12-13-2019 9:50 AM

Leveraging Cloud-based NFV and SDN Platform Towards Quality-Driven Next-Generation Mobile Networks

Hassan Hawilo, *The University of Western Ontario*

Supervisor: Shami, Abdallah, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical and Computer Engineering

Abstract

Network virtualization has become a key approach for Network Service Providers (NSPs) to mitigate the challenge of the continually increasing demands for network services. Tightly coupled with their software components, legacy network devices are difficult to upgrade or modify to meet the dynamically changing end-user needs. To virtualize their infrastructure and mitigate those challenges, NSPs have started to adopt Software Defined Networking (SDN) and Network Function Virtualization (NFV). To this end, this thesis addresses the challenges faced on the road of transforming the legacy networking infrastructure to a more dynamic and agile virtualized environment to meet the rapidly increasing demand for network services and serve as an enabler for key emerging technologies such as the Internet of Things (IoT) and 5G networking. The thesis considers different approaches and platforms to serve as NFV/SDN based cloud applications while closely considering how such an environment deploys its virtualized services to optimize the network and reducing their costs. The thesis starts first by defining the standards of adopting microservices as architecture for NFV. Then, it focuses on the latency-aware deployment approach of virtual network functions (VNFs) forming service function chains (SFC) in a cloud environment. This approach ensures that NSPs still meet their strict quality of service and service level agreements while considering both functional and non-functional constraints of the NFV-based applications such as delay, resource allocation, and intercorrelation between VNF instances. In addition, the thesis proposes a detailed approach on recovering and handling those instances by optimizing the decision of migrating or re-instantiating the virtualized services upon a sudden event (failure/overload…). All the proposed approaches contribute to the orchestration of NFV applications to meet the requirements of the IoT and NGNs era.

Keywords: Network function visualization, Software-defined networking, Quality of Service, Latency-aware Placement, Service Function Chaining, Microservices, Cloud computing, Scheduling, Migration, Re-instantiation, Redundancy, Interdependency, Computational Path.

Summary

The demand for broadband network connectivity has been increasing dramatically in the last decade. It gains additional momentum with the increase in the number of Internet-connected mobile devices, ranging from smartphones, tablets, and laptops to sensor networks, and machine-to-machine (M2M) connectivity. Although studies show that the return on such investments is minimal, this increasing demand is pushing network service providers to invest in infrastructure to keep up with the demand. Network expenditures depend highly on the infrastructure on which the network relies. The high cost of any network-improvement upgrade or new service release narrows the revenue margin of the service provider. Network operating challenges are not limited to the cost of expensive hardware devices, but also include increasing energy costs and the competitive market for highly qualified personnel with the skills necessary to design, integrate, and operate an increasingly complex hardware-based infrastructure. In addition, managing network infrastructure is another major concern of service providers. These issues do not affect revenue only, but they also increase time-to-market and limit innovation in the telecommunications industry. Therefore, network operators seek to minimize or even eliminate their dependence on proprietary hardware.

To achieve these targets, network service operators are investigating the integration of virtualization technology within the telecommunications industry. Virtualization technology emerged as a mean for information technology (IT) specialists to achieve more effective capital investments with higher returns on capital.

In this research work, the next generation of mobile core network entities are assessed and reengineered to be adopted as a virtualized entity suitable for cloud environment deployment. The virtualization process through network function virtualization (NFV) and software-defined

networking (SDN) will pave the way to adopt and orchestrate heterogeneous network for better utilization of resources (wireless and computing resources) while migrating the benefits of cloud computing era to the telecommunication industry such as scaling on-demand, pay-as-you-go, and providing resources as services.

Co-Authorship

This thesis contains the following manuscripts that have been submitted, accepted, and published.

1. H. Hawilo, M. Jammal and A. Shami, "Orchestrating network function virtualization platform: Migration or re-instantiation?," *IEEE 6th International Conference on Cloud Networking (CloudNet)*, Prague, 2017, pp. 1-6.
2. H. Hawilo, L. Liao, A. Shami and V. C. M. Leung, "NFV/SDN-based vEPC solution in hybrid clouds," *IEEE Middle East and North Africa Communications Conference (MENACOMM)*, 2018, pp. 1-6.
3. H. Hawilo, M. Jammal and A. Shami, "Network Function Virtualization-Aware Orchestrator for Service Function Chaining Placement in the Cloud," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 643-655, March 2019.
4. H. Hawilo, M. Jammal and A. Shami, "Exploring Microservices as the Architecture of Choice for Network Function Virtualization Platforms," in *IEEE Network*, vol. 33, no. 2, pp. 202-210, March/April 2019.
5. H. Hawilo, D.M. Manias, M. Jammal and A. Shami, "Dynamic Orchestration of NFV-based Applications using Migration or Re-instantiation," *Submitted to IEEE Transactions on Network and Service Management*, October 2019.

The following co-authors provided experimental and technical support for the studies listed above:

- A. Shami supervised the development of the work and provided technical expertise, opinion, and perspective based on his experience as a professor at Western University. He supervised the work done in this thesis.
- M. Jammal provided technical expertise on the design of the optimization model and heuristic solutions and helped in the manuscript preparation and review. She contributed to the work done in Chapter 2, Chapter 4, and Chapter 5.
- L. Liao provided technical expertise on the model design and has contributed greatly to defining the challenges faced in virtualizing the evolved packet core. She contributed to the work done in Chapter 3.
- V. C. M. Leung supervised the development of the work and provided technical expertise, opinion, and perspective based on his experience in the field. He supervised the work done in Chapter 3.
- D. M. Manias collaborated on the implementation of the proposed solution. He contributed to the work done in Chapter 5.

