/m_{max} \rceil$, $\lceil m_{cSGW}/m_{max} \rceil$, $\lceil m_{cPGW}/m_{max} \rceil$, and $\lceil m_{DPGW}/m_{max} \rceil$.

6.7 Experimental Setup

To validate both the models developed in this chapter and the operation of PES as well, a system-level simulator of an LTE network were used. The simulator was developed within the ns-3 environment [63].

It simulates the processing of the different LTE logical entities and the messages exchange between them at both the DP and the CP. Several candidate ECs sites are located randomly in the geographical area defined in the simulation scenario. An instance of the vEPC is deployed on every EC. The number

Table 6.1: Parameters Configuration of the LTE simulator

eNB setup		**Spatial Distribution of the UEs**	
Coverage area	$37.5 \times 37.5\ km^2$	Pixel size	$375 \times 375\ m^2$
Maximum Tx power	$20\ W$	ω_{max} rural UL	0.001202
Noise power	$4 \cdot 10^{-21}\ W/Hz$	ω_{max} rural DL	0.001163
Number of antennas	1	ω_{max} urban UL	0.012673
Antenna gain	$10\ dB$	ω_{max} urban DL	0.011592
Carrier frequency	$2.3\ GHz$	Location μ rural UL	11.573
Bandwidth	$20\ Mz$	Location μ rural DL	12.572
Noise figure	$8\ dB$	Location μ urban UL	17.7956
Std of log-normal shadowing	$8\ dB$	Location μ urban DL	18.93
Spectral efficiency	$10\ bits/Hz$	Scale σ rural UL	2.3055
Minimum SNR requirement	$3.5\ dB$	Scale σ rural DL	2.7985
Inactivity timer value	$10\ s$	Scale σ urban UL	2.1188
CP Service Processes		Scale σ urban DL	2.3991
μ_{MME}, μ_{cSGW}, and μ_{cPGW}	$6700\ packets/second$	**DP Service Processes**	
c^2_{sMME}, c^2_{cSGW}, and c^2_{scPGW}	0.65	μ_{DPGW}	$1813236\ packets/second$
$\overline{T}_{UE}$, $\overline{T}_{eNB}$, $\overline{T}_{HSS}$, and $\overline{T}_{PCRF}$	$1\ ms$	$T^{(max)}_{UE}$	$100\ \mu s$
$\overline{T}_{S6a}$ and $\overline{T}_{Gx}$	$1.5\ ms$	$T^{(max)}_{eNB}$	$200\ \mu s$
$\overline{T}_{S11}$ and $\overline{T}_{S5}$	$30\ \mu s$	**QoS Requirements**	
Propagation delays		$\overline{T}^{(CP)}_{budget}$	$25\ ms$
Speed of light in air	$3 \cdot 10^8\ m/s$	$T^{(DP)}_{budget}$	$1\ ms$
Speed of light in fiber	$2 \cdot 10^8\ m/s$	$P^{(EPC)}_{budget}$	10^{-6}

of instances per VNFC of the vEPC or network entities are computed based on PES algorithm described in the previous section.

The following subsections include implementation details of the simulator. Table 6.1 includes the configuration of the main simulation parameters which were considered to obtain the reported results.

6.7.1 Service Consumption and Spatial Distribution of the Users

Each user generates or consumes data and signaling traffic according to the abstract model described in Section 3.3 and the compound traffic model included in Table 3.6.

To distribute the users through the coverage area of the E-UTRAN, the simulator uses the spatial model of the traffic density in cellular networks proposed in [174]. According to that model, the spatial distribution of the traffic density can be approximated by the log-normal distribution. The parameters of the distributions changes depending on the type of zone. Here, we will consider two types of zones: rural and urban. The procedure to distribute the users through the coverage area comprises the following steps:

- The coverage area is divided into $M \times N$ square pixels. The size of these pixels should be the same as the one considered in the measurement setup (*i.e.*, the spatial resolution considered to adjust the log-normal distribution from the measurements).

- A Gaussian random field $\boldsymbol{\rho}^{(G)}$ is generated as:

$$\rho_{m,n}^{(G)} = \frac{2}{\sqrt{L}} \cdot \sum_{l=1}^{L} cos(i_l x_{m,n} + \phi_l) \cdot cos(j_l y_{m,n} + \Psi_l) \tag{6.31}$$

  where $x_{m,n}$ and $y_{m,n}$ are the coordinates of the center of the pixel (m, n), i_l and j_l are the angular frequencies which are uniformly distributed between 0 and ω_{max}, and ϕ_l and Ψ_l are the phases which are uniformly distributed between 0 and 2π.

  The higher the number of components L considered, the better $\rho_{m,n}^{(G)}$ approximate standard Gaussian random variables according to the central limit theorem [174]. The parameter ω_{max} is the spatial spread which decides the rate of fluctuations of the random field [174].

- The traffic density map whose elements are log-normally distributed is generated as follows:

$$\rho_{m,n} = exp(\sigma \rho_{m,n}^{(G)} + \mu) \qquad (bytes/km^2) \tag{6.32}$$

  where μ and σ are respectively the location and scaling parameters. By controlling these two parameters, the values of $\rho_{m,n}$ are adjusted to fit the statistics of traffic density for specific regions (*e.g.*, an urban or a rural area).

- Last, the number of users per pixel is simply computed as the traffic intensity $\rho_{m,n}$ at pixel (m, n) multiplied by the pixel area and divided by the average data traffic rate generated per user which is given by the compound traffic model used. The users positions within a pixel are distributed uniformly.

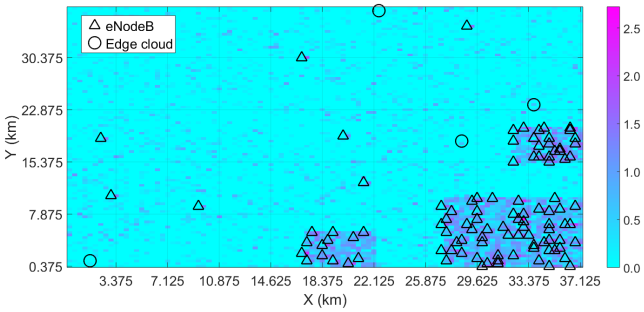


Figure 6.6: E-UTRAN deployment for a population density of 500 users per km^2.

6.7.2 E-UTRAN Deployment

To generate RAN deployment (*i.e.*, the locations of the eNBs in the coverage area), the simulator uses the heuristic proposed in [175]. The algorithm tries to minimize the number of eNBs deployed, while guaranteeing a minimum Signal-to-Noise Ratio (SNR) required per UE and while ensuring that the capacity of every eNB is greater than the capacity demand in its coverage area.

The capacity demand in the coverage area of an eNB is given by the average traffic demand per user and the density of users. The density of users, in turn, is given by the spatial distribution of the users (described in the previous subsection).

Figures 6.6 and 6.7 show two realizations of the E-UTRAN deployment for population densities of 500 UEs/km^2 and 1000 UEs/km^2, respectively. As can be seen, the scenario consists of three urban zones where most of the population is concentrated. Additionally, four candidates ECs are considered, whose positions were randomly generated. The same scenario were used to generate the simulation results described in this chapter.

6.7.3 Network Entities Instances Simulation

Each instance of the MME, cSGW, cPGW, and DPGW entities is simulated as a FCFS queue with one or more servers. The service time distribution considered for the CP entities (*e.g.*, MME, cSGW, and cPGW) is that one depicted in Fig.

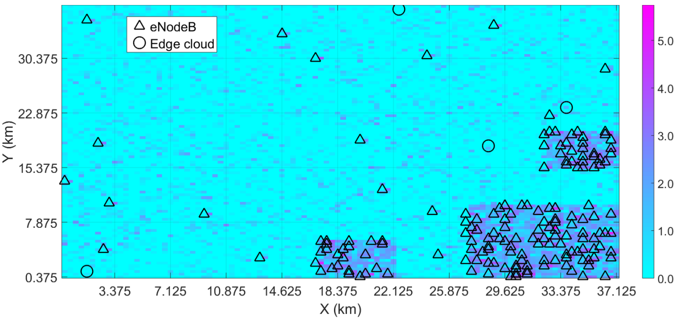


Figure 6.7: E-UTRAN deployment for a population density of 1000 users per km^2.

5.11 and described in Section 5.8.1 for the Worker (W) instance of the three-tiered vMME. The service process of the DPGW is deterministic, and its service rate was set to that one measured in [106] for a virtualized SGW/PGW.

The instances of the rest entities (*e.g.*, UE, eNB, HSS, and PCRF) and the delays of the interfaces (*e.g.*, transmission, propagation, and the processing up to transport layer) between any couple of entities instances are simulated as infinity servers, *i.e.*, constant processing delay without queuing waiting time.

6.8 PES Validation and Runtime

To carry out the performance evaluation of PES, the scenario shown in Figs. 6.6 and 6.7 was used. The considered metrics in the assessment were the algorithm runtime, the dimensioning of the network and computational resources, and the network QoS metrics defined in Section 6.2.2. These metrics were measured for different population densities. Additionally, we compared PES with two baseline approaches for the workload partitioning:

- Workload partitioning based on proximity. That is, each eNB is assigned to the closest EC (Voronoi diagram). This approach is labeled as "Voronoi" in the reported figures.
- A fully centralized approach. That is, all the eNBs are assigned to the same EC. The chosen EC is the nearest to the largest concentrations of users. This approach is labeled as "Centralized" in the figures.

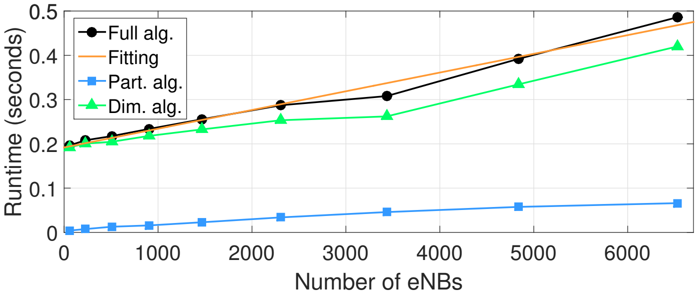


Figure 6.8: PES execution time.

6.8.1 PES Runtime

Figure 6.8 shows the runtime of PES versus the number of eNBs deployed at each population density studied. Each measurement was repeated five times and averaged it.

The dimensioning algorithm (labeled as “Dim. alg.”) and, more specifically, the CP dimensioning is the heaviest part of the full algorithm. The CP dimensioning algorithm achieves linear complexity because of the search space is limited at each iteration. Otherwise, the number of checks the algorithm would have to carry out at each iteration is 3 (number of entities to be dimensioned) multichoose $M_{CP} - M_{CP0}$. That is,

$$\left(\!\!\binom{3}{M_{CP} - M_{CP0}}\!\!\right) = \frac{(2 + M_{CP} - M_{CP0})!}{(M_{CP} - M_{CP0})! \cdot 2}. \tag{6.33}$$

where M_{CP} and M_{CP0} are respectively the number of processing instances allocated to the vEPC CP for a given iteration and the number of processing instances allocated in the initial assignment to fulfill the stability condition, *i.e.*, $M_{CP0} = \lceil \lambda_{MME}/\mu_{MME} \rceil + \lceil \lambda_{cSGW}/\mu_{cSGW} \rceil + \lceil \lambda_{cPGW}/\mu_{cPGW} \rceil$.

The partitioning algorithm (labeled as “Part. alg.”) takes linear time, as it was expected, since it requires $I + K$ iterations to assign all the eNBs in the worst case.

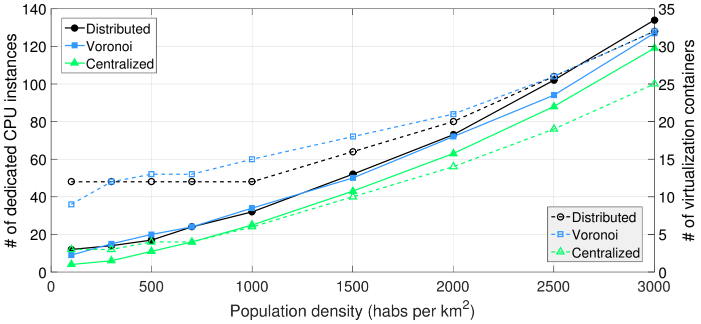


Figure 6.9: Total number of dedicated CPU instances and virtualization containers.

6.8.2 PES Validation

Figures 6.9 and 6.10 depict respectively the number of CPU instances required for the CP and the DP versus the population density. The computational resources were estimated by PES considering the aforementioned approaches for the workload partitioning among the candidates ECs to meet the performance requirements.

The CP has a higher demand of computational resources than DP in the considered scenario, though the throughput demand is three orders of magnitude higher for DP (see Figs. 6.11 and 6.12). This is because of the processing of the control messages is heavier than that one required for a data packet (see Table 6.1). It was observed that PES allocated most of the CP computational resources to MME entity, followed by cSGW. This is due to the fact that the MME has the highest visit ratio ($V_{MME} = 2.4196$, while $V_{cSGW} = 1.3585$ and $V_{cPGW} = 0.7170$).

It is worth mentioning that the computational resources allocated to the CP depend quadratically on the population density (see Fig. 6.9). That is because of the frequency of HOs and TAUs increases as the RAN is denser. As shown in Chapter 3, the rates of HO and TAU procedures are a quadratic function of the number of UEs, see (3.20) and (3.21).

Regarding the workload partitioning approaches, in general, Voronoi approach offers the best performance in terms of delay, see Figs. 6.13 and 6.14. This is in part due to it minimizes the propagation delays. On the other hand, the

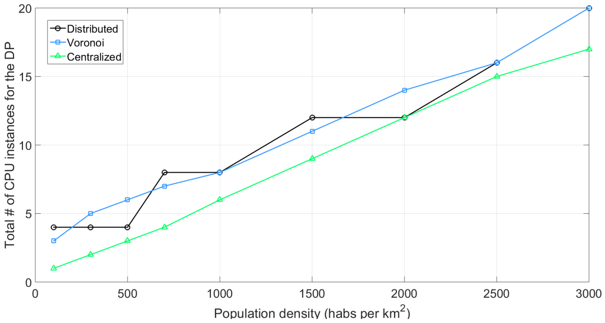


Figure 6.10: Total number of dedicated CPU instances for the DP.

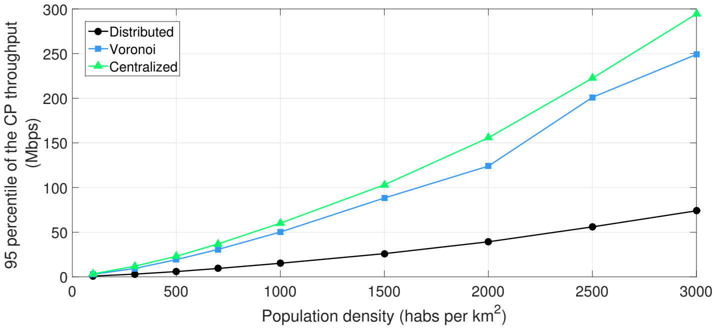


Figure 6.11: 95 percentile of the CP workload for the most loaded EC.

centralized approach leads to a better usage of the computational resources, see Figs. 6.9 and 6.10. Last, the distributed approach (Algorithm 3) minimizes the workload imbalances among the candidates ECs as shown in Figs. 6.11 and 6.12.

Figures 6.13 and 6.14 show the values of the QoS metrics obtained via simulation for the different population densities studied. The target performance metrics ($\overline{T}^{(CP)}_{budget} = 25\ ms$, $T^{(DP)}_{budget} = 1\ ms$, and $P^{(EPC)}_{budget} = 10^{-6}$) are always met, thus validating the proper operation of PES.

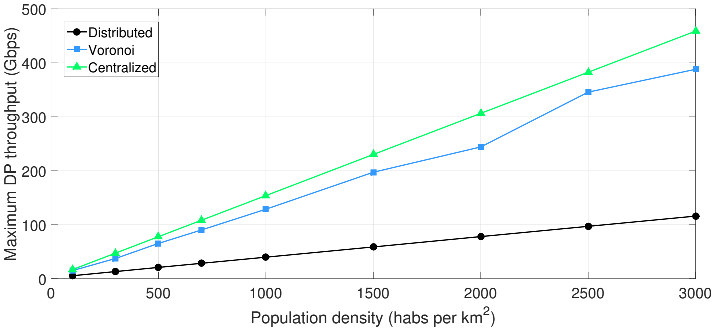


Figure 6.12: Maximum DP workload for the most loaded EC.

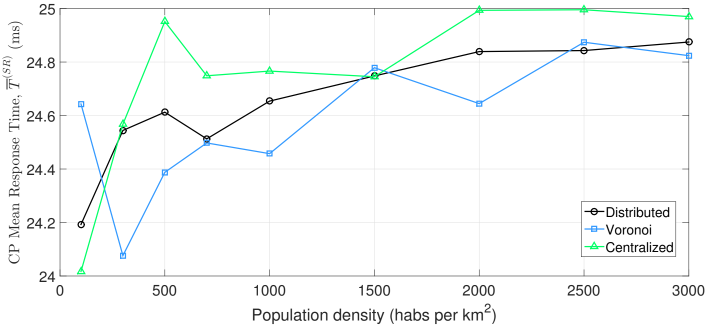


Figure 6.13: Mean response time of the CP.

6.9 Conclusions

The network slices for the different verticals need to be planned before their instantiation. Additionally, although NetSoft paradigm enables operators to dynamically adapt the resources allocated to each network slice and services, the on-demand plans offered by cloud providers are more expensive than reservation plans (up to 70% off the on-demand price). Then, the resource dimensioning during the network slices planning is important for operators to save money. Two major problems to be addressed during the planning of the network slices are the resources dimensioning and allocation.

On the one hand, the resources dimensioning refers to estimate the required computational, network, and virtual resources so that a given set of performance targets are achieved. On the other hand, the resource allocation deals with the

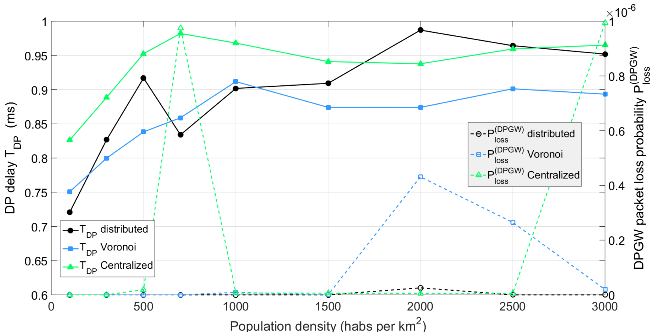


Figure 6.14: QoS of the DP.

choice of the substrate infrastructure, the PMs, and paths within the infrastructure where the network slice will be embedded. Moreover, the resources dimensioning and allocation processes have to be done in a manner that they ensure the QoS requirements for the target use case.

In this chapter, an integral solution for planning the LTE vEPC, which is tailored for the eMBB services group, has been described. The solution, which is dubbed ”Planner of the EPC as a Service” (PES), performs the resources dimensioning and allocation among several candidates ECs. One of the goals of PES is to distribute the workload or the resource allocation among the candidates ECs as equally as possible, thus minimizing workload imbalances. Additionally, PES also aims at maximizing the resources utilization and minimizing the latencies, while a set of performance requirements are guaranteed. The performance metrics considered by PES closely follow the 3GPP LTE specifications.

The dimensioning algorithms of PES are based on analytic performance models for both the CP and the DP. More precisely, the LTE CP has been modeled following the same process as that one detailed in Chapter 5, whereas the vEPC DP, leveraging on the workload characterization results showed in Chapter 3, has been modeled as a queue fed by a fBm process.