Acknowledgments

I am very thankful to Prof. A. Shami, who provided thorough and helpful support with close guidance throughout my masters' program. He has been my supervisor and friend.

Many thanks to my friends and colleagues for the words of support and help during my Ph.D. research program.

To my family, particularly my parents, thank you for your love, support, and unwavering belief in me. Without you, I would not be the person I am today.

Above all, I would like to thank my wife Manar Jammal for her love and constant support, for all the late nights and early mornings, and for keeping me sane over the past few months (years). Thank you for being my muse, editor, proofreader, and sounding board, but most of all, thank you for being my best friend.

Table of Contents

Abstract ii

Summary iv

Co-Authorship.. vi

Acknowledgments... viii

Table of Contents ix

List of Tables xiv

List of Figures xv

Acronyms 1

Chapter 1 6

1 Introduction 6

1.1 Thesis Outline 13

1.2 Thesis Contributions 14

1.2.1 Chapter 2 contributions 14

1.2.2 Chapter 3 contributions 15

1.2.3 Chapter 4 contributions 15

1.2.4 Chapter 5 contributions 16

Chapter 2 20

2 Exploring Microservices as the Architecture of Choice for Network Function Virtualization Platforms 20

2.1 Introduction 20

2.2 Microservices Architecture 23

2.3 Microservices and NFV: A Match Made with Modularity Cloud9 25

2.3.1 VNFC Bounded Context 25

2.3.2 VNFC Modularity 25

2.3.3 VNF Innovation and Evolution 26
2.3.4 VNF Flexibility and Scalability 26
2.3.5 VNFCs Interoperability 26
2.4 Microservices NFV and Mobile Edge Computing 26
2.5 Challenges of NFV Implementing Microservices 28
2.5.1 VNFCs Networking Complexity 29
2.5.2 VNFCs Service Discovery 29
2.5.3 VNFCs Service Monitoring, Logging, and Meta-Data Collection 30
2.5.4 VNFCs Security 30
2.5.5 Infrastructure Convergence 32
2.5.6 Routing Convergence 33
2.5.7 Inter- and Intra-connecting VNFCs 34
2.5.8 Placement of VNFCs 35
2.6 VNFCs Placement Modelling 37
2.6.1 Resources Capacity Constraints 38
2.6.2 Network Delay Constraints 38
2.6.3 Availability Constraints 39
2.6.4 Redundancy Constraints 39
2.6.5 Anchors Constraints 39
2.6.6 Orbital Area Constraints 39
2.7 VNFCs Placement Simulation 40
2.8 Conclusion 45
Chapter 3 48
3 NFV/SDN-based vEPC Solution in Hybrid Clouds 48
3.1 Introduction 48

3.2 Evolved Packet Core 51

3.3 Virtualizing EPC 53

3.4 NFV/SDN Based vEPC Solution 55

3.5 vEPC Entities Placement in DFVisor 60

3.5.1 MME and HSS FE 60

3.5.2 UDR, PCRF, OCS, and OFCS 61

3.5.3 SGW and PGW 64

3.6 Functionality Analysis 65

3.7 Conclusion 67

Chapter 4 72

4 Network Function Virtualization-Aware Orchestrator for Service Function Chaining Placement in the Cloud 72

4.1 Introduction 72

4.2 Background and Related Work 76

4.3 Motivation 81

4.3.1 VNF Placement Requirements 81

4.4 Problem Formulation 83

4.4.1 VNF Architecture 83

4.4.2 Requirements of VNFs Scheduler 84

4.5 Mathematical Formulation 86

4.5.1 Model Formulation 87

4.5.2 Model Complexity 88

4.6 BACON: NFV-Aware Placement Algorithm 90

4.6.1 Calculation of Betweenness Centrality 91

4.7 NFV-Aware Placement Simulation 95

4.7.1 A. Simulation Results and Evaluation 95

4.7.2 Component Intra-Communication Delay Comparative Analysis 98

4.7.3 SFC End-to-End Delay Comparative Analysis 103

4.8 Conclusion 107

Chapter 5 118

5 Orchestrating Network Function Virtualization Platform: Migration or Re-Instantiation? 118

5.1 Introduction 118

5.2 Related Work 121

5.3 Background and Motivation 125

5.3.1 ETSI MANO Framework 125

5.3.2 System Modeling and Requirements 127

5.3.3 Service Levels and their Requirements 132

5.3.4 VNF Placement Considerations in a Cloud Environment 132

5.3.5 Migration and Re-Instantiation 133

5.4 Optimization Model Formulation 134

5.4.1 Approach Model 135

5.4.2 Problem Constraints 136

5.4.3 Notations and decision variables: 137

5.4.4 Mathematical Formulation 138

5.5 MILP Model Complexity 140

5.6 Heuristic Solution 141

5.7 Results and Analysis 143

5.7.1 Small-Scale Network 143

5.7.2 Medium-Scale Network 144

5.7.3 Large-Scale Network 148

5.8 Conclusion 149

Chapter 6 156

Conclusion 156

6.1 Chapter 2 Summary 158

6.2 Chapter 3 Summary 158

6.3 Chapter 4 Summary 158

6.4 Chapter 5 Summary 159

Thesis Future Work 160

6.5 Elasticity mechanisms of NFV based cloud applications 160

6.6 Creation of a unified objective-aware management environment 161

7 Curriculum Vitae 162

List of Tables

Table 2.1: Challenges and Solutions of NFV microservices architecture adoption 31