Finally, the proper operation of PES has been validated by means of simulations. We have also carried out a comparison between different approaches to distribute the workload among a set of candidates ECs. More precisely, the results show that PES distributes the workload equally among the candidates ECs, while guaranteeing the performance requirements of the vEPC. Compared to other approaches such as a Voronoi diagram to carry out the workload partitioning among a set of candidates ECs, PES reduces the workload imbalances among the candidates ECs. This leads to improvements in terms of request acceptance and resource utilization [173]. Last, the results show that the PES time complexity exhibits linear dependence with the population density.

Chapter 7

Dynamic Resource Provisioning of Softwarized Networks

The Network Softwarization (NetSoft) paradigm facilitates the automation of the management operations and orchestration of the future networks [41]. The envisioned management practices include the automation of the scaling of network services which is commonly known as Dynamic Resource Provisioning (DRP). More precisely, DRP allows the system to adapt its computational resources autonomously depending on the current workload so that some performance requirements are met. This approach enables operators to handle workload fluctuations to keep the desired performance with great agility and reduced costs.

Nevertheless, the procedure of spinning-up or freeing virtual resources introduces a non-negligible delay [176]. Therefore, waiting until the system is overloaded or underutilized, so as to scale resources up or down, could negatively impact the user Quality of Experience (QoE), or lead to inefficient resources utilization. In this regard, the use analytical models, as that one described in Chapter 5, for predicting the performance of softwarized networks are an appropriate and agile solution to this problem.

Based on the performance model described in Chapter 5, this chapter explains an algorithm for performing DRP of softwarized networks. This kind of algorithms are also called Dynamic Auto Scaling Algorithms (DASAs) in the literature [137]. The rest of the chapter is organized as follows. Section 7.1 briefly reviews the related literature. Section 7.2 describes the proposed DRP algorithm. Section 7.3 introduces the Network Slicing Orchestration System (NSOS) defined in 5G!Pagoda project and addresses its performance modeling. The dynamic provisioning of this NSOS will be one of the study cases considered to validate the operation of the proposed DRP algorithm. Section 7.4 includes simulation results to evaluate the performance of the proposed DRP algorithm and to validate its proper operation. Finally, Section 7.5 draws the main conclusions.

7.1 Related Works

The DRP problem has been tackled previously in the literature for multi-tier web applications [177].The solutions proposed for web applications can be broadly categorized into rule and model based approaches. The rule based approaches are basically based on reinforcement learning, statistical machine learning and fuzzy control. In contrast, the model based approaches relies on control theory and Queuing Theory (QT). Compared to rule based approaches, model based approaches require more domain knowledge, but they can provide Quality of Service (QoS) guarantees, while ensuring system stability [177]. Due to these distinct advantages, model based approaches became more popular for the resource management in multi-tier web systems [177].

As an example, in [178], *Urgaonkar et al.* propose DRP solutions based on QT for multi-tier Internet applications. To predict the workload, the authors employ a combination of predictive and reactive methods that determine when to provision the resources, both at large and small time scales. To predict the performance of the system and determine how much to provision, each tier instance is modeled as an isolated G/G/1 queue. A target response time is set for each tier and they determine its sizing regardless of the rest of tiers.

In the context of softwarized networks there are several works that have addressed the DRP problem using both rule based approaches [179, 180] and model based approaches [108, 137, 165].

In [179], *Mijumbi et al.* propose a graph neural network-based algorithm for the dynamic resource management of chains of Virtual Network Functions (VNFs). The proposal exploits the VNF Forwarding Graph (VNFFG) information and the resource utilization profiles of the Virtual Network Function Components (VNFCs) that make up the VNFs chain to predict future resource requirements. Specifically, the future required resources for each VNFC are predicted from its observed resource utilization profile and that observed at is neighbors. To evaluate the performance of the solution, the authors use a deployment of a virtualized IP Multimedia Subsystem (IMS). According to their results, the proposal can achieve a prediction accuracy of about 90%, though it might require a considerable amount of memory for large chains of VNFs.

In [180], *Arteaga et al.* suggest an adaptive scaling mechanism for Network Functions Virtualization (NFV) based on Q-Learning and Gaussian Processes. In the Q-Learning method, an agent interacts with an environment and learns by trial and error. This approach is adaptive but mistaken decisions may be taken until the agent learns an optimal scaling policy. To evaluate the solution, the authors considers a virtualized Mobility Management Entity (MME) with a three-tiered design. Specifically, the mechanism aims at scaling in/out the Worker (W) tier, while keeping the mean response time of the virtualized Mobility Management Entity (vMME) under a given threshold. The authors corroborate by simulation that their proposal outperforms the mechanisms based on static threshold rules, which are widely used, in terms of accuracy.

In [137], *Ren et al.* propose a DRP algorithm for the virtualized Evolved Packet Core (vEPC) considering the capacity of legacy network equipment already deployed. To evaluate the performance of their solution, they model each logical node of the vEPC as a M/M/m/K queue and assume that the VNF instantiation time is exponentially distributed. The algorithm employ two thresholds, which depend on the number of requests in the system, to decide when to power up or down a VNF instance. The authors evaluate and validate their proposal by means of simulation and numerical results. They conclude that DASA can significantly reduce operation cost and the performance models provide a quick way to evaluate the cost-performance tradeoff and system design.

Tanabe et al. in [165] develop a bi-class (*e.g.*, Machine-to-Machine (M2M) and Mobile Broadband (MBB) communications) queuing model for the vEPC. The Control Plane (CP) and User Plane (UP) of the vEPC are respectively modeled as M/M/m/m and M/D/1 nodes. This model constitutes the core of the vEPC-ORA method which aim at optimizing the resource assignment for the

CP and Data Plane (DP) of the vEPC. The authors evaluate the blocking rate reduction effect of the vEPC-ORA solution by numerical analysis. The results show that vEPC-ORA technique minimizes the blocking rates of M2M sessions and smartphone sessions.

In [108], we analyze the performance of a vMME with a three-tiers design, inspired by web services, by using a Jackson network, *i.e.*, a network of M/M/m queues. In that work, each queue represents a tier of the vMME. Additionally, we validate the model by simulation and show that it provides fairly good results for computational resources dimensioning.

Invariably, all the works based on QT performance modeling [108,137,165,178] carry out the sizing of each entity of the system independently of the rest of entities. The DASA proposed in the present chapter relies on the model introduced in Chapter 5. In that model, a network service or chain of VNFs is modeled as an open network of G/G/m queues, where each queue represents an VNFC instance. Moreover, the DASA described in this chapter takes as input a target global system response time (or delay budget) and automatically distributes it among the constituent entities of the system, while performing the sizing of the system. This leads to a better utilization of resources.

7.2 DASA for Network Services

7.2.1 Problem Formulation for the Resources Dimensioning

Let us assume a network service composed of a set of N_E interconnected entities E each implementing a specific network function. Each entity $e \in E$ of the network service is deployed as a VNF and might have multiple instances running on different Virtual Machines (VMs) or OS-level containers. The amount of computational resources (in number of processing instances, say) allocated to the instance i of the entity e is denoted by $m_i^{(e)}$. The network service has to meet an overall mean delay constraint. Let T denote the overall mean delay metric of the network service to be kept below a threshold T_{max}. The metric T is a weighted sum of the mean response times of the different entities, *i.e.*, $T = \sum_{e \in E} \delta_e \cdot T_e$, where T_e is the mean response time of the entity e and $\delta_e \in \mathbb{R}$ is an entity-specific constant. Assuming that the workload is distributed among the instances of a given entity e according to its processing capacity, the mean response time of each entity e can be computed from the mean response times of its different instances $T_i^{(e)}$ as:

$$T_e = \frac{\sum_i m_i^{(e)} \cdot T_i^{(e)}}{\sum_i m_i^{(e)}}.$$

The problem of resources dimensioning of the network service, considering that the objective is to minimize the amount of required computational resources, can be formulated as follows:

$$minimize \left(\sum_{e \in E} \sum_{i} m_i^{(e)} \right) \tag{7.1}$$

Subject to :

$$C1: \quad T \leq T_{max}, \tag{7.2}$$

$$C2: \quad m_i^{(e)} \leq m_{max} \quad \forall \quad k \in [1, K] \cap \mathbb{N} \tag{7.3}$$

Constraint 1 guarantees that the overall mean delay metric T to serve the current workload is under a maximum delay threshold T_{max}. Constraint 2 limits the maximum number of processing instances allocated to the instance i of the entity $e \in E$. The second constraint is considered because physical server has a maximum number of physical cores and they are shared among several VMs. Consequently, the higher the number of physical cores requested for an entity instance the more difficult is to find room for them in the substrate infrastructure.

Note that other performance requirements could be also considered such as a maximum jitter and/or a maximum packet loss probability. Here, we only focus on guaranteeing an overall mean delay metric for the network service. This is because the dimensioning algorithm proposed in the next subsection relies on the performance model for network services described in Chapter 5 which only provides the mean response time per entity.

7.2.2 Dynamic Resource Provisioning

The DRP algorithms enables the network service to adapt its resources depending on the workload predicted in the short-term future so that a set of performance requirements are always guaranteed. Here, as mentioned in the previous subsec-

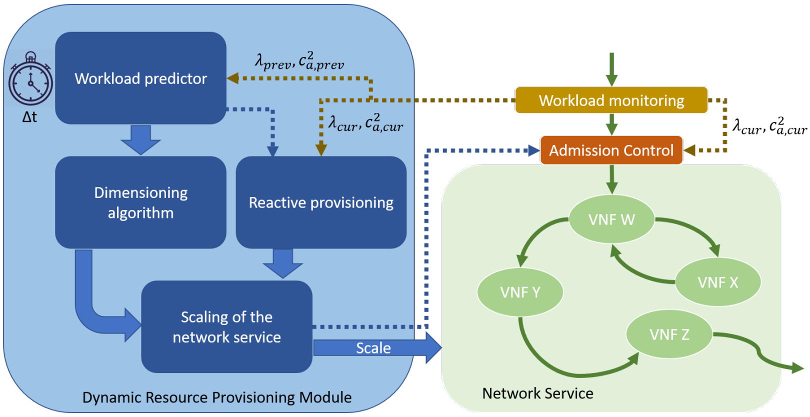

Δt
Workload predictor
$\lambda_{prev}, c^2_{a,prev}$
$\lambda_{cur}, c^2_{a,cur}$
Dimensioning algorithm
Reactive provisioning
Scaling of the network service
Scale
Dynamic Resource Provisioning Module
Workload monitoring
Admission Control
$\lambda_{cur}, c^2_{a,cur}$
VNF W
VNF X
VNF Y
VNF Z
Network Service

Figure 7.1: Main functional blocks of the DRP solution for a network service.

tion, we will consider $T \leq T_{max}$ as performance requirement, where T is a generalization of the mean response time of the network service defined as a weighted sum of the mean response times of its constituent entities. The dynamic and automatic adaptation of the resources allocated to the network service leads to a better utilization of them. Moreover, it improves the availability of the network service as it can remain operational before unexpected workload surges.

Figure 7.1 depicts the main steps of the DRP solution proposed for a network service. The main functional blocks of the DRP module are the following:

- **Workload predictor**: This block is responsible for estimating the peak demand for the network service until the next decision to provision is taken. The workload predictor is executed synchronously every Δt units of time. The value of Δt could be established from statistics of the workload arrival process in order to find a balance between the rate of scaling requests issued by the DRP module and resources savings. The workload predictor might be implemented by using for instance machine learning techniques. This block receives as input the statistics of the peak traffic workload arriving at the network service (*e.g.*, mean arrival rate λ_{prev} and Squared Coefficient of Variation (SCV) of the packet inter-arrival times $c^2_{a,prev}$) during the last period Δt. These statistics are measured by a workload monitoring agent and reported to the DRP module every Δt units of time. As output, this block provides the predicted values of the mean arrival rate λ and the SCV of the packet inter-arrival times c_a^2 of the peak traffic demand for the next period of length Δt.

ii) **Dimensioning algorithm**: This block is in charge of the sizing of the computational, network, and the number of virtualization containers (VMs and or OS-level containers) from λ and c_a^2 so that $T \leq T_{max}$ during the next period of length Δt.

To that end, we propose an heuristic based on the analytical model proposed in Chapter 5 for the resources dimensioning of a network service (see Algorithm 5). As input, the algorithm requires T_{max}, λ, c_a^2, $\boldsymbol{\mu}$, and $\boldsymbol{c_s^2}$. The inputs $\boldsymbol{\mu}$ and $\boldsymbol{c_s^2}$ are column vectors containing respectively the mean service rate and the SCV of the service times for all the entities in the set E. The mean service rate μ_e and the SCV of the service times c_{es}^2 per entity can be measured offline for a given processing instance type.

The Algorithm 5 searches for the minimum number of processing instances to be allocated to the network service so that $T < T_{max}$. The algorithm iterates until $T < T_{max}$. At each iteration it seeks for the entity $e^* \in E$ that most contributes in the reduction of T when one additional processing instance is allocated to such entity e^*. Then, the algorithm actually assigns one additional processing instance to entity e^*.

Note that, in Algorithm 5, $\oslash$ denotes the Hadamard division, defined as the element-wise division of two vectors, *i.e.*, each element of the vector placed at the left side of the operator is divided by the corresponding element of the vector located to the operator's right. In the same way, the operator $\lceil \cdot \rceil$ returns the result to apply the ceiling function, which provides the nearest integer up of its argument, component by component for the vector located inside the operator.

Table 7.1 includes the main notation used in Algorithm 5.

- **Scaling of the network service**: This block is in charge of initiating the required procedures for allocating or releasing the network service resources. As input, it uses the required processing instances per entity $\boldsymbol{m}$ provided by either the dimensioning algorithm or the reactive provisioning block whose functionality is described in the next bullet point. It keeps track of the resources currently allocated to the network services. Then, it can determine how much resources have to be reserved or freed given the output of the dimensioning algorithm or the reactive provisioning block.

- **Reactive provisioning**: This block receives frequently the current statistics of the traffic demand (λ_{cur} and $c^2_{a,cur}$) measured by the workload monitoring agent. Its mission is to trigger asynchronous resources scaling requests when it detects an unexpected workload surge that has not been foreseen by the workload predictor.

It is worthy to note that the DRP module interacts with the admission control procedure of the network service. The admission control procedure enables the network service to decline excess requests during temporary overloads. To that end, the admission control procedure might use the current statistics of the traffic demand (λ_{cur} and $c^2_{a,cur}$) provided by the workload monitoring agent and the information of current capacity of the network service to serve requests. Although the reactive provisioning block will react before unexpected workload surges, the reaction time might be non-negligible (execution time of the reactive provisioning algorithm, time to carry out the procedures of resources reservation, execution time of the resources embedding algorithm, time to instantiate new VMs, etc.). Then, it is required an admission control mechanism to guarantee that the performance requirements for the network service are met all the time.

For the interested reader we recommend the reference [164] that addresses the auto-scaling procedures for network services defined in ETSI NFV standards. Moreover, it explains how the DRP algorithms are integrated within the NFV management and orchestration framework.

7.3 Study Case: Dynamic Provisioning of an E2E Network Slicing Orchestration System

A NSOS is in charge to create, manage, and operate a large number of network slices on top of a common network infrastructure in a scalable, dynamic, and reliable manner, while satisfying their requirements [11, 181].

Table 7.1: Primary notation used in Algorithm 5.

Notation	Definition
N_E	Number of constituents entities of the network service.
$\mathbf{1}_{N_E \times 1}$	Column vector of dimension N_E with all its entries being 1.
T_{max}	Target mean delay metric for the network service.
T	Mean delay metric of the network service.
λ	External mean arrival rate to the network service.
c_a^2	SCV of the packet inter-arrival times to the network service.
$\boldsymbol{\mu}$	Vector containing the mean service μ_e rate for each entity e.
$\boldsymbol{c_s^2}$	Vector containing the SCV of the service times c_{es}^2 for each entity e.
$\boldsymbol{I}$	Vector storing the number of instances per entity.
$\boldsymbol{m}$	Vector storing the number of processing instances m_e allocated to each entity e.
m_{max}	Maximum number of processing instances that can be allocated to a single entity instance.
$\boldsymbol{T_E}$	Vector storing the mean response time T_e for each entity e.
$\boldsymbol{P}$	Transition probability matrix of the network of queues that models the network service.
$\boldsymbol{\lambda_E}$	Vector storing the aggregated mean arrival rate λ_e at each entity e.
$\boldsymbol{\lambda_I}$	Vector storing the aggregated mean arrival rate for each and every instance of all the entities.
$\boldsymbol{c_{aI}^2}$	Vector storing the SCV of the inter-arrival times for each and every instance of all the entities.
δ_e	A constant associated with the entity e ($T = \sum_{e \in E} \delta_e \cdot T_e$).

Algorithm 5 Dimensioning algorithm.

Input: T_{max}, λ, c_a^2, $\boldsymbol{\mu}$, $\boldsymbol{c_s^2}$.

Output: Required number of instances (or virtualization containers) $\boldsymbol{I}$ per entity and the processing instances to be allocated to each entity $\boldsymbol{m}$.

```
Initialization Compute $\boldsymbol{\lambda_E}$ using (5.27); $\boldsymbol{m} = \lceil \boldsymbol{\lambda_E} \oslash \boldsymbol{\mu} \rceil$; $\boldsymbol{I} = \lceil \boldsymbol{m}/m_{max} \rceil$; Compose the network of queues and compute $\boldsymbol{P}$; Compute internal flows parameters $\boldsymbol{\lambda_I}$ and $\boldsymbol{c_{aI}^2}$ using (5.12)-(5.18); Estimate $\boldsymbol{T_E}$ given the initial stability conditions using (5.19)-(5.23); $T = \sum_{e \in E} \delta_e \cdot \boldsymbol{T_E}(e)$; $\boldsymbol{T_E}^{(prev)} = \boldsymbol{T_E}$.
while $T > T_{max}$ do
    $\boldsymbol{m_{aux}} \Leftarrow \boldsymbol{m} + \mathbf{1}_{N_E \times 1}$; $\boldsymbol{I_{aux}} = \lceil \boldsymbol{m}/m_{max} \rceil$;
    Recompose the network of queues for $\boldsymbol{m_{aux}}$ and $\boldsymbol{I_{aux}}$; and recompute $\boldsymbol{P}$.
    Recompute $\boldsymbol{\lambda_I}$ and $\boldsymbol{c_{aI}^2}$ using (5.12)-(5.18);
    Estimate $\boldsymbol{T_E}$ using (5.19)-(5.23) with the above input parameters ($\boldsymbol{m_{aux}}$, $\boldsymbol{P}$, $\boldsymbol{\lambda_I}$, $\boldsymbol{c_{aI}^2}$);
    $e^* = \underset{e \in E}{\operatorname{argmax}} \left( \delta_e \cdot \left( \boldsymbol{T_E}^{(prev)}(e) - \boldsymbol{T_E}(e) \right) \right)$;
    $\boldsymbol{m}(e^*) \Leftarrow \boldsymbol{m}(e^*) + 1$; $\boldsymbol{I}(e^*) \Leftarrow \lceil \boldsymbol{m}(e^*)/m_{max} \rceil$;
    $T \Leftarrow T - \delta_e \cdot \left( \boldsymbol{T_E}^{(prev)}(e^*) - \boldsymbol{T_E}(e^*) \right)$; $\boldsymbol{T_E}^{(prev)}(e^*) \Leftarrow \boldsymbol{T_E}(e^*)$;
end while
```

This section describes and addresses the modeling of the NSOS proposal developed within the 5G!Pagoda project [181, 182]. Later in this chapter, we will evaluate the performance of the DASA presented in the previous section considering this use case. Enabling the auto-scaling of the NSOS itself is essential to get the full automation of the network slicing ecosystem.

7.3.1 5G!Pagoda project's NSOS

The 5G!Pagoda project's NSOS relies strongly on the foundation of a hierarchical architecture that incorporates dedicated entities per domain to manage every segment of the mobile network (*e.g.*, Radio Access Network (RAN), Backhaul Network (BN), and Core Network (CN)) for a scalable orchestration of federated network slices. This End-to-End (E2E) NSOS is composed of a global orchestrator and multiple domain-specific orchestrators and their respective system components. Its main entities are itemized and briefly explained below:

- **Global Orchestrator (GO)** is responsible for receiving network slice orchestration requests from slice providers and orchestrating the slices with a global view in the specified cloud domain(s).
- **System Awareness Engine (SAE)** is responsible for keeping the state, context and the running resources of the entire global orchestration system's entities. It also functions as a global system monitor, which helps keep track of the system's performance.
- **Resource Awareness Engine (RAE)** is a single system entity which keeps the record of the total resources available on the underlying infrastructure of the orchestration system.
- **Domain-Specific Orchestrator (DSO)** is basically in charge of orchestrating network slices from a particular administrative domain whose operation spans across a particular network region. Every DSO operating in a particular region has at least one of the following system components needed to actualize a complete E2E network slice orchestration.
  - **Domain-Specific Network Function Virtualization Orchestrator (DSNFVO)** is responsible for communicating directly with the

region's **Domain-Specific Virtualized Infrastructure Manager (DSVIM)** which is in charge of providing virtual resources for the instantiation of virtual network functions.

- **Domain-Specific Radio Resource Orchestrator (DSRRO)** which is solely responsible for orchestrating and allocating radio resources available on already deployed evolved NodeBs (eNBs) for the utilization of network slices.
- **Domain-Specific Software-Defined Networking Controller (DSSDN-C)** is the entity responsible for connecting the various orchestrated sub-slices making up a network slice including that of the radio access network.
- **Domain-Specific Virtualized Infrastructure Manager (DSVIM)** Domain-Specific Virtualized is responsible for the provisioning of virtualized resources to the orchestrated network functions that make up the different sub-slices of a complete network slice.
- **Domain-Specific eNBs (DSeNBs)** are the already deployed set of eNBs running in a particular region administered by a DSRRO, running under a particular DSO.

The interconnection of the above entities is shown in Fig. 7.4.

Figures 7.2 and 7.3 depict the sequence diagram of the Slice Orchestration Request procedure in the 5G!Pagoda project's NSOS architecture.

7.3.2 NSOS Queuing Model

The modeling approach we will follow to model the NSOS is the same as that described in Chapter 5. Then, the model consists in an open network of G/G/m queues, where each queue represents an instance of the system like the GO or a DSO, see Fig. 7.4. For the sake of simplicity, only one instance per entity is shown in Fig. 7.4.

The mean response time of each entity instance can be computed by using equations (5.12)-(5.22) (Queuing Network Analyzer (QNA) method). However, to derive the overall mean response time of the NSOS we will need to modify the analysis of the original QNA method [156].

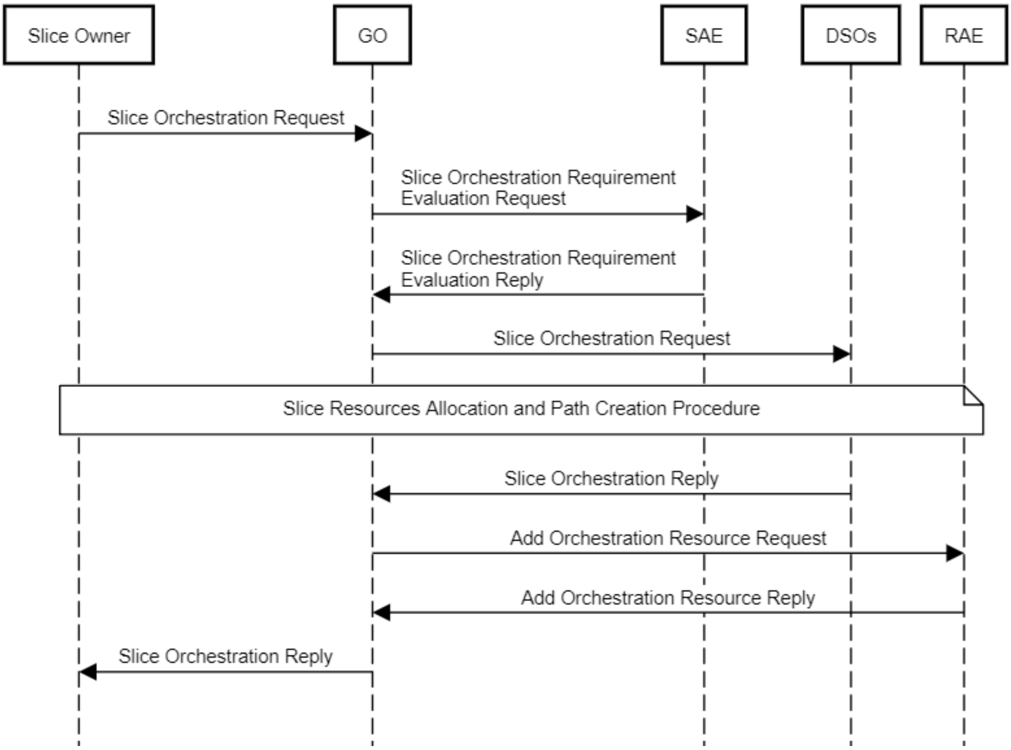

Slice Owner
GO
SAE
DSOs
RAE
Slice Orchestration Request
Slice Orchestration Requirement Evaluation Request
Slice Orchestration Requirement Evaluation Reply
Slice Orchestration Request
Slice Resources Allocation and Path Creation Procedure
Slice Orchestration Reply
Add Orchestration Resource Request
Add Orchestration Resource Reply
Slice Orchestration Reply

Figure 7.2: 5G!Pagoda project's Slice Orchestration Request call-flow.

As shown in Fig. 7.3, the DSO entity sends requests in parallel to the DSNFVO and DSRRO entities for the allocation of resources during a given slice orchestration request procedure. This blocks the call-flow at the DSO to serve a given slice request until both the DSNFVO and the DSRRO answer the request. This behavior can be captured by modeling the subnetwork composed of the DSNFVO, DSVIM, DSRRO, and DSeNBs for a given domain d as a fork/join subnetwork with two parallel branches (*e.g.*, DSNFVO/DSVIM and DSRRO/DSeNBs). Then, the branch with the highest response time will determine the response time of the fork/join subnetwork.

Let $\mathcal{K}_e \subset 1, ..., K$ with $|\mathcal{K}_e| = K_e$ be the subset of indexes associated with the instances of the entity $e \in E = \{$ GO, SAE, RAE, DSO_d, $DSNFVO_d$, $DSVIM_d$, $DSSDNC_d$, $DSRRO_d$, $DSeNBs_d$ $\}$, where the subindex $d \in [1, 2, ..., D]$ is included to specify the domain, which may be associated with a

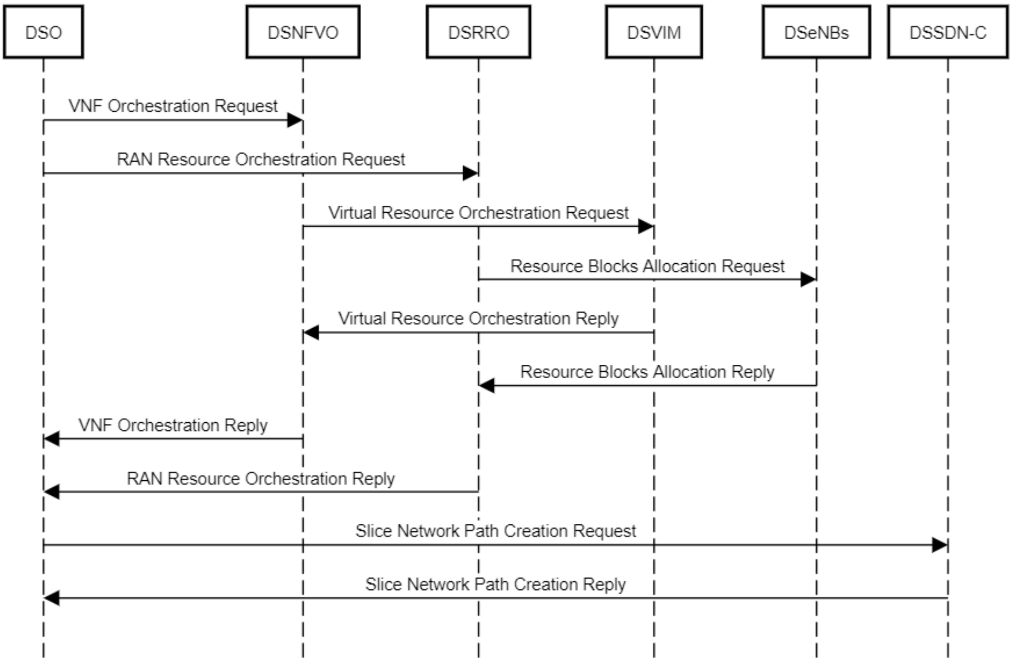

DSO
DSNFVO
DSRRO
DSVIM
DSeNBs
DSSDN-C
VNF Orchestration Request
RAN Resource Orchestration Request
Virtual Resource Orchestration Request
Resource Blocks Allocation Request
Virtual Resource Orchestration Reply
Resource Blocks Allocation Reply
VNF Orchestration Reply
RAN Resource Orchestration Reply
Slice Network Path Creation Request
Slice Network Path Creation Reply

Figure 7.3: 5G!Pagoda project's Slice Resources Allocation and Path Creation call-flow.

specific geographical region, the entity e belongs to. Then, assuming that the workload is distributed among the instances of a given entity e, the mean response time T_e of the entity e is given by

$$T_e = \sum_{k \in \mathcal{K}_e} \frac{m_k^{(e)}}{\sum_{k \in \mathcal{K}_e} m_k^{(e)}} \cdot T_k^{(e)} \cdot V_e, \tag{7.4}$$

where $m_k^{(e)}$ is the number of processing instances allocated to the instance k of the entity e, V_e is the visit ratio of the entity e, and $T_k^{(e)}$ is the mean response time of the instance k of the entity e.

If we model the $DSNFVO_d$, $DSVIM_d$, $DSRRO_d$, and $DSeNBs_d$ entities of the domain d as a fork/join subnetwork of queues with two parallel branches, the mean response time of this fork/join subnetwork will be given by

$$T_d^{(FJS)} = max\left(T_{DSNFVO_d} + T_{DSVIM_d}, T_{DSRRO_d} + T_{DSeNBs_d}\right) \tag{7.5}$$

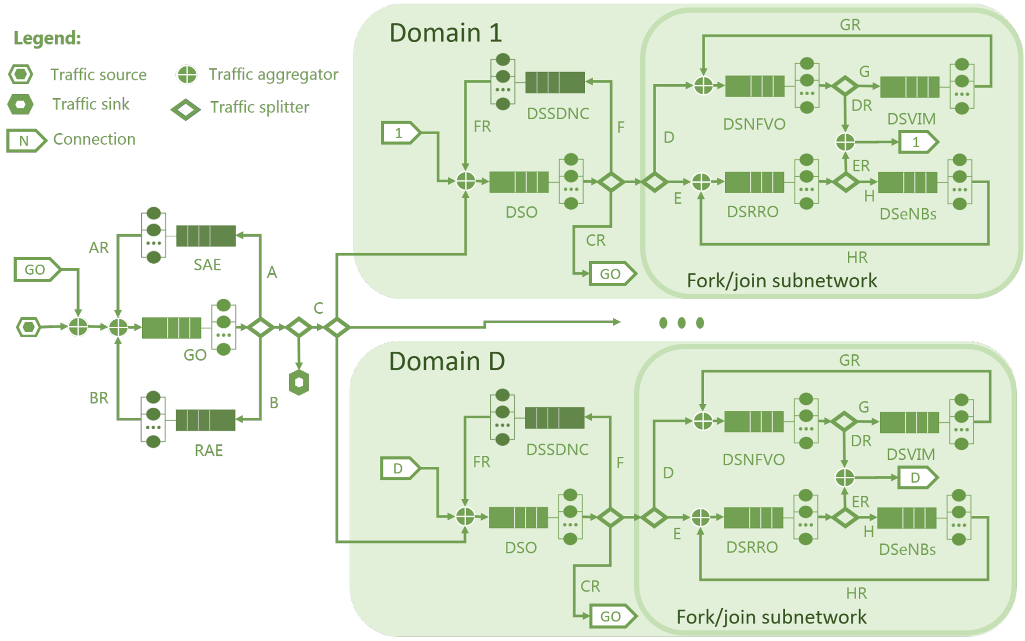


Figure 7.4: Queuing model of the 5G!Pagoda's project NSOS.

And the overall mean response time T of the NSOS can be computed as:

$$T = T_{GO} + T_{SAE} + T_{RAE} + \sum_{d=1}^{D} \left(T_{DSO_d} + T_d^{(FJS)} \right) \tag{7.6}$$

The steady-state transition probabilities, which are input parameters to solve the network of queues, can be derived directly from the sequence diagram of the Slice Orchestration Request procedure shown in Figs. 7.2 and 7.3.

Let p_{de}^{se} denote the transition probability from the source entity $se \in E$ to the destination entity $de \in E$. Trivially, p_{de}^{se} equals the number of incoming signaling messages to the source entity se divided by the number of messages sent from entity se to the destination entity de during a whole slice orchestration request procedure. The transition probabilities between entities are included in Table 7.2, where α_d is the percentage of the total incoming slice orchestration requests to the NSOS which are addressed to the domain d.

Assuming that the workload is distributed among the instances of a given entity e according to its processing capacity, the transition probability $p_{de_j}^{se_i}$ from

Table 7.2: Transition probabilities between entities for the NSOS queuing model.

DE / SE	GO	SAE	RAE	DSO_d	$DSNFVO_d/$ $DSRRO_d$	$DSVIM_d/$ $DSeNBs_d$	$DSSDNC_d$
GO	0	0.25	0.25	$\alpha_d{\cdot}0.25$	0	0	0
SAE	1	0	0	0	0	0	0
RAE	1	0	0	0	0	0	0
DSO_d	1/3	0	0	0	1/3	0	1/3
$DSNFVO_d/$ $DSRRO_d$	0	0	0	0.5	0	0.5	0
$DSVIM_d/$ $DSeNBs_d$	0	0	0	0	1	0	0
$DSSDNC_d$	0	0	0	1	0	0	0

the instance i of the source entity $se \in E$ to the instance j of the destination entity $de \in E$ is given as

$$p_{de_j}^{se_i} = \frac{m_j^{(de)}}{\sum_{k \in \mathcal{K}_e} m_k^{(de)}} \cdot p_{de}^{se} \tag{7.7}$$

where $m_j^{(de)}$ is the number of CPU cores allocated to the instance j of the destination entity de.

7.4 Results

In this section, we will assess the time complexity and optimality of the DRP algorithm introduced in Section 7.2 and verify its proper operation. Two study cases were considered in the validation of the proposed DRP algorithm:

i) The DRP of the NSOS presented and modeled in Section 7.3.

ii) The DRP of the vEPC CP whose architecture description and modeling have been addressed in Sections 6.2 and 6.4.

7.4.1 Experimental Setup

To verify the proper operation of the DRP algorithm, we developed two queuing simulators within the ns-3 environment [63] to simulate the operation and delays of the NSOS and the vEPC CP.

The same approach as described in Section 6.7.3 was used to simulate the entities instances. That is, each entity instance is simulated as a First-Come, First-Served (FCFS) queue with one or several generic servers. The specific number of servers for any entity instance depends on the processing instances allocated to it for a given workload. The distribution of the service times for each processing instance allocated to a given entity is that one depicted in Fig. 5.11 for the W instance of a three-tiered vMME. This distribution has a mean of 100 μs (or, equivalently, a service rate equal to 10000 $packets/second$) and a SCV equal to 0.65.

The simulators also implements the corresponding procedures and messages exchange between the entities instances. On the one side, the NSOS simulator implements the call flow carried out when an incoming slice orchestration request arrives at the NSOS (see Figs. 7.2 and 7.3). On the other side, the vEPC CP simulator implements the call flows of the Service Request (SR), S1-Release (S1R), X2-based Handover (HO), and Tracking Area Update (TAU) procedures (see Figs. 4.2, 4.3, and 4.4).

To the best of our knowledge, there is no work in the literature addressing the characterization of the slice orchestration requests generation process. Then, the same workload generation process was considered for both simulators. More precisely, the requests arrive at the network services according to a Poisson distribution considering the results included in Section 3.6.3.2. The aggregated mean arrival rate was modulated for one day period according to the temporal distribution measured in [12] for the aggregated mobile traffic (see Fig. 7.5). For the vEPC CP study case, the percentage of each Long-Term Evolution (LTE) signaling procedure type was determined by using (3.19), (3.20), (3.21), and (3.22), and considering the scenario shown in Fig. 3.6.

The simulator also includes the implementation of the DRP algorithm described in Section 7.2. We used an ideal predictor for the workload *i.e.*, it can predict without error the peak workload demand until the next decision to provision will be taken. The decisions to provision were taken periodically every 10 minutes. For the NSOS study case, the target mean response time of the system to serve a slice orchestration request was set to 2 ms, *i.e.*, $T_{max} = 2\ ms$. For the vEPC CP study case, we considered that $T = \delta_{MME} \cdot T_{MME} + \delta_{cSGW} \cdot T_{cSGW} +$

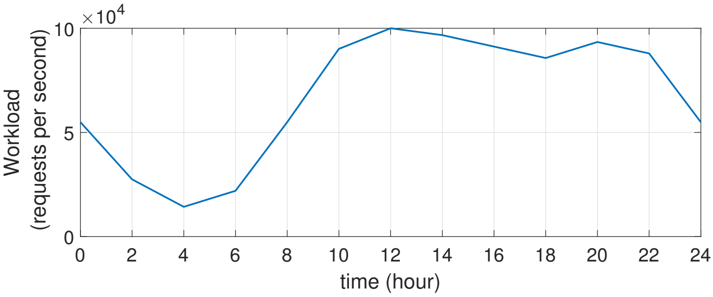


Figure 7.5: The temporal distribution of the aggregated mean arrival rate at the network services (*e.g.*, NSOS -slices orchestration requests per second- and vEPC CP -LTE signaling procedures per second-) for one day period [12].

$\delta_{cPGW} \cdot T_{cPGW} = 5 \cdot T_{MME} + 2 \cdot T_{cSGW} + 2 \cdot T_{cPGW} < T_{max} = 1.5\,ms$ (see Sections 6.2.2 and 6.6).

Table 7.3 includes the configuration of the main parameters for all the simulations carried out in the subsequent experimental evaluation. Except where otherwise noted, for the NSOS study case, we will consider the workload is equally distributed among the different domains, *i.e.*, $\alpha_d = 1/D$.