Table 2.2: The model input data 42

Table 2.3: Functional group members 44

Table 3.1: SDN architecture comparison.. 66

Table 4.1: Small-Scale DC Network Dataset 97

Table 4.2: Large-Scale DC Network Dataset 97

Table 4.3 Small-scale DC network functional group members count 103

Table 4.4 Large-scale DC network functional group members count 103

Table 5.1: Small-Scale Network Topology 142

Table 5.2: Medium-Scale Network Topology 145

Table 5.3: Large-Scale Network Topology 149

List of Figures

Figure 2.1: Inter- and Intra- connections of VNFCs 35

Figure 2.2: The orbital area of a given VNFC 40

Figure 2.3: Computational paths delays 43

Figure 3.1: Co-existence of vEPC with cloud applications 51

Figure 3.2: Enhanced Controller 58

Figure 3.3: Enhanced Switch 59

Figure 3.4: vEPC entities placement with DFVisor in public and private cloud with multi-tenant support 62

Figure 4.1: Service function chain and computational path of NFV of different 82

Figure 4.2: BACON: The proposed heuristic algorithm. 93

Figure 4.3: Intra-connection delay between VNF instances of types MME and HSS. 99

Figure 4.4: Intra-connection delay between VNF instances of types MME and SGW. 99

Figure 4.5: Intra-connection delay between VNF instances of types SGW and PGW......... 100

Figure 4.6: The end-to-end delays of SFCs in small-scale DC network. 105

Figure 4.7: The end-to-end delays of SFCs in large-scale DC network. 106

Figure 4.8: The placement zones of VNFs depending on their sponsors and 107

Figure 5.1: Stages of Failure Management 131

Figure 5.2: Small-Scale Network Downtime 143

Figure 5.3: Small-Scale Interdependent Component Delay 145

Figure 5.4: Small-Scale Computational Path Delay 145

Figure 5.5: Medium-Scale Network Downtime 146

Figure 5.6: Medium-Scale Computational Path Delay 147

Figure 5.7: Difference between Optimal and Heuristic Solutions 147

Figure 5.8: Average Delay Across Computational Paths - Large Scale 148

Acronyms

3GPP	3rd Generation Partnership Project
AS	Autonomous System
ASw	Aggregation Switches
AWS	Amazon Web Services
BACON	Betweenness Centrality Algorithm for Component Orchestration Of NFV Platform
BC	Betweenness Centrality
BGP	Border Gateway Protocol
BSS	Business Support Systems
CAPEX	Capital Expenditure
CDN	Content Distribution Network
CN	Core Network
COTS	Commercial-Off-The-Shelf
CSP	Cloud Service Provider
D2D	Device-To-Device
DC	Data Center
DevOp	Development and Operations
DFVisor	Distributed FlowVisor
DNS	Domain Name Server
EPC	Evolved Packet Core

ETSI	European Telecommunications Standards Institute
FE	Frontend
GPRS	Packet Radio Service
GRE	Generic Routing Encapsulation
GTP	GPRS Tunneling Protocol
HA	High Availability
HLR-FE	Home Location Register Frontend
HSPA	High Speed Packet Access
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
ICT	Information and Communications Technology
IMS	Information Management System
IoT	Internet of Things
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Networks
LTE	Long Term Evolution
M2M	Machine-To-Machine
MANO	Management and Orchestration
MEC	Mobile Edge Computing
MFC	Mobile Flow Controller
MFFE	Mobile Flow Forwarding Engine

MILP	Mixed Integer Linear Programming
MME	Mobile Management Unit
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NFVO	Network Function Virtualization Orchestrator
NSP	Network Service Providers
OCS	On-Line Charging System
OFCS	Off-Line Charging System
OPEX	Operating Expenditure
OSPF	Open Shortest Path First
OSS	Operations Support Systems
P2P	Peer-To-Peer
PaaS	Platform as A Service
PB	Programmable Buffer
PCRF	Policy and Charging Rules Function
PDN-GW	Packet Data Network Gateways
PGW	Packet Data Network Gateway
PoP	Point of Presence
QoE	Quality of Experience
QoS	Quality of Service
RPC	Remote Procedure Call
SaaS	Software as A Service
SDN	Software Defined Networking

SFC	Service Function Chain
SGSN	Support Node
SGW	Serving Gateway
SLA	Service Level Agreements
SOA	Software Architecture
SR-IOV	Single-Root Input/Output Virtualization
TCP	Transmission Control Protocol
TOR	Top of The Rack
UDR	User Data Repository
vCPU	Virtual Central Processing Unit
vEPC	Virtualized Evolved Packet Core
VIM	Virtualized Infrastructure Manager
VM	Virtual Machines
VM	Virtual Machine
VNFC	Virtual Network Function Component
VNFCs	Virtual Network Function Components
VNFD	Virtual Network Function Descriptor
VNFFG	Virtual Network Function Forwarding Graph
VNFM	VNF Manager
VNFO	VNF Orchestrator
VNFs	Virtualized Network Functions
VNI	Virtual Network Infrastructure
vNIC	Virtual Network Interface Controllers

VoLTE	Voice-Over-LTE
VoLTE	Voice-Over LTE
VPN	Virtual Private Networks
vSwitch	Virtual Switch
VxLAN	Virtual Extensible Local Area Network
WAN	Wide Area Netwrok