7.4.2 Performance of the Dimensioning Algorithm

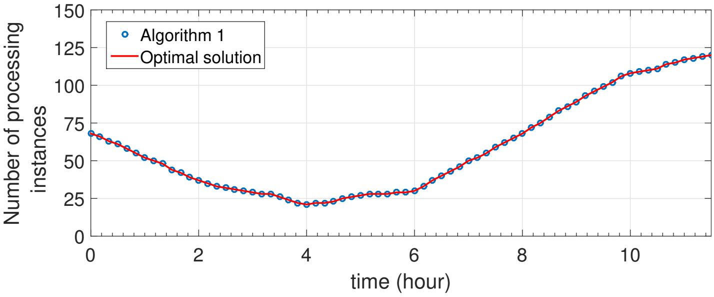


Figure 7.6: Comparison of the required computational resources estimated by our proposed dimensioning algorithm and the optimal solution.

In order to gauge the performance of the resources dimensioning algorithm (*i.e.*, the Algorithm 5), the following 3 metrics were considered:

Table 7.3: Parameters configuration.

External arrival processes	
Arrival process at the network service	Modulated Poisson process whose arrival rate versus time is given in Fig. 7.5
Service processes	
Service rate of each processing instance	10000 packets per second
SCV of the processing instance service time	0.65
Maximum number of processing instances allocated to any entity instance, m_{max}	10
QoS requirements	
T_{max} in the NSOS study case	2 ms
T_{max} in the vEPC CP study case	1.5 ms

i) the difference between the resources estimated by the algorithm and the optimal solution,

ii) the difference between the mean response time achieved by the algorithm and the optimal solution, and

iii) the time complexity of the algorithm.

Although the results included in this section are for the NSOS study case, similar results were obtained and identical conclusions were reached for the the vEPC CP study case.

The optimal solution of the dimensioning problem was computed by using the exhaustive search method. As in Algorithm 5, we initialized the initial processing instances allocation to each entity using the condition for queuing network stability (*i.e.*, $\boldsymbol{m_e} = \lceil \boldsymbol{\lambda_e} \oslash \boldsymbol{\mu_e} \rceil$). Then, assuming there are N_E different entities in the network service, $M_0 = \sum_{i=1}^{N_e} \boldsymbol{m_e}(i)$ processing instances are allocated in the initial assignment to fulfill the stability condition. Then, the exhaustive search algorithm iterates until the system response time is below the maximum delay

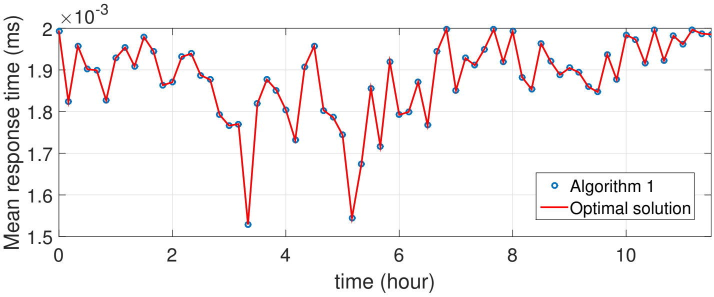


Figure 7.7: Comparison of the overall mean response time of the system achieved by our algorithm and the optimal solution.

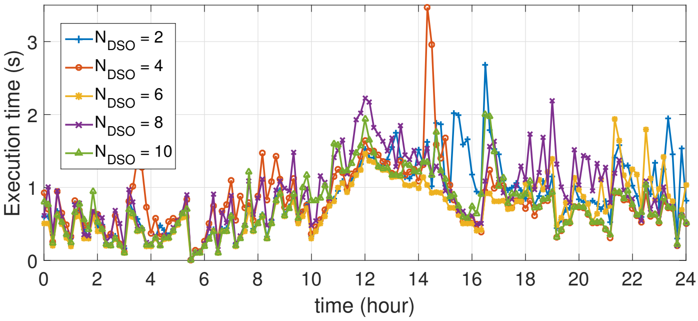


Figure 7.8: Resources dimensioning algorithm time complexity.

threshold, *i.e.*, $T < T_{max}$. At each iteration the brute-force algorithm increments by one the total number of processing instances allocated to the whole system, M. Thus, it checks every combination to allocate the $M - M_0$ processing instances among the entities of the system. It chooses that allocation that achieves the minimum mean response time of the system. Observe that the number of checks the algorithm has to carry out at each iteration is N_e multichoose $M - M_0$, that is

$$\left(\!\!\binom{N_E}{M - M_0}\!\!\right) = \frac{(N_E + M - M_0 - 1)!}{(M - M_0)!(N_E - 1)!}. \tag{7.8}$$

Figures 7.6 and 7.7 depict respectively the comparison between our algorithm and the optimal solution to estimate the amount of required computational resources and the mean response time achieved with each estimation. Because finding the optimal solution by the brute-force approach is computationally intensive, the scenario considered in this evaluation had only one DSO entity and the evaluation was carried out only for the first twelve hours of the day. As it is observed, the dimensioning algorithm described in this chapter achieves the optimality goals for the scenario considered.

Finally, the time complexity of the dimensioning algorithm was assessed (see Fig. 7.8). To that end, its execution time was measured for different number of DSOs N_{DSO} or domains D. It should be noted that each additional DSO included in the scenario accounts for six additional logical entities in the NSOS (*i.e.*, DSO, DSSDNC, DSNFVO, DSVIM, DSRRO, DSeNBs). Then, for $N_{DSO} = 2$ the slices orchestration system is composed of 15 different entities ($N_E = 15$). Each point in Fig. 7.8 represents the average of the measurements obtained for 3 independent runs.

As shown in Fig. 7.8, the execution time of the dimensioning algorithm does not exhibit a clear dependence with the number of entities N_E. This result is explained by the fact that the number of queues K in the model is given by the number of instances for all the entities. Although for low workloads the number of queues is given by the number of entities (at least we will have one instance per entity), for high workloads the number of queues are dominated by the workload. The shape of the algorithm execution time roughly resembles the shape of the temporal distribution of the workload (see Fig. 7.5). Then, we can conclude that the algorithm exhibits linear time complexity with the workload. Note that the execution time of the dimensioning algorithm clearly depends on how close it is the optimal solution to the initial solution provided by the algorithm which is based on the stability condition for the model. This explains the short-term fluctuations observed in the algorithm time complexity over time.

7.4.3 Proper Operation Verification of the DRP Algorithm

Finally, we will check that the DRP algorithm detailed in Section 7.2 works properly.

7.4.3.1 NSOS Study Case

In this assessment, $N_{DSO} = 3$ or $D = 3$ was considered.

Figure 7.9 depicts the total required number of processing instances over time predicted by the DRP algorithm. In the same way, Fig. 7.10 shows the number of processing instances allocated to each entity over time according to our DRP algorithm. As the GO has to process the highest number of messages per control procedure, it presents the greatest demand of resources.

As it is shown in Fig. 7.11, for the resources allocation performed by the DRP algorithm the maximum delay threshold is always met, thus verifying the proper operation of the algorithm.

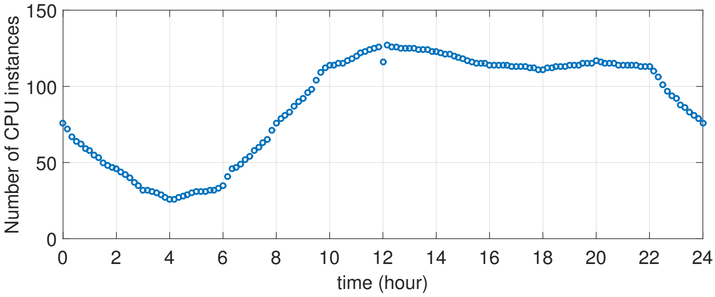


Figure 7.9: Total number of processing instances required by the NSOS to met the delay budget (2 ms).

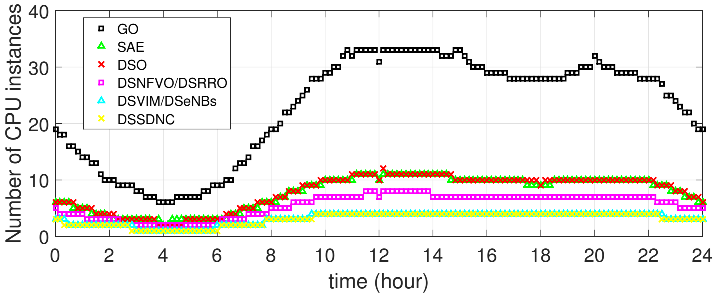


Figure 7.10: Number of processing instances allocated to each NSOS entity to met the delay budget (2 ms).

7.4.3.2 vEPC CP Study Case

In this evaluation, the simulation scenario had 2000000 User Equipments (UEs) and a population density of 1000 inhabitants per km^2. Note that the HO and TAU generation rates depend on the Evolved-Universal Terrestrial Radio Access Network (E-UTRAN) density, which, in turn, depends on the population density in our simulation tools. As performance requirement, we considered that the

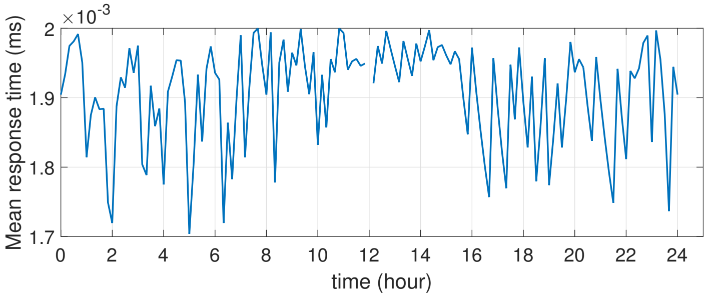


Figure 7.11: NSOS response time.

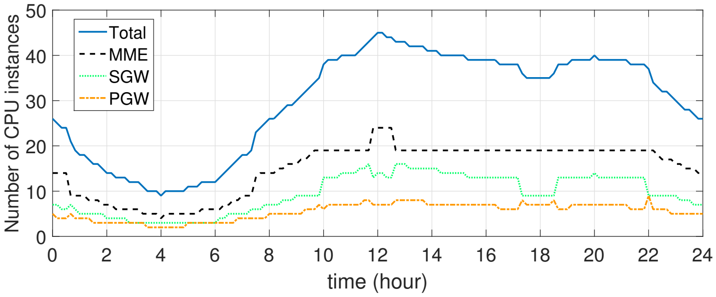


Figure 7.12: Number of dedicated CPU instances per entity of the vEPC CP.

average elapsed time to move an UE from IDLE state to ACTIVE state has to be kept below 25 ms. The processing and propagation delays assumed for the LTE entities and interfaces are those included in Table 6.1. This implies a processing delay of 1.5 ms for the vEPC to execute a whole SR procedure in the worst case scenario (see Sections 6.2.2 and 6.6).

Figure 7.12 depicts the required computational resources estimated by our DRP algorithm. As it is shown in Fig. 7.13, the LTE CP delay budget is always met, thus validating the operation of our DRP algorithm for the vEPC CP study case.

In order to show that the joint dimensioning of all entities leads to a better resource utilization, we carried out the sizing of each entity separately. The overall processing delay budget $T^{(vEPC)}_{proc-budget}$ was distributed as follows: 56% for the MME, 22% for the cSGW, and 22% for the cPGW. Note that a different assignment can be considered as long as $T^{(e)}_{budget} > 1/\mu_e$ (a necessary but not a sufficient

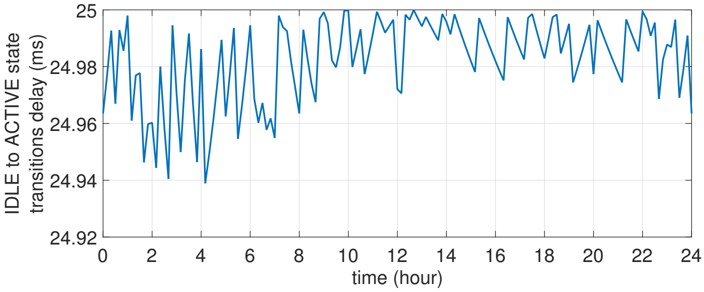


Figure 7.13: Mean response time of the LTE CP to move a UE from IDLE state to ACTIVE state.

condition to guarantee that the dimensioning problem has a solution), where $T_{budget}^{(e)}$ is the processing budget assigned to entity e and μ_e is the service rate of a processing instance allocated to the entity e. Figure 7.14 depicts the total number of required processing instances estimated by performing the dimensioning for each entity separately (labeled as "individual sizing") and the sizing for each entity jointly (labeled as "joint sizing"). Compared to the approach of sizing each entity separately, the joint sizing achieves up to 11% of resources saving for the setup considered. As mentioned before, a different distribution of $T_{proc-budget}^{(CP)}$ among the entities will yield different results. For instance, assigning 33% of the processing budget ($T_{proc-budget}^{(vEPC)} = 2.5\,ms$) to each entity, we achieved a resources saving of up to 21%. Therefore, performing the dimensioning of all the entities jointly leads to resources saving.

7.5 Conclusions

The DRP algorithms allow network services to automatically and dynamically scale their resources so that a set of performance requirements are always met. This leads to a better resource utilization as the network services have not to be over-dimensioned anymore as in the traditional networking approach. Then, operators can handle workload fluctuations to keep the desired performance with great agility and reduced costs.

In this chapter, we have proposed a DRP solution for network services. The main functional blocks of the DRP solution are a workload predictor, a dimension-

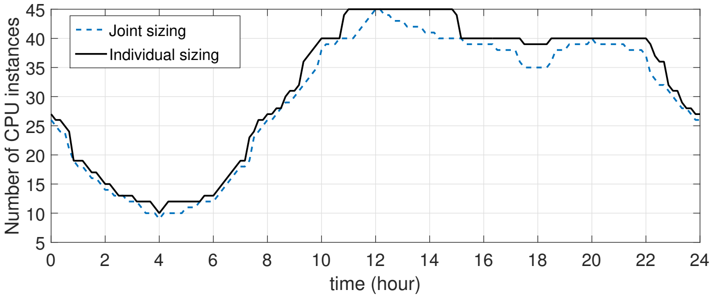


Figure 7.14: Comparison between individual dimensioning per entity and joint dimensioning of all entities in terms of computational resources.

ing algorithm, a handler responsible for the scaling of the network service, and a reactive provisioning module. The workload predictor is executed synchronously and is responsible for estimating the peak demand for the network service until the next decision to provision is taken. The dimensioning algorithm relies on the performance model proposed in Chapter 5 to carry out the estimation of the amount of required resources from the output of the workload predictor. The handler of the scaling of the network service is in charge of triggering the corresponding NFV scaling procedures and keeps track of the resources currently allocated to the network service. Last, the reactive provisioning module triggers asynchronous scaling requests when it detects an unexpected workload surge that has not been foreseen by the workload predictor.

The proper operation of the DRP solution has been validated by means of simulations for two use cases:

i) Dynamic provisioning of an E2E NSOS whose performance modeling have been also addressed in this chapter.

ii) Dynamic provisioning of the vEPC CP whose performance modeling have been tackled in Chapter 6.

The performance of the proposed dimensioning algorithm in terms of execution time complexity and optimality has been measured for both study cases. The results show that the execution time of the dimensioning algorithm exhibits a linear dependence with the workload and apparently no dependence with the number of constituent entities of the network service. Regarding the optimality, it has been observed that, given a workload and a target mean delay metric (any metric that is a function of the mean response times of the constituent entities) for the network service, the algorithm finds the minimum number of required resources to fulfill the performance requirement of the network service. Moreover, among the set of solutions for the resources distribution among the constituent entities it finds that one providing the best QoS.

Finally, the algorithm performs the dimensioning of all the constituent entities of a network service jointly. The results show that compared to the approach of sizing each entity separately, the joint sizing leads to a better utilization of resources. That is because the algorithm distributes the processing delay budget optimally (in terms of resources saving) among the entities of the network service given a workload and setup for the network service.

Chapter 8

Conclusions and Outlook

Network Softwarization (NetSoft), whose main enablers are Software Defined Networking (SDN) and Network Functions Virtualization (NFV), has emerged as a promising paradigm during the past few years to meet the needs of the future networks. NetSoft is deeply impacting and bridging Telecom and IT industries, and it will radically transform the way network are designed and operated to deliver services and applications in an agile and cost effective way.

In this thesis we have studied the feasibility to adopt NetSoft paradigm for realizing the future 5G mobile networks, which is currently a hot topic for research in mobile broadband networks. Moreover, we have addressed the modeling of the workload and the performance of the future softwarized mobile networks. Based on the resulting models, we have proposed solutions for the automated planning and the Dynamic Resource Provisioning (DRP) of the softwarized mobile networks. The rest of the chapter is organized as follows. Section 8.1 draws the main conclusions extracted from the work carried out in this thesis. Section 8.2 lists the main contributions of this thesis. Finally, Section 8.3 discusses directions for future work related to the topics covered in this thesis.

8.1 Main Conclusions

The most relevant conclusions extracted from the work developed in this thesis are the following:

Ch1 One of the major challenges of 5G technology is to accommodate a wide range of use cases with different stringent requirements. For the sake of simplicity, these use cases have been broadly categorized into the following service groups: i) enhanced Mobile Broadband (eMBB), ii) massive Machine Type Communication (mMTC), and iii) Ultra-Reliable and Low Latency Communications (URLLC). NetSoft paradigm overcomes this issue by enabling the concept of Network Slicing which refers to create isolated, fully automated, programmable, flexible, and service-customized networks, known as network slices, on top of a common physical infrastructure.

Ch2 NFV and SDN paradigms are expected to play a paramount role in the future mobile networks because they offer high level of programmability, flexibility, scalability, cost effectiveness, automation, and agility to deploy new services and network functionalities.

We have proposed an architecture for the future mobile networks that integrates both paradigms. In contrast to current Third Generation Partnership Project (3GPP) mobile networks, the proposed architecture employs Multiprotocol Label Switching (MPLS) tunnels at User Plane (UP) instead of GPRS Tunneling Protocol (GTP) tunnels which introduce significant overhead and its management might be cumbersome. It also provides flexibility to fix the endpoints of the MPLS tunnels, thus the internal communications within the network have not to pass through the core gateway (e.g., Regional Router (RR)), which is the bottleneck of the UP, anymore. This feature also enables the deployment of local breakouts within the Core Network (CN) for accessing local services with low delay.

The proof of concepts carried out in this thesis support the feasibility of the proposed softwarized mobile network architecture in terms of performance, and thus the practicability of the adoption of NFV and SDN paradigms in future mobile networks. First, the performance offered by today's SDN switches is enough to realize the UP of 5G mobile networks. Second, the proposed architecture can effectively provide mobility support, which is one of the most important features of the mobile networks, while fulfilling the Control Plane (CP) delay targets of 5G mobile networks.

Ch3 Understanding the characteristics of the foreseen traffic demands for both the CP and the UP is of utmost importance to design and optimize the future 5G mobile networks.

Besides the traditional human-centric services, referred to as Mobile Broadband (MBB) services, 5G technology will efficiently support mMTCs which are characterized by a huge number of connected low-cost devices equipped with long-life batteries typically transmitting infrequent, small, and non-delay-sensitive data. In contrast to traditional human-centric services or MBB services, mMTCs exhibit a highly homogeneous traffic which might be coordinated on small timescales. Markov-modulated Poisson processs (MMPPs) are well suited for traffic source modeling of mMTCs, offering accurate results and linear time complexity with the number of devices. Interestingly, the results obtained in this thesis show that mMTCs generates roughly 3.5 times more signaling workload than MBB services. This result suggests the definition of new, and more lightweight and energy-efficient signaling procedures for mMTCs.