Chapter 1

1 Introduction

With the immense connectivity, our world has today, meeting the high demands for networking services has become a critical goal to ensure the seamless, highly available, and secure delivery of every network services [1]. In the era of 5G networking, Internet of Things (IoT) and massive content delivery networks, Network Service Providers (NPSs) face major challenges in adapting to the surge of demand for network services. Today, the finishing stones for 5G networks are being placed where through this architecture everything around us can become connected and register as a new member in the massive networking crowd. The number of connected IoT devices with cellular connections around the world is projected to grow to 1.5 billion by the year 2022. Netflix is considered one of the largest content delivery networks today and by the 3rd quarter of 2019 has reached 158.33 million subscribers [2]. These subscribers actively request content with extremely high expectations when it comes to the received quality. This large increase in demand has put NSPs against challenges that limit their capabilities in efficiently keeping up with those demands, this is mostly because their underlying hardware infrastructure is “not flexible” hindering them from quickly adapting to the dynamic needs of today’s network users [3]. Adding, removing, or modifying their services on the legacy network infrastructure is extremely costly, reflecting on their CAPEX and OPEX expenditures. The main reason for such high costs is that the software aspect of those services is strongly coupled with its hardware. The current infrastructure of the majority of NSPs has dedicated hardware for their services. This results in demolishing the revenue margin when a new service or upgrade is released. This challenge does not only impact the direct cost, but it is also critical from the energy point of view, where excess resources can get assigned or deployed to anticipate the user demands, this leads to having a large number of resources left unutilized but still running while consuming energy [4]. Another key challenge that rises when demands increase on rigid network infrastructure is the lack of qualified technical personnel. To mitigate these challenges, NSPs have started their network-wide adaptation of virtualized environments.

Software-Defined Networking (SDN) and Network Function Virtualization (NFV) are one of the key platforms today for network providers to use for a rapid expansion of their services, adding innovation, and lowering their costs, while also maintaining their Quality of Service and tackling the rapid expansions of networking demands. SDN and NFV have shown great potential in improving the economics of networking in parallel to giving the ability to design and quickly deploy new innovative service capabilities [5]. Below is an overview of those key technologies.

Experimenting with routing protocols at Stanford University, led to the novel creation of the OpenFlow protocol, a protocol that allows the remote programming of flow tables in switching systems. This protocol opened the opportunity to separate the control plane software from the data plane hardware [5]. This concept is now known as SDN. In SDN the control plane is designed and implemented as software components that can run on any industry-standard hardware. This separates the control plane from the data plane and gives the full control of the network operations to the SDN controller, resulting in a simple data plane hardware that obeys the controller’s instructions with minimal or no decisions to be taken at those nodes (switches, simple routers). With such an approach, the control plane now becomes more powerful and capable of quickly deploying new services. another key benefit is the significantly reduced cost as low-cost commodity switches combined with control plane software running on standardized servers are much more economical than traditional complex routers that deliver the same functionality.

Another key concept that shines as a tool to tackle this rapid growth is NFV. In traditional network approaches and most of the current legacy systems, the services offered by a network provider are implemented as dedicated appliances on proprietary hardware. Those appliances contain software functionalities and components that cannot be separated from the hardware. This approach is expensive to maintain, it also makes it a resource (Employee working hours…) intensive process to add or modify those tightly coupled services. NFV proposes that these services transition to becoming software-based services that can be hosted on any industry-standard off-the-shelf hardware. with such an approach, the cost of offering key network functionalities is dramatically reduced, because specialized hardware components are expensive to build, and this cost directly reflects on the consumer as the network equipment vendors need to maintain their return on investments on these hardware products. In addition, with virtualizing these services, it becomes far less expensive to maintain them, more convenient to deploy, relocate, or terminate services, and optimize the network performance by dynamically assigning resources when needed. This approach has only become possible due to the recent advancement in industry servers, they are now capable of surpassing purpose-built hardware in terms of memory, CPU processing power, and energy consumption while still be much more cost-effective. Examples of network functions that can be virtualized are:

- Message Router
- Content Delivery Networks

- Session Border Controllers
- Deep Packet Inspection
- Firewall

The goal of this thesis is to pave the road towards the seamless, efficient, and reliable transformation of the legacy network architectures, towards a fully virtualized environment through the utilization of the state-of-the-art NFV and SDN research. This work focuses on how to solve key challenges in the road towards decoupling the software components from their proprietary hardware and create an optimized management environment to ensure that NSPs can still maintain or even surpass their current Quality of Service standard, meet strict service level agreements (SLAs), and be able to adapt to the rapid increase in demand while moving their services to a completely virtualized environment that allows them to optimize resource allocation, meet strict delay requirements, ensure high availability of their services, and dynamically adapt to end-user demands by deploying, modifying, or terminating certain network services in real-time.

Recently, NFV and SDN have captured the interest of researchers in academia and industry where several approaches have been proposed. Gember Jacobson et al. identify operational challenges in several of the proposed network function state transfer frameworks such as: safety, scalability, and efficiency and propose methods including packet reprocessing and Peer-to-Peer (P2P) transfers in an effort to reduce latency and state transfer time [6]. Rajagopalan et al. propose Split/Merge, a system that enables the dynamic scaling in and out along with distributed load elasticity [7]. Woo et al. propose a framework that meets performance thresholds and allows for the elastic scaling of VNFs [8]. These studies deal with the migration mechanism, however, they do not consider the practical implementation of this mechanism in a virtualized network as many migration-aware constraints such as availability and resources have been discarded. Taleb et al. [9] implement virtualized EPC (vEPC) using the cloud computing environment and demonstrate the feasibility of providing vEPC as a service. The authors also propose a comparative analysis of various architectures. Baba et al. [10] present and implement a vEPC architecture based on the VNFs. The architecture satisfies the requirements of the machine-to-machine service computing with reduced resources. The authors achieve a 27% CPU time reduction with the proposed architecture. A smart VNF placement to deploy multi-tier cloud applications is proposed by PACE [11]. However, PACE overlooks many of the requirements that affect the VNF placement to achieve the desired QoS in multi-tier cloud-based applications. These requirements include the VNF dependency hierarchy, delay tolerance, and anti/co-location constraints. An efficient and scalable VNF provisioning framework is proposed in E2 [12]. E2 is a framework that manages the VNFs by combining traffic engineering and the best VNF placement. It is suitable for a private cloud that serves a single type of applications and provides specific functionalities, such as traffic offloading to proprietary switches. E2 has discarded the various placement constraints, such as the instances inter and intra-dependency and the delay tolerance between components. Bari et al. [13] propose an optimization algorithm for the VNF placement with a simplified set of constraints. The latter only considers the deployment cost, the resources requirement, and the processing delay. This optimization algorithm discards the placement constraints that satisfy the carrier-grade requirements of the VNF applications, such as the VNF chaining, reliability, and delay tolerance constraints. Mohammad Khan et al. [14] formulate a mixed-integer linear programming optimization model for VNFs placement and traffic flow routing while minimizing resource utilization. However, the proposed solution has focused on minimizing computational resources while ignoring non-functional constraints such as redundancy, dependency, and availability. Sahel et al. [15] focus on the network service chaining problem by formulating an integer linear programming model and a heuristic algorithm. The proposed solution is based on two segments: a decomposition selection with a backtracking phase and a mapping phase; leading consequently to suboptimal solutions.

Most of the aforementioned approaches address solutions through private cloud interfaces, which are completely owned and controlled by cloud service providers. Furthermore, they donot consider the co-existence of different applications hosted with the virtualized mobile core network entities in the cloud platform. So far, the proposed solutions for the NFV-SDN framework and virtualizing EPC are applicable mostly to small networks within a private cloud. Private clouds are groups of data centers owned by the network service providers, and these providers have full control over the entire infrastructure (physical servers, underlying core networks, virtual environments, and orchestrators). These solutions do not address multi-tenant support and co-existence with variant cloud applications. To unleash the potentials of the virtual network functions of an NFV service, they should have the capability to be deployed in a hybrid cloud. A hybrid cloud is a composite of different cloud types (private, public, and community clouds); its architecture requires both on-premises (private) and off-site (public) cloud infrastructure. Within this architecture, service providers can host user-critical information applications in private clouds while hosting computationally demanding applications in public clouds in different geographical locations. ETSI NFV group has proposed a management and orchestration framework that could be integrated with the current IT virtualization environment to enhance VNF lifecycle management and orchestration. This framework is only intended to describe which entities are required to allow VNF management by the orchestrators, and how they might be integrated within operations and business support systems (OSS/BSS). ETSI NFV group does not clarify how NFV will support multi-tenant and on-demand scalability, which are objectives specified by NFV. To increase the adoption of NFV in the telecommunication industry, NFV entities should meet all carrier-grade requirements with respect to performance, fault resilience, high availability, scalability, quality of service (QoS), and governments geo-restrictions. To achieve the desired requirements, it is important to observe and investigate the main functional blocks involved in service provisioning. Therefore, in this thesis, various challenges have been addressed and mitigated by providing novel solutions. These challenges are summarized as follows:

- The adoption microservice architecture for NFV platform components that can be hosted in a dynamic environment ranging from mobile edge computing to cloud environment. The major challenges and requirements of the microservices architecture are addressed to fully exploit the potentials of its adoption in NFV.
- The implementation of the SDN platform components that comply with microservices architecture and enable hyper-scalability of the platform.
- The VNFs placement problem in a service function chain while satisfying the carrier-grade requirements.
- The modeling of the carrier-grade functionality requirements imposed on service function chains.

The following section highlights the contribution of this work in detail towards mitigating the challenges that the NSPs face today in adapting to NFV/SDN technologies.

1.1 Thesis Outline

This section highlights the structure of this thesis. Chapter 2 aims at exploring the architecture of microservices as an architecture of choice for NFV platforms. The chapter addresses the challenges of the microservices architecture to be chosen as a solution for building dynamic environments through NFV technologies, the chapter presents a scheduler with the goal of minimizing network delays while considering several network constraints and restrictions. Chapter 3 introduces a solution to seamlessly transform the Evolved Packet Core (EPC) network to a fully virtualized environment in the cloud. This migration is done with a fine-grained QoS while the network maintains its abilities to immensely scale. The chapter also provides a solution to manage the co-existence of the Virtualized Network Functions (VNFs) in private and public multi-tenant architectures. Chapter 4 proposes in detail a novel solution for an NFV orchestrator for the optimal and efficient placement of Virtualized Network Functions (VNFs) in the cloud. The chapter proposes a novel Mixed Integer Linear Programming model that considers the carrier-grade requirement of an NFV application while minimizing both the end-to-end and intra-delays of the SFCs, the chapter also proposes a novel heuristic solution namely the Betweenness centrality Algorithm for Component Orchestration of NFV platform (BACON), to address the time complexity challenges of the MILP model. Chapter 5 aims at solving one of the key challenges in the adaptation of NFV/SDN technologies, the challenge of optimally handling the failure of the offered network functions, specifically in optimally choosing the decision of migrating network services or instantiating a new instance of that service. The chapter proposes both and optimization MILP model that considers the carrier-grade requirements of the network and aims at minimizing the SFC delays. The heuristics solution takes into consideration the same goals of the model but mitigates the challenge of the time complexity characteristics of the MILP in order to adapt this novel solution to large and practical rea-world scenarios in the areas of IoT and 5G networking.