Regarding the MBB services, their signaling workload depends on the user mobility and behavior, Radio Access Network (RAN) setup, inactivity timer value, and stochastic characteristics of UP traffic. Moreover, the aggregated signaling generation process in mobile networks is roughly Poissonian. Whereas the aggregated UP traffic for MBB services presents Self-Similarity and Long-Range Dependence features, thus it can be modeled as a fractal Brownian motion (fBm) process.

Ch4 The three-tiered decomposition of the Virtual Network Functions (VNFs), inspired by web services, entails several advantages such as a high level of flexibility, elasticity and resiliency; and to ease the dynamic scaling of the VNF. However, this VNF design option increases the VNF response time, as every packet has to pass through several nodes (*e.g.*, Front-End (FE), Worker (W), and DataBase (DB)). That is the main reason why this kind of VNF decomposition has been mainly considered for CP functionalities, where the delays constraints are less stringent than for UP functionalites.

The results obtained in this thesis show that it is feasible in terms of performance and scalability to use this decomposition for the virtualized CP entities of a mobile network. For instance, a three-tiered virtualized Mobility Management Entity (vMME) with ten m3.xlarge processing instances of the Amazon EC2 service allocated to the W tier can serve up to 37000 LTE signaling procedures per second, while guaranteeing a mean response time of 1 ms.

Ch5 Performance modeling has interesting applications in the context of softwarized networks to enable the automation of the deployment and dynamic scaling of network services. For instance, it can be used to translate the performance requirements of a network service into the required computational, network, and virtual resources (resource dimensioning).

Queuing networks are a proper tool to model the performance of any composition of VNFs. To derive the performance metrics of the network, Queuing Theory (QT) has two standard methodologies of analysis: Mean Value Analysis (MVA) algorithm and Jackson networks assumptions. On the one hand, Jackson networks offer limited accuracy in some scenarios, as they assume Poisson arrivals and service times processes. On the other hand, MVA exhibits long execution times in scenarios characterized by a large number of active user sessions such as cellular networks. In addition, MVA is a methodology to solve closed queuing networks. To model a chain of VNFs, which might be an open system in nature, using a closed queuing network, we have to assume that a packet flow utilizes the resources of only one Virtual Network Function Component (VNFC) instance at a given time. Then, MVA is not general enough to capture the behavior of all possible chains of VNFs.

To overcome the above-mentioned limitations, the Queuing Network Analyzer (QNA) method [156] could be used to derive the performance metrics of the chains of VNFs. QNA method is an approximate technique to solve networks of G/G/m queues. The most interesting aspect of QNA method is that it provides a set of linear equations to estimate the second-order moments of the internal flows.

We have evaluated experimentally the accuracy of the QNA method considering a three-tiered vMME. The results show that it outperforms MVA and Jackson networks techniques in terms of estimation error for the use case studied. Specifically, for medium and high utilizations, QNA method achieves less than half of error compared to Jackson and MVA approaches.

Ch6 NetSoft paradigm enables operators to dynamically adapt the resouces allocated to each network slice and services. However, on-demand plans offered by cloud providers are more expensive than reservation plans. More precisely, resources can be purchased as a reservation for up to 70% off the on-demand price. Then, the resource dimensioning and allocation processes during the network slices planning is important for operators to save money.

3GPP standards define a set of performance requirements for mobile networks. For the CP, it is defined a delay budget to move an User Equipment (UE) from idle to active state. For the UP, the performance requirements are a maximum delay budget and a maximum packet loss probability. The UP delay budget is the maximum time it takes for a packet to travel from the SGi interface to the UE application.

The resource dimensioning and its allocation among a set of candidates Edge Clouds (ECs) are closely related and to perform them in a coordinated way brings some benefits in terms of resources saving or Quality of Service (QoS) improvement. The joint optimization of resource dimensioning and allocation of the virtualized Evolved Packet Core (vEPC) can be formulated as a multi-objective optimization problem, where the objectives might be to minimize the workload imbalances among the set of candidates ECs, the amount of resources, and delays. The primary constraints of this problem is to guarantee the QoS requirements specified by 3GPP for mobile networks.

We have tackled the problem of vEPC planning using a performance modeling approach. We have validated the proper operation of the proposed solution, which is dubbed “Planner for the Evolved Packet Core (EPC) as a Service” (PES), by means of simulations.

We have also compared the algorithm included in PES to perform the workload distribution among the candidates ECs with other different approaches (*e.g.*, Voronoi and fully centralized). The results show that Voronoi approach leads to a better QoS, the fully centralized approach minimizes the amount of resources, and PES algorithm reduces the workload imbalances among the candidates ECs. Thus, PES algorithm leads to improvements in terms of request acceptance and resource utilization.

Last, the results show that the PES time complexity exhibits linear dependence with the population density.

Ch7 Traditionally, mobile networks were overdimensioned to cope with the workload expected for next years. Once the processing capacity of a network element was close to its limit (*e.g.,* CPU load of 70%), its hardware was upgraded to meet future needs while maintaining the same software and architectural design. This approach leads to inefficient resource utilization and rigid networks. NFV puts an end to this issue by enabling the autoscaling of softwarized networks.

DRP algorithms are in charge of deciding when to scale a system and how many resources allocate or release to it, while a set of performance requirements are always met. Broadly, a DRP algorithm for network services might consist of the following functional blocks: i) workload predictor, ii) dimensioning algorithm, iii) scaling of the netowrk service, and iv) reactive provisioning module. Additionally, an access control mechanism, which declines excess traffic during temporary overloads, for the network service is required to guarantee that the performance requriments are always met. That is because the reaction time of the DRP procedure before unexpected workload surges is non-neglible.

In this thesis, we have proposed a DRP algorithm for network services that relies on performance modeling. The correctness of the solution have been validated by means of simulations for two use cases:

i) Dynamic provisioning of an End-to-End (E2E) Network Slicing Orchestration System (NSOS).
ii) Dynamic provisioning of the vEPC CP.

The results show that the proposed DRP algorithm finds the optimal solution in terms of both resource saving and processing latency for the two use cases studied. The results also show that the computational complexity of the proposed DRP algorithm is a linear function of the workload.

Finally, the proposed DRP algorithm performs the dimensioning of all the constituent entities of a network service jointly. The results show that compared to the approach of sizing each entity separately, the joint sizing leads to a better utilization of resources.

8.2 Research Contributions

The research contributions resulting from this thesis are listed below:

- A proposal of a softwarized mobile core network architecture that exploits the benefits of SDN and NFV paradigms [44, 48]. Moreover, a feasibility study of this architecture has been carried out in this thesis showing its practicability to become a candidate architecture for the 5G mobile networks.

- An SDN-based implementation of the Handover (HO) procedure using OpenFlow (OF) protocol for partially virtualized mobile networks [48]. Additionally, the impacts of the mobility support on a softwarized mobile network have been listed and analyzed.

- An abstract model for the workload generation process for both CP and Data Plane (DP) in mobile networks has been defined [108,183,184]. Among its applications, this abstract model is of interest to develop realistic mobile workload generators to artificially stress virtualized mobile network for research purposes.

- Analytical expressions has been derived and validated to estimate the signaling workload in mobile networks [108, 183]. The expressions show that the signaling workload depends on the stochastic characteristics of the DP traffic, the user behavior, the RAN setup, and the users' behavior and speed. These expressions allow us to estimate the signaling workload generated per user in a mobile network with agility, circumventing the development of a simulator.

- Definition of two compound traffic models to emulate the traffic demands for the future 5G mobile networks [108, 183, 184].

- The characterization and modeling of the aggregated generation workload processes in mobile networks [184]. We have shown that the aggregated signaling arrival process is roughly Poissonian, whereas the aggregated DP traffic arrival process exhibits Self-Similarity and Long Range Dependence features and it can be modeled as a fractional Brownian Motion process.

- Scalability analysis and evaluation of a vMME with a three-tiered design [108,183].
- An analytical model for assessing the performance of softwarized networks and its experimental validation [111,185]. In this context, a comparison between the different QT methodologies of analysis for network of queues (*e.g.*, MVA algorithm, Jackson networks, and QNA method) in terms of estimation error has been also carried out [111,185].
- An integral solution to perform the planning of a softwarized mobile core network [184]. The solution enables the automation of the deployment of softwarized mobile core networks.
- A solution to carry out the DRP of network services [186,187], which enables the automation of the scaling of network services.

8.3 Future Work

Based on the work carried out in this thesis, several open issues and improvements lie ahead.

1. Derivation of analytical expressions to estimate the parameters of the fBm process that models the aggregated UP traffic in a mobile network given a compound traffic model. The combination of these expression with those ones provided in Chapter 3 to estimate the mean signaling rates, *i.e.*, (3.2)-(3.10), would provide an agile and accurate way to predict aggregated workload in mobile networks without the need for complex and time-consuming simulations.
2. Development of a generic methodology for analyzing performance guarantees in network of queues with feedback. The idea is to provide a mathematical framework capable of estimating a high-order percentile (*e.g.*, 99th) of performance metrics for queuing networks with feedback and considering arbitrary external arrival and service processes.
3. Extension of the DRP solution proposed in Chapter 7:

(a) Design of workload monitoring systems and predictors tailored for future mobile networks.

(b) Inclusion of additional performance requirements in the dimensioning problem such as a minimum service availability or a maximum packet loss probability.

(c) Inclusion of the restrictions imposed by ETSI NFV standards. NFV standards consider a discrete set of instantiation levels among which a network service instance can be resized throughout its lifecycle [164]. These levels are defined at design time and cannot be modified at operation time [164]. This prevents the DRP algorithms from being fully flexible in allocating physical and virtual resources. Moreover, some transitions between instantiation levels might require Virtual Machines (VMs) migrations (i.e., to change the Physical Machine (PM) in which a given VM is hosted). VM migration might be undesirable as it leads to service disruption.

(d) Design of the reactive provisioning module and the admission control procedure of the network service.

4. Formulation and solving of the joint optimization problem of resource dimensioning and embedding. The resource dimensioning and network service embedding problems have been addressed separately through the literature. However, they are coupled as both impact on the E2E delay of the network service. Then, considering both problems jointly might entail some benefits such as resources saving or easing the embedding. The more restrictive the processing delay budget is, the greater the amount of required resources that the dimensioning algorithm will estimate. The more restrictive the propagation delay budget is and the higher the amount of resources to allocate, the more difficult the embedding process will be.

   Additionally, to the best of our knowledge, the NFV affinity constraints have not been considered in the network service embedding solutions proposed in the literature. That is, it might be required that a group of the constituent VNFs of a network service or a set of VNFC instances are deployed on the same PM, resource zone or network Point of Presence (PoP).

Appendices

Appendix A

Resumen

El presente apéndice incluye un amplio resumen en castellano de la memoria de tesis con el objetivo de cumplir con la normativa de la Escuela de Posgrado de la Universidad de Granada referente a la redacción de tesis doctorales cuando éstas son escritas en inglés.

A.1 Introducción y Motivación

Las redes celulares han cambiado radicalmente el modo en el que las personas se comunican. Desde la introducción de las comunicaciones móviles a principios de los cincuenta en Europa, EEUU y Japón, éstas han evolucionado a un ritmo vertiginoso hasta convertirse en las actuales y complejas redes de cuarta generación (4G). Las redes móviles 4G ofrecen altas prestaciones en términos de tasas de transmisión, latencias, seguridad, y soporte a la movilidad de los usuarios. Sin embargo, éstas poseen arquitecturas monolíticas y altamente especializadas para dar soporte a los servicios móviles de banda ancha (MBB -*Mobile Broadband*-), las cuales no son adecuadas para satisfacer la diversidad de requisitos que impondrán los futuros servicios de telecomunicaciones. Además, las redes móviles actuales están basadas en el uso de dispositivos, donde el hardware está verticalmente integrado con el software, que ofrecen una programabilidad limitada o nula. Esto hace que las redes móviles actuales presenten una baja escalabilidad, flexibilidad y elasticidad; y altos costes de despliegue y mantenimiento. Los inconvenientes mencionados han motivado a los organismos de estandarización, la industria de las telecomunicaciones, y la comunidad científica a empezar trabajar en la definición de lo que serán las futuras redes móviles de quinta generación (5G).

En las siguientes subsecciones se ahondará en las principales tendencias y limitaciones de las redes móviles actuales que motivan la definición de las redes móviles 5G. También se presentará el paradigma de softwarización de la red como solución clave para satisfacer las necesidades de las futuras redes móviles 5G.

A.1.1 Motivaciones en la Definición de las Redes Móviles 5G

Las principales tendencias que motivan la definición de las redes móviles 5G se listan a continuación:

- La explosión de la demanda de tráfico en las redes móviles, la cual se espera que sea 20000 veces mayor en 2030 que en 2010. Este crecimiento se deberá principalmente al aumento del número de dispositivos móviles y la aparición de servicios MBB con altas demandas de tráfico como el vídeo 8K y la realidad virtual. Es por ello que se prevé que las futuras redes móviles tendrán que ofrecer una tasa de descarga por usuario de entre 100 Mbps y 1 Gbps.

- La creciente adopción de las comunicaciones masivas de tipo máquina (mMTC -*massive Machine Type Communications*-) cuyo número de dispositivos conectados se espera que alcance los 7000 millones en 2020. Las mMTCs se refiere a un gran número de dispositivos de bajo coste y con baterías de larga duración comunicándose de forma autónoma (sin intervención humana). Este tipo de comunicaciones está típicamente caracterizadas por transmisiones infrecuentes, con baja sensibildad a retardos y con un bajo volumen de datos. Por un lado, las redes móviles actuales no son capaces de soportar la densidad de dispositivos conectados que traerán las mMTCs en el futuro. Por otro lado, las redes móviles actuales no cumplen los requisitos de eficiencia energética requeridos para este tipo de comunicaciones.

- La incesante aparición de nuevos servicios que impondrán requisitos muy estrictos de calidad de servicio para las futuras redes móviles. Entre las nuevas aplicaciones se encuentran la automatización de procesos industriales, la cirugía remota o los vehículos autónomos. Este tipo de aplicaciones requieren una alta fiabilidad de la red (99.999%) y una latencia muy reducida (del orden de 1 ms).

- En el contexto de los servicios de banda ancha se espera dar soporte a escenarios con requisitos más severos en términos de densidad de conexiones, demanda de tráfico, cobertura y movilidad de los usuarios (hasta 500 km/h).

Las tendencias listadas arriba dan lugar a un amplio espectro de casos de uso que presentan requisitos de rendimiento muy diversos, estrictos y a veces conflictivos entre sí. El principal objetivo de las redes móviles 5G es el de dar soporte a todos estos casos de uso empleando una infraestructura de red común. Esto constituye todo un desafío considerando las arquitecturas y enfoques de diseño de las redes móviles actuales.

A.1.2 Limitaciones de las Redes Móviles Actuales

En las redes de móviles actuales, las distintas funciones de red (enrutamiento, cortafuegos, soporte a la movilidad, inspección profunda de paquete, transcodificadores de vídeo, etcétera) se implementan por medio de dispositivos costosos, propietarios y de altas prestaciones donde el hardware y el software están fuertemente acoplados. Este enfoque introduce las siguientes limitaciones:

- CAPEX elevado. Los operadores tienen que hacer fuertes inversiones en infraestructura durante el despliegue de la red para adquirir este hardware especializado y encontrar espacio para acomodarlo. Además, los operadores tienen que esperar que este harware permanezca operativo entre ocho y diez años para asegurar un retorno sobre la inversión razonable. Este problema se ha agravado con la popularización de las tarifas plans que, aunque generó a los operadores un incremento inicial de sus ingresos, conduncen a un estancamiento de los mismos.

- OPEX elevado. Este enfoque implica un elevado consumo de recursos, y requisitos exigentes de energía y refrigeración. Por un lado, los operadores tienen que encargarse de proporcionar energía y refrigeración para el harware que implementa la funcionalidad de la red. Por otro lado, los operadores necesitan personal altamente cualificado para integrar y operar el hardware de red. Las herramientas de gestión de este harware, además de complejas, son específicas del fabricante.

- Baja escalabilidad, flexibilidad y elasticidad. Actualmente las redes móviles son dimensionadas estáticamente y a largo plazo, durante la fase de planificación, para hacer frente a los picos en las demandas de tráfico previstos para los próximos años. Una vez que la carga de trabajo de la red se acerca al limite de capacidad de la misma (p. ej. un 70% de utilización en los recursos de red), el operador escalará la capacidad de este hardware. Este modo de proceder viene impuesto por los altos costes del hardware propietario y por la necesidad de instalar y configurar manualmente nuevo hardware cada vez que la capacidad de la red tiene que ser extendida. Este enfoque presenta las siguientes desventajas:
  - Un desaprovechamiento de recursos, dado que la mayor parte del tiempo la red está sobredimensionada.
  - La centralización de la funcionalidad de la red que incrementa los retardos de propagación.
  - La disponibilidad de la red está comprometida ante un aumento inesperado en la demanda de tráfico.
- Dificultad y lentitud para el despliegue de nuevos servicios e inclusión de nueva funcionalidad en la red.

En las redes de comunicaciones actuales también existe un fuerte acoplamiento entre el plano de control (CP -*Control Plane*-) y el plano de datos (DP -*Data Plane*-). Entendiendo el DP como aquellos dispositivos de red tales como enrutadores y conmutadores que se encargan de dar soporte a la transmisión del tráfico de datos. Mientras que el CP se encarga principalmente de configurar las tablas de encaminamiento de los mencionados dispositivos de red. En las redes actuales el CP está descentralizado y fuertemente acoplado con el DP en cada conmutador y enrutador. Este enfoque presenta tres desventajas principales:

- Incrementa el CAPEX, dado que todos los dispositivos de red tienen que implementar la funcionalidad del plano de control.

- La toma de decisiones para el cumplimiento de las políticas de servicio se tiene que llevar a cabo en todos y cada uno de los dispositivos de red, lo que incrementa los tiempos de procesamiento.

- Dificulta el despliegue de nuevos servicios y la inclusión de nuevas funcionalidades en la red, dando lugar a redes rígidas y estáticas. Esto se debe a que el CP tiene que ser modificado en cada dispositivo de forma individualizada a través de la instalación y configuración de nuevo *firmware*, dando lugar a largos procesos de configuración de dispositivos que son propensos a errores.

A.1.3 Paradigma de Softwarización de la Red

El paradigma de softwarización de la red (NetSoft -*Network Softwarization*-) es un enfoque general para diseñar, implementar, desplegar, gestionar y mantener las redes de comunicaciones por medio de programación *software*. Este paradigma explota las características inherentes del *software* tales como su flexibilidad y su agilidad durante todo el ciclo de vida del equipamiento y/o componentes de red, en aras de crear las condiciones que permitan el re-diseño de la red y las arquitecturas de los servicios, la optimización de los costes y los procesos, la gestión autónoma de la red y la creación de valor añadido en las infraestructuras de red.

Bajo el paradigma NetSoft, es posible desplegar, sobre una infraestructura física común, diferentes particiones de red (*network slices*) aisladas, flexibles, programables, con una gestión y operación completamente automatizadas, y adaptadas a un servicio o conjunto de servicios concretos. A este concepto se le conoce como particionado de red (*network slicing*) y permitirá a los operadores cubrir los diferentes escenarios de mercado y casos de uso definidos para las redes móviles 5G con gran agilidad y costes reducidos. Por otro lado, el paradigma NetSoft acabará con los problemas de flexibilidad, escalabilidad, elasticidad, dificultad para incluir nuevos servicios, y rentabilidad que presentan las redes de comunicaciones actuales.