The sections below discuss the details of the chapter contributions and more details about the challenges that were tackled during each chapter.

1.2 Thesis Contributions

The major contributions of this thesis are summarized as follows.

1.2.1 Chapter 2 contributions

NFV is an emerging key technology that overcomes many challenges facing network service providers, such as reducing the capital and the operating expenses and satisfying the growing demand for mobile services. Integrating NFV with MEC and cloud environment requires an architecture that enables efficient implementations and deployments of NFV entities. Microservices architecture is a promising implementation of service-oriented architecture with recognized advantages in terms of modularity and continuous delivery. This chapter envisions microservices architecture as the solution of choice for building NFV platforms that are hosted in a dynamic environment ranging from MEC to a cloud environment. The chapter addresses the major challenges and requirements of the microservices architecture to fully-exploit the potentials of its adoption in NFV. It also proposes some potential solutions that alleviate these issues. Besides, the chapter discusses the need for agile and modular NFV entities along with MEC to realize various applications. To this end, the chapter discusses explicitly a novel NFV microservices entities scheduler optimization model. The proposed scheduler aims at minimizing network delays while taking into consideration various functional and non-functional constraints. The evaluation of the simulation results demonstrates that the proposed model minimizes the computational paths' latencies and improves the performance and availability of the NFV service chains.

1.2.2 Chapter 3 contributions

This chapter proposes a novel NFV/SDN-based solution that allows the migration of the mobile core network to the cloud with a fine-grained QoS while maintaining the scalability of the virtualized network function entities. Additionally, the proposed solution facilitates the co-existence of virtualized network function instances and IT could applications in a multi-tenant private and public cloud architecture.

1.2.3 Chapter 4 contributions

NFV has been introduced by NSPs to overcome various challenges that hinder them from satisfying the growing demand for networking services with higher return-on-investment. The association of NFV with the leading technologies of IT virtualization and software-defined networking is paving the way for flexible and dynamic orchestration of the VNFs, but still, various challenges need to be addressed. The VNFs instantiation and placement problems on Data Center's (DC) servers are key enablers to achieve the desired flexible and dynamic NFV applications. In this chapter, we have addressed the VNF placement problem by providing a novel Mixed Integer Linear Programming (MILP) optimization model and a novel heuristic solution, Betweenness centrality Algorithm for Component Orchestration of NFV platform (BACON), for small- and large-scale DC networks. The proposed solution addresses the VNF placement while taking into consideration the carrier-grade nature of the NFV applications and at the same time, minimizing the intra- and end-to-end delays of the Service Function Chain (SFC). Also, the proposed approach enhances the reliability and the Quality of Service (QoS) of the SFC by maximizing the count of the functional group members. To evaluate the performance of the proposed solution, this chapter conducts a comparative analysis with an NFV-agnostic algorithm and a greedy-k-NFV approach, which is proposed in the literature work. Also, the chapter defines the complexity and order of magnitude of the MILP model and BACON. BACON outperforms the greedy algorithms especially the greedy-k-NFV solution and has a lower complexity, which is calculated as $O((n^3 - n^2)/2)$. The simulation results show that finding an optimized VNF placement can achieve minimal SFCs delays and enhance the QoS accordingly.

1.2.4 Chapter 5 contributions

Network function virtualization (NFV) provokes the evolution of network functions to overcome various challenges facing the network service providers (NSPs). To exploit the advantages of virtualization technology, NFV platforms should use the cloud environment to provide their services. Typically, an NFV service is represented by a service function chain (SFC) that consists of multiple virtualized network functions (VNFs). Hosting and orchestrating these VNFs in a cloud environment are challenging tasks. In this chapter, we discuss the VNF orchestration problem from the perspective of VNF's migration and re-instantiation mechanism to achieve carrier grade-aware NFV services in a cloud-based platform. This paper also provides detailed insights on the NFV system modeling, building blocks, and various challenges hindering its cloud adoption. Also, a novel mixed-integer linear programming (MILP) optimization model and a graph-based heuristic solution are proposed as solutions to facilitate the NFV platform orchestration in a cloud environment. These approaches decide between triggering either VNF's migration or re-instantiation while achieving minimal downtime of the VNF, satisfying carrier-grade requirements, and finding an optimal placement for the migrated or re-instantiated VNF that minimizes the SFC delays. The proposed solutions are compared to two availability-agnostic greedy algorithms. The simulation results show that finding an optimized decision whether to migrate or re-instantiate a VNF while associating it with an optimal placement can achieve a minimal VNF's downtime and SFCs delays and can enhance the quality of service accordingly.

References

1. M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, “Software-Defined Networking: State of the Art and Research Challenges,” *Computer Networks*, 2014, pp. 74-98.
2. Netflix, “Number of Netflix paying streaming subscribers worldwide from 3rd quarter 2011 to 3rd quarter 2019 (in millions”, *Statista*, October 2019.
3. 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.-Dec. 2014.
4. M. Jammal, H. Hawilo, A. Kanso and A. Shami, "Mitigating the Risk of Cloud Services Downtime Using Live Migration and High Availability-Aware Placement," *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2016, pp. 578-583.
5. M. Taylor, “A Guide to NFV and SDN,” https://info.metaswitch.com/hs-fs/hub/415294/file-1827221136-pdf/PDFs/metaswitch-white-paper-guide-to-nfv-and-sdn.pdf, 2014.
6. T. Yarygina and A. H. Bagge, “Overcoming Security Challenges in Microservice Architectures,” IEEE Symposium on Service-Oriented System Engineering (SOSE), 2018, pp. 11-20.
7. Cisco and Netapp, “Converged Vs. Hyper-Converged Infrastructures: Understanding The Impact On Your Organization,” White Paper, 2016.

8. NUTANIX, "The Secret to Google's Datacenter and Why it Matters to Every Enterprise," https://www.nutanix.com/2014/02/20/the-secret-to-googles-datacenter-and-why-it-matters-to-every-enterprise/, February 2017.
9. Taleb et al., "EASE: EPC as a service to ease mobile core network deployment over cloud," IEEE Netw. Mag., vol. 29, no. 2, pp. 78–88, Mar. 2015.
10. H. Baba, M. Matsumoto, and K. Noritake, "Lightweight virtualized evolved packet core architecture for future mobile communication," in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Mar. 2015, pp. 1811–1816.
11. L. E. Li et al., "PACE: Policy-aware application cloud embedding," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2013, pp. 638–646.
12. P. Shoumik et al., "E2: A framework for NFV applications," in Proc. 25th Symp. Oper. Syst. Princ., Oct. 2015, pp. 121–136.
13. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," IEEE Trans. Netw. Service Manage., vol. 13, no. 4, pp. 725–739, Dec. 2016.
14. A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. K. Ramakrishnan, and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," in Proc. 21st IEEE Int. Workshop (LANMAN), Apr. 2015, pp. 1–6.
15. S. Sahhaf et al., "Network service chaining with optimized network function embedding supporting service decompositions," Comput. Netw. J., vol. 93, pp. 492–505, Dec. 2015.

Chapter 2

2 Exploring Microservices as the Architecture of Choice for Network Function Virtualization Platforms

2.1 Introduction

Network service providers (NSPs) are certainly facing challenges in satisfying the rapid increase in network connectivity demands while maintaining the required quality of service (QoS). Also, over-the-top application providers are continuously harvesting the traditional NSPs' revenue streams. These changes in the competition landscape narrow the return-on-investment margin and overwhelm the networking infrastructure of the NSPs. With the inevitable presence of networking infrastructure in any application stack of information and communications technology (ICT), NSPs leverage their ability to deliver reliable service and enhance extensive customer intimacy to explore new business opportunities. This can increase the NSP average revenue per user. NSPs are also seeking accretion of new applications into their service models to enhance and expand their enterprise services portfolio beyond the connectivity realm. To achieve this desired vision, NSPs have projected the need for a programmable automated infrastructure that drives real-time, flexible, and user-application-centric network connectivity services. However, the dependency of the current network on an extravagant proprietary complicated infrastructure prevents the NSP from realizing automated programmable networks without overwhelming their capital and operating expenditure (CAPEX and OPEX) budgets. Virtualization technology emerges as an intriguing solution for this challenge. Virtualization technology has been originally introduced as a solution to achieve a smaller footprint and efficient utilization of computing resources in enterprise data centers (DCs). To this end, NSPs investigate the opportunity to employ virtualization within their infrastructure to lower their CAPEX and OPEX investments.

A major milestone has been reached when a group of NSPs under the European Telecommunications Standards Institute (ETSI) introduces the network function virtualization (NFV). NFV is the technology that migrates the networking functions from the proprietary hardware to virtual network functions (VNFs). The latter is implemented as software applications running on commercial-off-the-shelf (COTS) information technology (IT) infrastructure. NFV utilizes various IT virtualization techniques based on commodity hardware (computing resources, storage, and networking) to consolidate network function applications. This consolidation enables the NSPs to take advantage of the lower cost and innovative dynamics of traditional IT infrastructure. In that context, a powerful companion technology to NFV is software-defined networking (SDN): the technology that introduces real-time network programmability. With the effective integration between these two paradigms, NSPs can expect major improvements in component modularity and implementation agility. This improvement will have a direct impact on CAPEX, OPEX, and time-to-market application releases. Besides, rapid innovation will emerge in the ICT industry.

In the ETSI definition of the basic architecture standards for the VNFs, each VNF consists of one or more virtual network function components (VNFCs) [1]. VNFCs implement various functionalities that provide the service defined by the VNF descriptor (VNFD). This architecture allows the standardization group to have well-defined interfaces for the VNFs' services while granting the VNFCs implementation freedom to the VNF software providers. Having well-defined standard interfaces of VNFs provides stable software releases while enabling interoperability of VNFs between various software provider vendors. The

VNFCs implementation freedom drives the innovation and evolution of the VNF services and provides the capability of flexible management and orchestration of the VNFCs lifecycle based on functional and non-functional constraints.

NSPs intend to deploy NFV services in cloud environments to take advantage of their businesses and service models, such as pay-as-you-go and scale up or down on-demand. Furthermore, NFV is expected to complement the mobile edge computing (MEC) technology to provide accelerated content delivery and better application responsiveness, such as intelligent edge data caching, to enhance the quality of user experience (QoE). MEC has been introduced by ETSI as a technology that enables the deployment of services and applications in the edge network to achieve the closest proximity to the end-user [2]. With these intentions, new software development perspectives should be adopted by NFV to ease the VNFs deployments and their integration with the cloud and MEC environment.

Since VNFs are constructed by chaining various VNFCs to provide networking services, this chapter envisions microservices architecture, the emerging implementation of service-oriented software architecture (SOA), as the solution of choice for developing a VNF. In the foreseen design, each VFNC is a microservice component by itself. Microservices architecture allows the VNFs services to be more flexible in the hosting environment where the virtualized functionalities can adopt various manageability scopes to meet the functional and non-functional constraints. To fully exploit the potentials of adopting the microservices architecture in NFV, it is necessary to define the major challenges introduced by this architecture and address them accordingly.

This chapter discusses the adoption of microservices architecture in NFV and provides a guideline to design a placement scheduler for the VNFCs. The main contributions of this work can be summarized as follows:

i) Define the major challenges of adopting microservices within NFV platforms.

ii) Define the requirements for microservices architecture to fully-exploit the potentials of its adoption in NFV.

iii) Propose some potential solutions that alleviate the challenges of adopting micro-services within NFV platforms.

iv) Discuss explicitly a novel optimization model for the NFV microservices entities' scheduler. The model aims at minimizing the computational paths network delays while taking into consideration various functional and non-functional constraints.

2.2 Microservices Architecture

In the last decade, ICT industry has witnessed major breakthroughs in terms of ways the world interacts and exchanges information. With the inevitable dependency on mobile smart devices that ranges from personal use to the internet of things (IoT) connected devices, new paradigms of applications have emerged, such as social media, video-on-demand applications, and software as a service. These paradigms are associated with advances in computing resources services. Cloud computing accompanied by virtualization is introduced as an infrastructure foundation to meet the rapid and increasing demands of computing resources with minimal CAPEX and OPEX investments. Adopting cloud computing services in an application development requires remodeling of the application architecture to exploit the benefits of cloud services, such as scaling on-demand.

Traditionally, web-based applications are developed using monolithic architecture. The latter is a software with a vertically integrated stack that is executed in a single process. This practice of software development facilitates the application deployment and networking where multiple instances can easily reside behind a load-balancer to satisfy the application service demands. However, the change in the application nature and the increase in the complexity and demand of the provided services introduce various challenges to monolithic applications. The tightly coupled codebase is typically a result of the monolithic application, which imposes high-risk association with any code change or addition of a new feature. Applying any change to one component can seemingly affect the whole system functionality. Moreover, the monolithic application does not support component reusability, which hinders the scalability of an individual component. This can cause an inefficient utilization of computing resources.

Microservices architecture has evolved to mitigate monolithic architecture challenges by introducing distributed systems with lightweight components. Each component performs a specific workload in an independent manner. Components are defined as microservices in this architecture. Microservice is a kind of software that is contained in its process and typically uses web-based protocols, such as transmission control protocol (TCP), hypertext transfer protocol (HTTP), or remote procedure call (RPC) protocol to communicate. Despite that microservices architecture is proposed as a solution to have efficient scalable distributed systems, it introduces new challenges.

2.3 Microservices and NFV: A Match Made with Modularity Cloud9

Leading ICT equipment vendors have rushed to build and release various proof-of-concepts designs and prototypes of VNFs running on COTS computing resources. However, these prototypes are based on traditional network function development and monolithic stack development that can only scale vertically and are limited to the computing performance of the underlying bare-metal servers [3]. Since networking functions' applications thrive on the power of computing resources, NSPs faced with the challenge of re-engineering VNFs to enable horizontal scaling. Being in the process of fully adopting cloud computing to build the Telco-cloud, NSPs aims at adopting the best performing architectures in the web-scale development world where scalable and distributed applications reside, such as Amazon, Google, and Netflix platforms. Microservices architecture is considered the best-fit architecture to assist NFV in achieving its goals. Defining VNFCs as microservices provides the following advantages:

2.3.1 VNFC Bounded Context

Each VNFC performs a limited set of functionalities, which results in a small code base limiting the scope of bugs. Furthermore, the standalone nature of microservices facilitates direct testing of functionalities in isolation with respect to the VNF provided service.

2.3.2 VNFC Modularity

This means gradual transitions to updated versions of VNFCs. The newer versions of VNFCs can be deployed simultaneously with the old ones. The VNFCs that depend on the old versions can be gradually modified to interact with the updated VNFCs, which is known by rolling upgrade. With this approach, NFV can adopt VNFCs continuous integration and can greatly ease the VNF software maintenance VNFC Bounded Context

Each VNFC performs a limited set of functionalities, which results in a small code base limiting the scope of bugs. Furthermore, the standalone nature of microservices facilitates direct testing of functionalities in isolation with respect to the VNF provided service.

2.3.3 VNF Innovation and Evolution

By exploiting the independency characteristic, new NFV microservices can be easily introduced to the production services without disrupting their operations.

2.3.4 VNF Flexibility and Scalability

VNF building blocks, VNFCs, can be scaled up or down independently according to the service demand.

2.3.5 VNFCs Interoperability

With a microservices architecture, VNFCs can be deployed in a heterogeneous manner. Various VNFCs provided by different vendors or developed using different programming languages and frameworks can still be interconnected as long as they implement the right communication interfaces.

2.4 Microservices NFV and Mobile Edge Computing

Designing high bandwidth networks with negligible latency is the intent of the service providers to serve many emerging applications, such as Internet of Everything, device-to-device (D2D) communication, voice-over-LTE (VoLTE), on-demand video streaming (4K and 8K videos), augmented reality, and various internet protocol (IP) multimedia subsystem (IMS) services. Implementing such broadband mobile networks requires efficient utilization of the assigned spectrum for wireless communication and the distribution network infrastructure. It also requires placing the data-hosting application servers in closest proximity to