El paradigma NetSoft está principalmente basado en el uso de las tecnologías NFV (*Network Functions Virtualization* -Virtualización de las Funciones de Red-) y SDN (*Software-Defined Networks* -Redes Definidas por Software-).

Por un lado, la tecnología NFV permite desacoplar las funciones de red del hardware propietario. Las funciones de red (por ejemplo, firewalls, sistemas de detección de intrusiones, pasarelas de red, etcétera) se definen como bloques funcionales dentro de una infraestructura de red que poseen una operación e interfaces externas bien definidas. De este modo, las funciones de red se ejecutan como componentes software, los cuales se conocen como VNFs (*Virtualized Network Functions* -Funciones Virtualizadas de Red-), en servidores genéricos y de bajo coste.

Por otro lado, la tecnología SDN permite la separación completa de los planos de control y datos en los nodos de red, haciendo la red programable. Así, con esta tecnología, el plano de datos se compone de un conjunto de dispositivos de red con la funcionalidad mínima para dar soporte a la recepción, encaminamiento y transmisión de paquetes. Estos dispositivos de red son monitorizados y programados por una entidad externa lógicamente centralizada y conocida como controlador SDN (el cual constituye el plano de control de la red).

A.2 Objetivos

Los objetivos principales de la presente tesis son el estudio de la integración de los paradigmas SDN y NFV en las futuras redes móviles 5G y el diseño de soluciones para la automatización del despliegue y escalado de la red. Para ello, la realización de esta tesis aborda los siguientes objetivos específicos:

1. Diseño de una arquitectura para el núcleo de una red móvil 5G adoptando los paradigmas NFV y SDN. Este objetivo consta de los siguientes subobjetivos:

    1.1 Revisión y definición de casos de uso para 5G.

    1.2 Diseño y evaluación de arquitectura del núcleo de red 5G. Se diseñará una arquitectura de red adaptada a las necesidades de las redes móviles 5G, las cuales vendrán dadas por los casos de uso revisados y definidos previamente. El diseño que se propondrá será jerárquico y distribuido para mejorar la escalabilidad y reducir las latencias de propagación de la red. También se seguirá el principio de subsidiaridad (las decisiones se tomarán siempre en el nivel más bajo o lo más cerca posible de donde tienen efecto) para reducir retardos y evitar tener demasiada carga en las capas de agregación.

1.3 Diseño del soporte a la movilidad para la arquitectura del núcleo de una red móvil 5G.

1.4 Realización de pruebas de concepto que apoyen la viabilidad de la arquitectura en términos de rendimiento.

2. Modelado y evaluación de las prestaciones del núcleo de redes móviles 5G basadas en SDN y NFV. Con este objetivo se pretende desarrollar y evaluar modelos analíticos y de simulación para la estimación de métricas de rendimiento globales en redes basadas en SDN y NFV.

   2.1 Diseño y desarrollo de los modelos. Los modelos matemáticos que se desarrollen estarán basados en teoría de colas y/o cálculo de redes (*Network Calculus*).

   2.2 Validación de los módelos. Para validar dichos modelos se diseñará y desplegará un escenario típico del núcleo de las redes 4G en el que se virtualizarán las distintas entidades de red involucradas y/o se emplearán conmutadores SDN para dar soporte al plano de datos. También es necesario el diseño, desarrollo y ejecución de bancos de pruebas para dicha validación. Los resultados que se obtengan se contrastarán con los resultados predichos por los modelos teóricos, y en caso de ser necesario, se refinarán dichos modelos con el fin de que describan mejor el comportamiento real de los sistemas considerados.

3. Diseño, implementación y evaluación de soluciones basadas en SDN y NFV para la gestión autónoma del núcleo en redes móviles 5G. Usando como punto de partida los modelos teóricos y/o de simulación desarrollados en el objetivo 2 de la presente tesis, en este objetivo se diseñarán y evaluarán soluciones para la gestión autónoma de la red (i.e., automatización de los procesos de despliegue y auto-escalado de la red).

   3.1 Diseño y evaluación de un algoritmo de planificación del núcleo en redes móviles 5G. La solución estimará la cantidad de recursos computacionales, de red, y virtuales necesarios para dar servicio a una determinada zona geográfica y decidirá dónde se emplazarán dichos recursos considerando un conjunto de centros de datos candidatos.

3.2 Diseño y evaluación de un algoritmo de provisión dinámica de recursos. La red será capaz de conocer y adaptará de forma autónoma sus recursos computacionales y de red necesarios para soportar una determinada carga de tráfico de modo que se cumplan los acuerdos de nivel de servicio (SLAs -*Service Level Agreements*-).

A.3 Conclusiones

Ch2 Se espera que los paradigmas NFV y SDN juegen un papel fundamental en las redes móviles futuras debido a que ofrecen un alto grado de programabilidad, flexibilidad, escalabilidad, automatización, y agilidad para desplegar nuevos servicios y funcionalidades de red.

En esta tesis se ha propuesto una arquitectura novedosa para las futuras redes móviles que integra los paradigmas NFV y SDN. A diferencia de las redes móviles actuales y definidas por el 3GPP (*Third Generation Partnership Project*), la arquitectura propuesta emplea túneles MPLS (*Multiprotocol Label Switching*) en vez de túneles GTP (*GPRS Tunneling Protocol*) para dar soporte a la movilidad en el plano de datos. Los túneles GTP introducen una sobrecarga significativa en cada paquete de datos y su gestión puede ser compleja. Además el uso de SDN proporciona flexibilidad a la hora de configurar los puntos finales de los túneles MPLS. De este modo, las comunicaciones internas (aquellas entre terminales pertenecientes a la misma red) no tienen que atravesar la pasarela del núcleo (enrutador que proporciona conectividad hacia las redes externas), el cual es el principal cuello de botella en el DP. Esta característica también posibilita el despliegue de *local breakouts* dentro del núcleo de la red móvil para acceder a servicios locales con una baja latencia.

El soporte a la movilidad de los usuarios, el cual es una de las principales funcionalidades de las redes móviles, ha sido abordado para la arquitectura propuesta. Y se ha definido el procedimiento de traspaso de celdas (*Handover*) para la arquitectura considerando el protocolo OpenFlow para controlar los distintos conmutadores y enrutadores SDN de la red de transporte.

En la presente tesis se han llevado a cabo varias pruebas de concepto para verificar la viabilidad de la arquitectura softwarizada propuesta para redes móviles en términos de rendimiento. Los resultados obtenidos con la realización de las pruebas de concepto apoyan la practicabilidad de los paradigmas NFV y SDN en las futuras redes móviles. Más concretamente, los resultados obtenidos sugieren que:

- Las prestaciones de los conmutadores SDN comerciales son suficientes para soportar los requisitos de las redes móviles 5G.
- La arquitectura softwarizada propuesta puede soportar eficazmente la movilidad de los usuarios, mientras que cumple con los requisitos de latencia para el CP de las redes móviles 5G.

Ch3 Es importante conocer el volumen y las características de la demanda de tráfico que se espera en los próximos años para diseñar y optimizar las futuras redes móviles 5G.

La tecnología 5G dará soporte a las mMTCs las cuales están caracterizadas por un gran número de dispositivos de bajo coste y gran autonomía que transmiten con poca frecuencia mensajes con un bajo volumen de información. A diferencia de los servicios de banda ancha (MBB), el tráfico generado por las mMTCs es altamente homogéneo y puede estar coordinado en pequeñas escalas de tiempo (coordinadamente un gran número de dispositivos genera un mensaje de alarma ante un evento -p. ej. un incendio-). Los procesos de Poisson modulados por cadenas de Markov (MMPPs -*Modulated Markov Poisson Processes*-) son útiles para modelar con precisión las fuentes de tráfico de las mMTCs. Además, la complejidad computacional de estos modelos depende linealmente del número de dispositivos considerado. Curiosamente, los resultados obtenidos en esta tesis demuestran que cada dispositivo de una mMTC genera aproximadamente 3.5 veces más tráfico de señalización que un usuario de servicios de banda ancha. Este resultado sugiere la definición de nuevos procesos de señalización en las redes móviles que consideren la naturaleza del tráfico generado por las mMTCs y que, por tanto, introduzcan una carga de tráfico de control menor en la red y sean energéticamente más eficientes.

En la presente tesis se han derivado y validado expresiones analíticas para estimar el tráfico de señalización generado por los usuarios de servicios móviles de banda ancha. Estas expresiones indican que el tráfico de señalización generado por servicios MBB depende del comportamiento y movilidad del usuario, la densidad y distribución espacial de las estaciones base, el tiempo de expiración del temporizador de inactividad, y las características estocásticas del tráfico del plano de datos.

También se han definido modelos de tráfico compuestos (que integran los servicios más representativos) para simular la demanda de tráfico prevista en las futuras redes móviles 5G. A partir de estos modelos se han estudiado los procesos agregados de generación de tráfico de señalización y de datos en las redes móviles 5G. Los resultados obtenidos sugieren que la generación agregada del tráfico de señalización en las redes móviles aproximadamente obedece a un proceso de Poisson. Mientras que el proceso agregado de generación tráfico de datos exhibe características de autosimilaridad con respecto al tiempo y dependencia a largo plazo y, por ende, podría modelarse con un proceso de movimiento browniano fraccional.

Ch4 La descomposición de una VNF se refiere a que la funcionalidad de dicha VNF puede implementarse de forma distribuida. Es decir, la VNF puede estar compuesta de varios componentes (conocidos como VNFCs) cuyas instancias se ejecutan en máquinas virtuales (VMs -*Virtual Machines*-) o contenedores independientes. Algunos beneficios de la descomposición de una VNF pueden ser una mejor utilización de los recursos computacionales, una mayor robustez de la VNF o facilitar la incrustación de la VNF en la infraestructura subyacente.

Así, por ejemplo, una VNF podría descomponerse de acuerdo a una arquitectura de tres niveles (*three-tiers*) donde los VNFCs serían del tipo: FE (*front-end*), W (*worker*), y DB (*Database*). El nivel FE hace las veces de interfaz externa de la VNF y distribuye la carga entre las instancias de W. El nivel W implementa la lógica de la VNF para el procesamiento de los paquetes. Por último, el nivel DB almacena la información de estado de la

VNF. Este tipo de descomposición de la VNF está inspirada por la arquitectura típica de los servicios Web y presenta numerosas ventajas, como un aumento en la flexibilidad y disponibilidad de la VNF, y una reducción en la complejidad del procedimiento de auto escalado de la VNF. La principal desventaja de este tipo de descomposición es que aumenta el tiempo de respuesta de la VNF, dado que cada paquete tendrá que pasar por varios nodos.

En la presente tesis se ha estudiado la descomposición de tres niveles para un MME virtualizado (vMME -*virtualized Mobility Management Entity*). En la operación considerada para el vMME con tres niveles, el nivel W accede dos veces a la DB por cada paquete que procesa: una vez para recuperar la información de estado necesaria y otra para actualizarla. Aunque esta operación aumenta la carga del nivel DB, al mismo tiempo facilita la lógica de distribución de carga del nivel FE, cuyo escalado horizontal (creación de nuevas instancias del componente) presenta mayor complejidad por actuar como interfaz externa de la VNF.

En el estudio llevado a cabo se ha analizado y evaluado la capacidad y la escalabilidad de un vMME con un diseño de tres niveles. La métrica de escalabilidad usada considera que un sistema es escalable si su productividad (valor entregado por el sistema por unidad de tiempo) crece al mismo ritmo que el coste de su escalado. La productividad del sistema se determina evaluando el rendimiento del sistema una vez que éste ha sido escalado y puede depender de métricas de calidad de servicio como la capacidad, el tiempo de respuesta, la variabilidad del retardo, la disponibilidad y/o la probabilidad de pérdida de paquetes. En nuestro estudio hemos considerado la productividad como una función de la capacidad y el tiempo de respuesta del vMME. Entendiendo la capacidad del vMME como el número de usuarios activos generando señalización en la red móvil que el vMME puede soportar, para una determinada configuración, sin que su tiempo de respuesta supere un cierto umbral.

Se ha propuesto un modelo para evaluar el tiempo de respuesta de un vMME con un diseño de tres niveles basado en una red de colas de Jackson. Este modelo ha sido validado por medio de simulaciones a nivel del sistema. Los resultados de validación sugieren que el modelo de colas es lo suficientemente preciso como para aplicarlo al dimensionamiento de los recursos computacionales requeridos por el vMME. En concreto, el modelo de colas es capaz de estimar la capacidad del vMME con tres niveles para una configuración dada (cantidad de recursos computacionales asignados al vMME) con un error relativo inferior al 5.5%.

En el estudio de escalabilidad se ha considerado el rendimiento de los recursos computacionales y la tarificación ofrecidos por el servicio de infraestructura *Amazon Elastic Compute Cloud* (EC2). Los resultados obtenidos demuestran que el vMME con un diseño de tres niveles es escalable para cargas del plano de control de hasta 37000 procedimientos de señalización por segundo considerando un tiempo medio de respuesta de 1 ms para el vMME. Para evaluar la escalabilidad del vMME más allá de este punto sería necesario definir una estrategia de escalado para el nivel DB (sólo se consideró una única instancia de DB con una capacidad fija).

Los resultados obtenidos en esta tesis demuestran la viabilidad en términos de rendimiento de virtualizar los principales bloques funcionales del plano de control de una red móvil, así como de considerar la descomposición de tres niveles, inpirada por los servicios Web, para implementar los bloques funcionales virtualizados del plano de control de una red móvil.

Ch5 El modelado analítico del rendimiento de las redes de computadores es una metodología ágil para evaluar la calidad de servicio de las mismas. Además, en el contexto de las redes softwarizadas, el modelado del rendimiento tiene dos aplicaciones clave: el provisionamiento dinámico de recursos y la incrustación de la red en una infraestructura subyacente. Estas aplicaciones posibilitan la automatización del despliegue y el escalado de los servicios y funcionalidades de la red.

En la presente tesis se ha propuesto un modelo analítico basado en redes de colas para estimar las métricas de rendimiento de composiciones arbitrarias de VNFs. Para resolver la red de colas resultante se ha empleado el método QNA (*Queuing Network Analyzer*), el cual es una técnica aproximada para estimar las principales métricas de rendimiento de una red de colas G/G/m arbitraria.

En primer lugar, la correcta implementación del modelo propuesto ha sido verificada por medio de simulaciones. En segundo lugar, la precisión del modelo ha sido validada experimentalmente. Para la validación experimental del modelo se consideró un vMME con un diseño de tres niveles. El entorno experimental constaba de un conjunto de máquinas físicas o servidores interconectados por un conmutador 10Gbps Ethernet. Como entorno de virtualización se usó Linux KVM (*Kernel-based Virtual Machine*).

Por último, se ha realizado una comparación entre el método QNA y las técnicas estándar empleadas para resolver redes de colas: MVA (*Mean Value Analysis*) y la metodología para resolver redes de Jackson. Los resultados obtenidos muestran que el método QNA supera a las técnicas estándar en términos de estimación de error para el caso de uso estudiado. Más concretamente, para utilizaciones medias y altas de los recursos computacionales de la VNF, el método QNA ofrece un error de estimación dos veces menor que las metodologías de Jackson y MVA.

Ch6 En el contexto de las redes móviles softwarizadas, las particiones de red (*network slices*) de los distintos verticales necesitan ser planificadas como paso previo a su creación. Además, aunque el paradigma de softwarización de las redas posibilitará a los operadores la adaptación dinámica y automatizada de los recursos asignados a cada partición de red, los planes bajo demanda ofrecidos por los proveedores de infraestructura son más costosos que los planes de reserva de recursos a largo plazo (hasta el 70% de descuento en comparación los planes bajo demanda). Es por ello que el dimensionamiento de las particiones de red durante la fase de planificación de las mismas es importante para que los operadores ahorren dinero.

Los dos problemas principales que han de ser abordados durante la planificación de las particiones de red son el dimensionamiento de los recursos requeridos por las mismas y la reserva de dichos recursos en la infraestructura subyacente (incrustación de la partición de red). Por un lado, el dimensionamiento de los recursos se refiere a la estimación de los recursos computacionales, de red, y virtuales requeridos por la partición de red, de modo que se garantice que la partición de red cumplirá un conjunto de objetivos de rendimiento cuando ésta sea instanciada. Por otro lado, la incrustación de la partición de red está relacionada con la selección de la infraestructura subyacente (cuando por ejemplo hay múltiples *clouds* potenciales para reservar recursos), y las máquinas físicas y los enlaces específicos de la infraestructura de red donde se reservarán los recursos estimados en la fase de dimensionamiento.

En la presente tesis se ha propuesto una solución integral para llevar a cabo la planificación automatizada de un LTE EPC (*Evolved Packet Core*) virtualizado (vEPC). La solución, que ha sido denominada PES (*Planner of the EPC as a Service*), lleva a cabo el dimensionamiento de los recursos de un vEPC y la asignación de los mismos entre un conjunto de *Edge Clouds* candidatos. Para ello, PES usa como entrada un mapa con la densidad de población de la zona geográfica donde el operador pretende dar cobertura y las posiciones de los *Edge Clouds* candidatos. Los principales objetivos del algoritmo de optimización para la planificación del vEPC incluido en PES son:

- Distribución equitativa de la carga entre los *Edge Clouds* candidatos, para así minimizar los desequilibrios de carga.
- Minimizar la cantidad de recursos utilizados.
- Minimizar las latencias del vEPC.

Además, PES garantiza que el vEPC cumplirá con los objetivos considerados para sus métricas de rendimiento. En esta tesis, se han considerado las métricas de rendimiento definidas por los estándares definidos por el 3GPP (*3rd Generation Partnership Project*) para las redes móviles LTE. En concreto, se garantiza que el tiempo de respuesta del plano de control del vEPC (tiempo requerido por el plano de control para mover un usuario de estado inactivo a estado activo -procedimiendo de solicitud de servicio-) está por debajo de un umbral, y un retardo máximo (tiempo que tarda un paquete en llegar al usuario desde la interfaz SGi -interfaz de la pasarela de la red móvil hacia las redes externas-) y probabilidad de pérdida de paquetes maxima para el plano de datos.

Para llevar a cabo el dimensionamiento de los recursos del vEPC, PES utiliza modelos analíticos para estimar el rendimiento del vEPC dada una configuración concreta. Para modelar el plano de control se usa el procedimiento propuesto en el Capítulo 5. Para el modelado del plano de datos, considerando los resultados obtenidos para la caracterización del tráfico de datos agregado en el Capítulo 3, se emplea cálculo de redes (*Network Calculus*). Usando este *framework* teórico, se estima el rendimiento del DP de una red móvil a partir de una red de colas cuyo proceso externo de llegada es un proceso de movimiento browniano fraccional.

Se ha medido la complejidad computacional de PES y se ha concluido que ésta exhibe una dependencia lineal con la densidad de población de la zona de cobertura.

La correcta operación de PES ha sido validada por medio de simulaciones.

Por último, se ha comparado el algoritmo de PES para la distribución de la carga entre los *Edge Clouds* candidatos con otras aproximaciones: Voronoi (la carga se asigna considerando el *Edge Cloud* con menor distancia al usuario) y un enfoque completamente centralizado (se selecciona un único *Edge Cloud*, el que minimiza los retardos de propagación considerando la distribución de los usuarios en la zona de cobertura) . Los resultados muestran que Voronoi minimiza las latencias, mientras que el enfoque centralizado minimiza la cantidad de recursos computacionales requeridos. Por su parte, el algoritmo incluido en PES minimiza los desequilibrios de carga entre los *Edge Clouds* candidatos. Esto se traduce en mejoras en términos de disponibilidad del vEPC y utilización de recursos.

Ch7 Los algoritmos de provisionamiento dinámico de recursos (DRP -*Dynamic Resource Provisioning*- permiten el escalado dinámico y automatizado de los servicios de red, mientras que se garantizan un conjunto de objetivos de rendimiendo para dicho servicio de red. El paradigma de softwarización de la red posibilitará el uso de algoritmos DRP para escalar automáticamente las funcionalidades y servicios de red. Esto implica una mejor utilización de los recursos en comparación con el enfoque usado tradicionalmente donde la red estaba sobre-dimensionada la mayor parte del tiempo. Usando algoritmos de DRP, los recursos de la red son asignados o liberados bajo demanda. De este modo, los operadores pueden lidiar con las fluctuaciones en la carga con agilidad y coste reducido, garantizando que sus servicios de red siempre van a cumplir con los requisitos de rendimiento.

En la presente tesis se ha propuesto una solución para llevar a cabo el DRP de una composición arbitraria de VNFs. Los principales bloques funcionales de la solución de DRP propuesta son:

- Predictor de carga: Este bloque es responsable de estimar la demanda de carga de pico para el servicio de red hasta que se tome la siguiente decisión de provisionamiento de recursos. La lógica de este bloque se ejecuta de manera síncrona cada cierto tiempo.
- Algoritmo de dimensionamiento: Este bloque emplea el modelo de performance propuesto en el Capítulo 5 para estimar la cantidad de recursos necesarios a partir de la estimación de la demanda de tráfico proporcionada por el predictor de carga.
- El manejador del escalado del servicio de red: Este bloque está encargado de disparar los procedimientos requeridos para la reserva o liberación de recursos para el servicio de red.
- Módulo de provisionamiento reactivo: Este bloque dispara peticiones de escalado del servicio de red de forma asíncrona cuando detecta un aumento inesperado en la demanda de tráfico (que no ha sido predicho por el predictor de carga).

La solución de DRP interactúa también con el mecanismo de control de admisión del servicio de red. El mecanismo de control de admisión es el encargado de rechazar el tráfico entrante al servicio de red cuando se dan excesos en la demanda que no estaban previstos. El procedimiento de control de admisión es necesario porque el tiempo de reacción del módulo de provisionamiento reactivo es significativo.

El correcto funcionamiento de la solución de DRP se ha valido por medio de simulaciones para dos casos de uso:

i) Provisionamiento dinámico de un sistema de orquestración de particiones de red extremo a extremo.

ii) Provisionamiento dinámico del plano de control del vEPC.

La complejidad computacional del algoritmo de dimensionamiento y su grado de optimalidad fueron medidos para los casos de estudio arriba mencionados. Los resultados muestran que el algoritmo propuesto de dimensionamiento exhibe una complejidad computacional lineal con la carga y aparentemente ninguna dependencia con el número de entidades que conforman el servicio de red. Respecto a la optimalidad del algoritmo, los resultados indican, que para una carga dada y una métrica de retardo objetivo (expresada como una función de los tiempos de respuesta de cada una de las entidades del servicio de red), el algoritmo encuentra la solución que minimiza la cantidad de recursos mientras se garantiza el requisito de rendimiento. Entendiendo como solución del problema de dimensionamiento la asignación de recursos a cada una de las entidades constituyentes del servicio de red. Además, de entre todas las soluciones que usan la mínima cantidad de recursos posible, el algoritmo encuentra aquella que maximiza la calidad de servicio.

El algoritmo de dimensionamiento del servicio de red propuesto realiza la estimación de los recursos requeridos por cada entidad del servicio de red de forma conjunta. El algoritmo de dimensionamiento ha sido comparado con el enfoque de dimensionar cada entidad de forma individual (el enfoque seguido en la literatura). Los resultados muestran que el dimensionamiento conjunto implica una mejor utilización de los recursos. Este hecho se explica porque el algoritmo distribuye de forma óptima el presupuesto de retardo global para el servicio de red. Mientras que en el enfoque de dimensionamiento individualizado de cada entidad, la distribución del presupuesto de retardo de procesamiento entre las entidades ha de realizarse manualmente y, por tanto, generalmente de forma subóptima.

Bibliography

[1] M. Wang. (2017, May) 5G, When Will We See It? [Online]. Available: https://medium.com/@miccowang/5g-when-will-we-see-it-7c436a4ad86c

[2] *IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond*, Recommendation ITU-R M.2083-0, August 2015.

[3] Open Networking Fundation. (2012, April) Software-Defined Networking: The new norm for networks.

[4] ETSI GS NFV 002 V1.1.1. (2013, October) Network Functions Virtualisation (NFV); Architectural Framework.

[5] S. Sesia, I. Toufik, and M. Baker, *LTE, The UMTS Long Term Evolution: From Theory to Practice.* Wiley Publishing, 2009.

[6] 3GPP TS 36.300 Version 10.12.0. (2014, December) Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2.

[7] 3GPP TS 23.401 Version 10.13.0. (2014, December) General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access.

[8] Netmanias. (2014, January) EMM Procedure 4. S1 Release. NMC Consulting Group. [Online]. Available: https://www.netmanias.com/en/post/techdocs/6110/emm-lte/emm-procedure-3-s1-release. Last accessed: June 2018.

[9] Netmanias. (2014, March) EMM Procedure 6. Handover without TAU - Part 2. X2 Handover. NMC Consulting Group. [Online]. Available: https://www.netmanias.com/en/post/techdocs/6257/emm-handover-lte/emm-procedure-6-handover-without-tau-part-2-x2-handover. Last accessed: June 2018.

[10] M. S. Obaidat, F. Zarai, and P. Nicopolitidis, *Modeling and Simulation of Computer Networks and Systems: Methodologies and Applications*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2015.

[11] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges," *Comm. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.

[12] H. Wang, J. Ding, Y. Li, P. Hui, J. Yuan, and D. Jin, "Characterizing the spatio-temporal inhomogeneity of mobile traffic in large-scale cellular data networks," in *Proceedings of the 7th International Workshop on Hot Topics in Planet-scale mObile Computing and Online Social neTworking*, ser. HOTPOST '15. New York, NY, USA: ACM, 2015, pp. 19–24.

[13] T. Dunnewijk and S. Hultén, "A brief history of mobile communication in europe," *Telematics and Informatics*, vol. 24, no. 3, pp. 164 – 179, 2007, mobile Communications: From Cellular to Ad-hoc and Beyond. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736585307000226

[14] H. Freeman and R. Boutaba, "Networking industry transformation through softwarization [the president's page]," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 4–6, 2016.

[15] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[16] R. Vannithamby and S. Talwar, *Towards 5G: Applications, Requirements and Candidate Technologies.* John Wiley & Sons, 2017.

[17] *5G Vision and Requirements*, White paper, IMT-2020 (5G) Promotion Group, May 2014.

[18] "Use cases and requirements," ICT–317959, Tech. Rep., February 2013.

[19] "5g cellular communications scenarios and system requirements," ICT–GA 318555, Tech. Rep., March 2013.

[20] "Scenarios and requirements," ICT–318784, Tech. Rep., July 2013.

[21] "Definition of scenarios and use cases," ICT–608637, Tech. Rep., December 2013.

[22] "Final definition of ijoin requirements and scenarios," ICT–317941, Tech. Rep., November 2014.

[23] "System scenarios and requirements specifications," ICT–619086, Tech. Rep., January 2015.

[24] "Updated scenarios, requirements and kpis for 5g mobile and wireless system with recommendations for future investigations," ICT–317669 METIS, Deliverable 1.5 Version 1, Tech. Rep., April 2015.

[25] N. Alliance, "5g white paper," Tech. Rep., 2015.

[26] "5g empowering vertical industries," ICT–317941, Tech. Rep., February 2016.

[27] *Feasibility Study on New Services and Markets Technology Enablers; Stage 1*, 3GPP TR22.891 V14.2.0, August 2016.

[28] *Requirements of the IMT-2020 network*, Recommendation ITU-T Y.3101, January 2018.

[29] *Terms and definitions for IMT-2020 network*, Recommendation ITU-T Y.3101, April 2018.

[30] M. Vincenzi, A. Antonopoulos, E. Kartsakli, J. Vardakas, L. Alonso, and C. Verikoukis, "Multi-tenant slicing for spectrum management on the road to 5g," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 118–125, October 2017.

[31] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing & softwarization: A survey on principles, enabling technologies & solutions," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[32] 3GPP, "Study on Architecture for Next Generation System," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.799, 12 2016, version 14.0.0.

[33] O. N. Fundation, "Sdn architecture," *ONF TR-521, Issue 1.1*, 2016.

[34] "Sdn overview," https://www.opennetworking.org/sdn-definition/, accessed: 2018-07-19.

[35] ETSI GS NFV-SWA 001 V1.1.1. (2014, December) Network Functions Virtualisation (NFV); Virtual Network Functions Architecture. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_nfv-swa001v010101p.pdf

[36] *OpenFlow Swicth Specification, version 1.4.0*, Open Networking Foundation (ONF) TS-012, October 2013.

[37] M. Smith, M. Dvorkin, Y. Laribi, V. Pandey, P. Garg, and N. Weidenbacher, "Opflex control protocol," *IETF, Apr*, 2014.

[38] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "Softcell: Scalable and flexible cellular core network architecture," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '13. New York, NY, USA: ACM, 2013, pp. 163–174. [Online]. Available: http://doi.acm.org/10.1145/2535372.2535377

[39] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "Nfv: state of the art, challenges, and implementation in next generation mobile networks (vepc)," *IEEE Network*, vol. 28, no. 6, pp. 18–26, Nov 2014.

[40] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, Firstquarter 2016.

[41] ETSI GS NFV-MAN 001 V1.1.1. (2014, December) Network Functions Virtualisation (NFV); Management and Orchestration. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_nfv-swa001v010101p.pdf

[42] ETSI GS NFV 003 V1.3.1. (2018, January) Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.03.01_60/gs_nfv003v010301p.pdf

[43] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 44–53, July 2013.

[44] P. Ameigeiras *et al.*, "Link-level access cloud architecture design based on SDN for 5G networks," *IEEE Network*, vol. 29, no. 2, pp. 24–31, Mar. 2015.

[45] J. Costa-Requena, J. L. Santos, V. F. Guasch, K. Ahokas, G. Premsankar, S. Luukkainen, O. L. Pérez, M. U. Itzazelaia, I. Ahmad, M. Liyanage, M. Ylianttila, and E. M. de Oca, "Sdn and nfv integration in generalized mobile network architecture," in *2015 European Conference on Networks and Communications (EuCNC)*, June 2015, pp. 154–158.

[46] T. Taleb, M. Corici, C. Parada, A. Jamakovic, S. Ruffino, G. Karagiannis, and T. Magedanz, "EASE: EPC as a service to ease mobile core network deployment over cloud," *IEEE Network*, vol. 29, no. 2, pp. 78–88, 2015.

[47] Y. Kyung *et al.*, "Software defined service migration through legacy service integration into 4G networks and future evolutions," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 108–114, Sept. 2015.

[48] J. Prados-Garzon, O. Adamuz-Hinojosa, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado, and J. M. Lopez-Soler, "Handover implementation in a 5g sdn-based mobile network architecture," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2016, pp. 1–6.

[49] F. Hu, *Opportunities in 5G Networks: A Research and Development Perspective.* CRC Press, 2016.

[50] D. Martín-Sacristán, J. F. Monserrat, J. Cabrejas-Peñuelas, D. Calabuig, S. Garrigas, and N. Cardona, "On the way towards fourth-generation mobile: 3gpp lte and lte-advanced," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 354089, Aug 2009. [Online]. Available: https://doi.org/10.1155/2009/354089

[51] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*, 1st ed. Orlando, FL, USA: Academic Press, Inc., 2011.

[52] H. Holma and A. Toskala, *LTE for UMTS - OFDMA and SC-FDMA Based Radio Access.* Wiley Publishing, 2009.

[53] Netmanias. (2013, September) LTE QoS: SDF and EPS Bearer QoS. NMC Consulting Group. [Online]. Available: https://www.netmanias.com/en/post/techdocs/5908/eps-lte-qos-sdf/lte-qos-sdf-and-eps-bearer-qos. Last accessed: June 2018.

[54] 3GPP TS 23.401 Rel 12. (2014) General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access.

[55] *OpenFlow Swicth Specification, version 1.4.0*, Open Networking Foundation (ONF) TS-012, October 2013.

[56] N. Varis, J. Manner, and J. Heinonen, "A layer-2 approach for mobility and transport in the mobile backhaul," in *2011 11th International Conference on ITS Telecommunications*, Aug 2011, pp. 268–273.

[57] C. Sieber, R. Durner, M. Ehm, W. Kellerer, and P. Sharma, "Towards optimal adaptation of nfv packet processing to modern cpu memory architectures," in *Proceedings of the 2Nd Workshop on Cloud-Assisted Networking*, ser. CAN '17. New York, NY, USA: ACM, 2017, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/3155921.3158429

[58] G. P. Katsikas, G. Q. M. Jr., and D. Kostić, "Profiling and accelerating commodity nfv service chains with scc," *Journal of Systems*

and Software, vol. 127, pp. 12 – 27, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121217300055

[59] M. S. Guideline, “METIS Deliverable D6. 1 Simulation Guidelines,” 2013.

[60] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. A. Uusitalo, B. Timus, and M. Fallgren, “Scenarios for 5g mobile and wireless communications: the vision of the metis project,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, May 2014.

[61] P. Rygielski, M. Seliuchenko, S. Kounev, and M. Klymash, “Performance analysis of sdn switches with hardware and software flow tables,” in *Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools on 10th EAI International Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS’16. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2017, pp. 80–87. [Online]. Available: https://doi.org/10.4108/eai.25-10-2016.2266540

[62] D. Stiliadis and A. Varma, “Latency-rate servers: A general model for analysis of traffic scheduling algorithms,” *IEEE/ACM Trans. Netw.*, vol. 6, no. 5, pp. 611–624, Oct. 1998. [Online]. Available: http://dx.doi.org/10.1109/90.731196

[63] G. F. Riley and T. R. Henderson, “The ns-3 Network Simulator,” in *Modeling and Tools for Network Simulation.* Springer, 2010.

[64] R. Nadiv and T. Naveh, “Wireless backhaul topologies: Analyzing backhaul topology strategies,” *Ceragon White Paper*, pp. 1–15, 2010.

[65] F. C. Kuo, F. A. Zdarsky, J. Lessmann, and S. Schmid, “Cost-efficient wireless mobile backhaul topologies: An analytical study,” in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Dec 2010, pp. 1–5.

[66] D. Han, S. Shin, H. Cho, J. m. Chung, D. Ok, and I. Hwang, “Measurement and stochastic modeling of handover delay and interruption time of

smartphone real-time applications on lte networks," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 173–181, March 2015.

[67] R. R. Roy, *Handbook of mobile ad hoc networks for mobility models.* Springer Science & Business Media, 2010.

[68] Ceragon, "Wireless Backhaul Topologies: Analyzing Backhaul Topology Strategies," pp. 1–15, August 2010. [Online]. Available: http://www.winncom.com/images/stories/Ceragon_Wireless_\\

[69] D. Singhal, M. Kunapareddy, V. Chetlapalli, V. B. James, and N. Akhtar, "LTE-advanced: handover interruption time analysis for IMT-A evaluation," in *2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN).* IEEE, 2011, pp. 81–85.

[70] T. Liu, "Implementing OpenFlow switch using FPGA based platform," in *Master Thesis, Dept. of Telematics, Norwegian University of Sci. and Technology*, June 2014.

[71] F. Dürr and T. Kohler, "Comparing the Forwarding Latency of OpenFlow Hardware and Software Switches," Tech. Rep. Comput. Sci. 2014/04, University of Stuttgart, Institute of Parallel and Distributed Systems (IPVS), Tech. Rep., 2014.

[72] H. Uppal and D. Brandon, "OpenFlow based load balancing," *CSE561: Networking Project Report, University of Washington*, 2010.

[73] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, "On controller performance in software-defined networks," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, ser. Hot-ICE'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 10–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228283.2228297

[74] Z. Li and M. Wilson, "User plane and control plane separation framework for home base stations," *Fujitsu Scientific and Tech. J.*, vol. 46, no. 1, pp. 79–86, 2010.

[75] Y. Takano, A. Khan, M. Tamura, S. Iwashina, and T. Shimizu, "Virtualization-Based Scaling Methods for Stateful Cellular Network Nodes Using Elastic Core Architecture," in *IEEE 6th Int. Conf. on Cloud Computing Technology and Science (CloudCom)*, 2014, pp. 204–209.

[76] G. Premsankar, K. Ahokas, and S. Luukkainen, "Design and Implementation of a Distributed Mobility Management Entity on OpenStack," in *IEEE 7th Int. Conf. on Cloud Computing Technology and Science (CloudCom)*, 2015, pp. 487–490.

[77] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "Reduced m2m signaling communications in 3gpp lte and future 5g cellular networks," in *2016 Wireless Days (WD)*, March 2016, pp. 1–3.

[78] I. N. Oteyo and E. Bainomugisha, "Volume of signaling traffic reaching cellular networks from mobile phones," in *2017 IEEE AFRICON*, Sept 2017, pp. 831–836.

[79] J. Navarro-Ortiz, S. Sendra, P. Romero, and J. M. Lopez-Soler, "A survey on 5g usage scenarios and traffic models," *Submitted to Wireless Communications and Mobile Computing*, 2018.

[80] "2016 global internet phenomena: Latin america & north america," Sandvine, Tech. Rep., 2016. [Online]. Available: https://www.sandvine.com/hubfs/downloads/archive/2016-global-internet-phenomena-report-latin-america-and-north-america.pdf

[81] NGMN Alliance, "Radio Access Performance Evaluation Methodology," January 2008.

[82] G.-f. Zhao, Q. Shan, S. Xiao, and C. Xu, "Modeling Web Browsing on Mobile Internet," *IEEE Communications Letters*, vol. 15, no. 10, pp. 1081–1083, October 2011.

[83] A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proceedings of*

the Seventh COnference on Emerging Networking EXperiments and Technologies, ser. CoNEXT '11. New York, NY, USA: ACM, 2011, pp. 25:1–25:12. [Online]. Available: http://doi.acm.org/10.1145/2079296.2079321

[84] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Analysis and modelling of YouTube traffic," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 4, pp. 360–377, June 2012.

[85] J. J. Ramos-Muñoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. M. López-Soler, "Characteristics of mobile youtube traffic," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 18–25, 2014.

[86] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Characterizing User Sessions on YouTube," in *Proc. SPIE 6818, Multimedia Computing and Networking 2008, 681806*, January 2008.

[87] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, "Detailed analysis of skype traffic," *IEEE Transactions on Multimedia*, vol. 11, no. 1, pp. 117–127, Jan 2009.

[88] T. D. Dang, B. Sonkoly, and S. Molnar, "Fractal analysis and modeling of voip traffic," in *11th International Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004,*, June 2004, pp. 123–130.

[89] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing user behavior in online social networks," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '09. New York, NY, USA: ACM, 2009, pp. 49–62.

[90] X. Zhou, Z. Zhao, R. Li, Y. Zhou, J. Palicot, and H. Zhang, "Understanding the nature of social mobile instant messaging in cellular networks," *IEEE Communications Letters*, vol. 18, no. 3, pp. 389–392, March 2014.

[91] M. Laner, N. Nikaein, P. Svoboda, M. Popovic, D. Drajic, and S. Krco, "8 - traffic models for machine-to-machine (m2m) communications: types and

applications," in *Machine-to-machine (M2M) Communications*, C. Antón-Haro and M. Dohler, Eds. Oxford: Woodhead Publishing, 2015, pp. 133 – 154.

[92] F. Li, X. f. Chi, and J. s. Zhang, "Flow characteristics analysis and modeling of m2m service," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 2, May 2012, pp. 459–463.

[93] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," in *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*, Aug 2013, pp. 1–5.

[94] T. Ali-Yahiya, *Understanding LTE and Its Performance*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[95] 3GPP, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.331, 09 2014, version 10.15.0.

[96] Netmanias. (2014, February) EMM Procedure 4. Service Request. NMC Consulting Group. [Online]. Available: https://www.netmanias.com/en/post/techdocs/6134/emm-lte/emm-procedure-4-service-request. Last accessed: June 2018.

[97] B. Hirschman, P. Mehta, K. B. Ramia, A. S. Rajan, E. Dylag, A. Singh, and M. Mcdonald, "High-performance evolved packet core signaling and bearer processing on general-purpose processors," *IEEE Network*, vol. 29, no. 3, pp. 6–14, May 2015.

[98] T. Deng, X. Wang, P. Fan, and K. Li, "Modeling and performance analysis of a tracking-area-list-based location management scheme in lte networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6417–6431, Aug 2016.

[99] HTTP Archive. Interesting stats. [Online]. Available: http://httparchive.org/interesting.php. Last accessed: July 2015.

[100] I. Tsompanidis, A. H. Zahran, and C. J. Sreenan, “Mobile network traffic: A user behaviour model,” in *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, IEEE. IEEE, May 2014, pp. 1–8.

[101] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, “Reduced M2M signaling communications in 3GPP LTE and future 5G cellular networks,” in *2016 Wireless Days (WD)*, March 2016, pp. 1–3.

[102] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, “Optimized LTE data transmission procedures for IoT: Device side energy consumption analysis,” in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 540–545.

[103] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. Navarro-Ortiz, and J. M. Lopez-Soler, “Narrowband IoT Data Transmission Procedures for Massive Machine-Type Communications,” *IEEE Network*, vol. 31, no. 6, pp. 8–15, November 2017.

[104] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of ethernet traffic,” *SIGCOMM Comput. Commun. Rev.*, vol. 23, no. 4, pp. 183–193, Oct. 1993. [Online]. Available: http://doi.acm.org/10.1145/167954.166255

[105] I. Norros, “On the use of fractional brownian motion in the theory of connectionless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 953–962, Aug 1995.

[106] A. S. Rajan, S. Gobriel, C. Maciocco, K. B. Ramia, S. Kapury, A. Singhy, J. Ermanz, V. Gopalakrishnanz, and R. Janaz, “Understanding the bottlenecks in virtualizing cellular core network functions,” in *The 21st IEEE International Workshop on Local and Metropolitan Area Networks*, April 2015, pp. 1–6.

[107] Simon Dredge. Taking Out the Trash: The Decomposition of Virtualized Network Functions. [Online]. Available:

https://www.metaswitch.com/blog/taking-out-the-trash-the-decomposition-of-virtualized-network-functions. Last accessed: June 2018.

[108] J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado, and J. M. Lopez-Soler, "Modeling and dimensioning of a virtualized MME for 5G mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4383–4395, 2017.

[109] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "MME support for M2M communications using network function virtualization," in *Proc. IEEE 12th Adv. Int. Conf. Telecommun.(AICT)*, 2016, pp. 106–111.

[110] Pilar Andres-Maldonado, Pablo Ameigeiras, Jonathan Prados-Garzon, Juan Jose Ramos-Munoz, and Juan Manuel Lopez-Soler, "Virtualized MME Design for IoT Support in 5G Systems," *Sensors*, vol. 16, no. 8, 2016.

[111] J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado, and J. M. Lopez-Soler, "Analytical modeling for virtualized network functions," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 979–985.

[112] G. Carella, M. Corici, P. Crosta, P. Comi, T. M. Bohnert, A. A. Corici, D. Vingarzan, and T. Magedanz, "Cloudified IP multimedia subsystem (IMS) for network function virtualization (NFV)-based architectures," in *2014 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2014, pp. 1–6.

[113] Project Clearwater. [Online]. Available: http://www.projectclearwater.org/. Last accessed: June 2018.

[114] S. Bose, *An Introduction to Queueing Systems.* Springer, 2014.

[115] J. Tate, N. Bogard, M. Holenia, S. Oglaza, and S. Tong, *IBM b-type Data Center Networking: Design and Best Practices Introduction.* IBM Redbooks, December 2010.

[116] A. K. Erlang, "Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges," *Electroteknikeren*, vol. 13, 1917.

[117] P. Jogalekar and M. Woodside, "Evaluating the scalability of distributed systems," *IEEE Transactions on parallel and distributed systems*, vol. 11, no. 6, pp. 589–603, June 2000.

[118] Amazon Web Services. Amazon EC2 Instances. [Online]. Available: http://aws.amazon.com/ec2/instance-types/. Last accessed: September 2015.

[119] B. Adler. (2010, March) White Paper - Load Balancing in the Cloud. RIGHTSCALE. [Online]. Available: https://www.yumpu.com/en/document/view/2675240/load-balancing-in-the-cloud-tools-tips-and-techniques. Last accessed: September 2015.

[120] Amazon Web Services Inc. (2015) Amazon Aurora Performance Assessment. [Online]. Available: http://d0.awsstatic.com/product-marketing/Aurora/RDS_Aurora_Performance_Assessment_Benchmarking_v1-2.pdf. Last accessed: September 2015.

[121] M. A. Gray, "Discrete event simulation: A review of simevents," *Computing in Science Engineering*, vol. 9, no. 6, pp. 62–66, Nov 2007.

[122] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed systems*, vol. 22, no. 6, pp. 931–945, 2011.

[123] Z. Li and M. Wilson, "User plane and control plane separation framework for home base stations," *Fujitsu scientific and technical journal*, vol. 46, no. 1, pp. 79–86, 2010.

[124] Z. Savic, "Lte design and deployment strategies," *Cisco*, 2011.

[125] 3GPP, "5G; Study on Scenarios and Requirements for Next Generation Access Technologies," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.913, 3 2017, version 14.2.0.

[126] S. Chaisiri, B. S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 164–177, April 2012.

[127] Amazon Web Services. Amazon EC2 Pricing. [Online]. Available: https://aws.amazon.com/ec2/pricing/. Last accessed: July 2018.

[128] Ericsson, "Characteristics. TECHNICAL PRODUCT DESCRIPTION," 43/221 02-AXB 250 05/8 Uen BK, April 2012.

[129] I. Chana and S. Singh, *Quality of Service and Service Level Agreements for Cloud Environments: Issues and Challenges.* Cham: Springer International Publishing, 2014, pp. 51–72. [Online]. Available: https://doi.org/10.1007/978-3-319-10530-7_3

[130] L. Kleinrock, *Theory, Volume 1, Queueing Systems.* New York, NY, USA: Wiley-Interscience, 1975.

[131] D. Bertsekas and R. Gallager, *Data Networks (2Nd Ed.).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

[132] S. K. Bose, *An Introduction to Queuing Systems.* 233 Spring Street, New York, New York 10013-1578: Kluwer Academic / Plenum Publishers, 2002.

[133] H. Chen and D. D. Yao, *Fundamentals of queueing networks: Performance, asymptotics, and optimization.* Springer Science & Business Media, 2013, vol. 46.

[134] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet.* Berlin, Heidelberg: Springer-Verlag, 2001.

[135] Y. Jiang and Y. Liu, *Stochastic network calculus.* Springer, 2008, vol. 1.

[136] M. Fidler and A. Rizk, "A guide to the stochastic network calculus," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 92–105, Firstquarter 2015.

[137] Y. Ren, T. Phung-Duc, J. Chen, and Z. Yu, "Dynamic auto scaling algorithm (dasa) for 5g mobile networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[138] K. TANABE, H. NAKAYAMA, T. HAYASHI, and K. YAMAOKA, "vepc optimal resource assignment method for accommodating m2m communications," *IEICE Transactions on Communications*, vol. advpub, 2017.

[139] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, April 2015, pp. 1–9.

[140] J. R. Jackson, "Networks of waiting lines," *Oper. Res.*, vol. 5, no. 4, pp. 518–521, Aug. 1957. [Online]. Available: http://dx.doi.org/10.1287/opre.5.4.518

[141] W. J. Gordon and G. F. Newell, "Closed queuing systems with exponential servers," *Oper. Res.*, vol. 15, no. 2, pp. 254–265, Apr. 1967. [Online]. Available: http://dx.doi.org/10.1287/opre.15.2.254

[142] M. Reiser and S. S. Lavenberg, "Mean-value analysis of closed multichain queuing networks," *J. ACM*, vol. 27, no. 2, pp. 313–322, Apr. 1980. [Online]. Available: http://doi.acm.org/10.1145/322186.322195

[143] J. P. Buzen, "Computational algorithms for closed queueing networks with exponential servers," *Commun. ACM*, vol. 16, no. 9, pp. 527–531, Sep. 1973. [Online]. Available: http://doi.acm.org/10.1145/362342.362345

[144] S. Bruell, G. Balbo, and P. Afshari, "Mean value analysis of mixed, multiple class bcmp networks with load dependent service stations," *Performance Evaluation*, vol. 4, no. 4, pp. 241 – 260, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0166531684900105

[145] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, April 1975. [Online]. Available: http://doi.acm.org/10.1145/321879.321887

[146] D. R. Cox, “A use of complex probabilities in the theory of stochastic processes,” in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 2. Cambridge University Press, 1955, pp. 313–319.

[147] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, “Analytic modeling of multitier internet applications,” *ACM Trans. Web*, vol. 1, no. 1, May 2007. [Online]. Available: http://doi.acm.org/10.1145/1232722.1232724

[148] J. Bi, Z. Zhu, R. Tian, and Q. Wang, “Dynamic provisioning modeling for virtualized multi-tier applications in cloud data center,” in *IEEE 3rd Int. Conf. on Cloud Computing (CloudCom)*, 2010, pp. 370–377.

[149] G. Faraci, A. Lombardo, and G. Schembra, “A building block to model an SDN/NFV network,” in *Communications (ICC), 2017 IEEE International Conference on.* IEEE, 2017, pp. 1–7.

[150] S. Gebert, T. Zinner, S. Lange, C. Schwartz, and P. Tran-Gia, “Performance modeling of softwarized network functions using discrete-time analysis,” in *Teletraffic Congress (ITC 28), 2016 28th International*, vol. 1. IEEE, 2016, pp. 234–242.

[151] Q. Duan, “Modeling and performance analysis for composite network–compute service provisioning in software-defined cloud environments,” *Digital Communications and Networks*, vol. 1, no. 3, pp. 181 – 190, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352864815000383

[152] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, and D. Simeonidou, “An analytical model for software defined networking: A network calculus-based approach,” in *Global Communications Conference (GLOBECOM), 2013 IEEE.* IEEE, 2013, pp. 1397–1402.

[153] A. K. Koohanestani, A. G. Osgouei, H. Saidi, and A. Fanian, “An analytical model for delay bound of openflow based sdn using network calculus,” *Journal of Network and Computer Applications*, vol. 96, pp. 31 – 38, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517302485

[154] G. Motika and S. Weiss, "Virtio network paravirtualization driver: Implementation and performance of a de-facto standard," *Computer Standards & Interfaces*, vol. 34, no. 1, pp. 36–47, 2012.

[155] M. Bux and U. Leser, "Dynamiccloudsim: Simulating heterogeneity in computational clouds," *Future Generation Computer Systems*, vol. 46, pp. 85–99, 2015.

[156] W. Whitt, "The queueing network analyzer," *Bell System Tech. J.*, vol. 62, no. 9, pp. 2779–2815, Nov. 1983.

[157] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An analytical model for multi-tier internet services and its applications," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 291–302, Jun. 2005. [Online]. Available: http://doi.acm.org/10.1145/1071690.1064252

[158] J. A. Kreibich, *Using SQLite*, 1st ed. O'Reilly Media, Inc., 2010.

[159] H. D. Chirammal, P. Mukhedkar, and A. Vettathu, *Mastering KVM Virtualization.* Packt Publishing Ltd, 2016.

[160] C. Gough, I. Steiner, and W. Saunders, *Energy efficient servers: blueprints for data center optimization.* Apress, 2015.

[161] M. Marzolla, "The Qnetworks Toolbox: A Software Package for Queueing Networks Analysis," in *Proc. 17th Int. Conf. on Analytical and Stochastic Modeling Techniques and Applications (ASMTA).* Berlin, Heidelberg: Springer-Verlag, 2010, pp. 102–116.

[162] 3GPP TR 28.801 Version 15.1.0. (2018, January) Telecommunication management; Study on management and orchestration of network slicing for next generation network (Release 15).

[163] W. Haeffner, J. Napper, M. Stiemerling, D. Lopez, and J. Uttaro, "Service function chaining use cases in mobile networks," *Internet Engineering Task Force*, 2016.

[164] O. Adamuz-Hinojosa, J. Ordonez-Lucena, P. Ameigeiras, J. J. Ramos-Munoz, D. Lopez, and J. Folgueira, "Automated network service scaling

in nfv: Concepts, mechanisms and scaling workflow," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 162–169, JULY 2018.

[165] K. Tanabe, H. Nakayama, T. Hayashi, and K. Yamaoka, "An optimal resource assignment for c/d-plane virtualized mobile core networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[166] T. Taleb and A. Ksentini, "Gateway relocation avoidance-aware network function placement in carrier cloud," in *Proceedings of the 16th ACM International Conference on Modeling, Analysis & Simulation of Wireless and Mobile Systems*, ser. MSWiM '13. New York, NY, USA: ACM, 2013, pp. 341–346. [Online]. Available: http://doi.acm.org/10.1145/2507924.2508000

[167] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, pp. 2402–2407.

[168] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying nfv and sdn to lte mobile core gateways, the functions placement problem," in *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, ser. AllThingsCellular '14. New York, NY, USA: ACM, 2014, pp. 33–38. [Online]. Available: http://doi.acm.org/10.1145/2627585.2627592

[169] T. Taleb, M. Bagaa, and A. Ksentini, "User mobility-aware virtual network function placement for virtual 5g network infrastructure," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 3879–3884.

[170] B. Martini, F. Paganelli, P. Cappanera, S. Turchi, and P. Castoldi, "Latency-aware composition of virtual functions in 5g," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, April 2015, pp. 1–6.

[171] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Combined virtual mobile core network function placement and topology optimization with latency bounds," in *2015 Fourth European Workshop on Software Defined Networks*, Sept 2015, pp. 97–102.

[172] A. Laghrissi, S. Retal, and A. Idrissi, "Modeling and optimization of the network functions placement using constraint programming," in *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, ser. BDAW '16. New York, NY, USA: ACM, 2016, pp. 52:1–52:8. [Online]. Available: http://doi.acm.org/10.1145/3010089.3010137

[173] D. Dietrich, C. Papagianni, P. Papadimitriou, and J. S. Baras, "Near-optimal placement of virtualized epc functions with latency bounds," in *Communication Systems and Networks*, N. Sastry and S. Chakraborty, Eds. Cham: Springer International Publishing, 2017, pp. 200–222.

[174] D. Lee, S. Zhou, X. Zhong, Z. Niu, X. Zhou, and H. Zhang, "Spatial modeling of the traffic density in cellular networks," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 80–88, 2014.

[175] D. Lee, S. Zhou, and Z. Niu, "Spatial modeling of scalable spatially-correlated log-normal distributed traffic inhomogeneity and energy-efficient network planning," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 1285–1290.

[176] R. Mijumbi, S. Hasija, S. Davy, A. Davy, B. Jennings, and R. Boutaba, "A connectionist approach to dynamic resource management for virtualised network functions," in *Network and Service Management (CNSM), 2016 12th International Conference on*. IEEE, 2016, pp. 1–9.

[177] D. Huang, B. He, and C. Miao, "A survey of resource management in multi-tier web applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1574–1590, Third 2014.

[178] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood, "Agile dynamic provisioning of multi-tier internet applications," *ACM Trans.*

Auton. Adapt. Syst., vol. 3, no. 1, pp. 1:1–1:39, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1342171.1342172

[179] R. Mijumbi, S. Hasija, S. Davy, A. Davy, B. Jennings, and R. Boutaba, "Topology-aware prediction of virtual network function resource requirements," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 106–120, March 2017.

[180] C. H. T. Arteaga, F. Rissoi, and O. M. C. Rendon, "An adaptive scaling mechanism for managing performance variations in network functions virtualization: A case study in an nfv-based epc," in *2017 13th International Conference on Network and Service Management (CNSM)*, Nov 2017, pp. 1–7.

[181] I. Afolabi, A. Ksentini, M. Bagaa, T. Taleb, M. Corici, and A. Nakao, "Towards 5G network slicing over multiple domains," *IEICE Transactions on Communications, Special section on Network Virtualization, Network Softwarisation, and Fusion Platform of Computing and Networking*, vol. 100B, no. 11, 11 2017. [Online]. Available: http://www.eurecom.fr/publication/5375

[182] "5g!pagoda project," https://5g-pagoda.aalto.fi/, accessed: 2018-07-19.

[183] J. Prados-Garzon, J. J. Ramos-Munoz, P. Ameigeiras, P. Andres-Maldonado, and J. M. Lopez-Soler, "Latency evaluation of a virtualized mme," in *2016 Wireless Days (WD)*, March 2016, pp. 1–3.

[184] J. Prados-Garzon, A. Laghrissi, M. Bagaa, T. Taleb, and J. M. Lopez-Soler, "A Complete LTE Mathematical Framework for the Network Slice Planning of the EPC," *Submitted to IEEE Transactions on Mobile Computing, 2018.*

[185] J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, P. Andres-Maldonado, and J. M. Lopez-Soler, "Performance Modeling of Chains of Virtualized Network Functions Based on Queuing Theory with Experimental Validation," *Submitted to IEEE/ACM Transactions on Networking, 2018.*

[186] J. Prados-Garzon, A. Laghrissi, M. Bagaa, and T. Taleb, "A queuing based dynamic auto scaling algorithm for the lte epc control plane," in *Accepted on 2018 IEEE Global Communications Conference (GLOBECOM 2018)*, Abu Dhabi, UAE, Dec. 2018.

[187] I. Afolabi, J. Prados-Garzon, M. Bagaa, and T. Taleb, "Modeling and Dynamic Provisioning of a Scalable E2E Network Slicing Orchestration System," *Submitted to IEEE Transactions on Mobile Computing, 2018.*

This thesis has been supported by the following projects and grants:

- Programa de Formación de Profesorado Universitario (FPU) grant from the Spanish Ministry of Education, Culture and Sport (FPU Grant ref. **13/04833**).
- Ayuda Complementaria FPU para Estancias Breves y Traslados Temporales grant from the Spanish Ministry of Education, Culture and Sport (ref. **EST16/00751**).
- National Research Project **TIN2013-46223-P** "Arquitectura para Redes Móviles 5G basada en Software Defined Networks" ("Architecture for 5G Mobile Networks based on Software Defined Networks") funded by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund.
- National Research Project **TEC2016-76795-C6-4-R** "5G-City: Gestión Flexible de Servicios 5G Orientada a Soportar Situaciones Críticas Urbanas" ("5G-City: Adaptive Management of 5G Services to Support Critical Events in Cities") funded by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund.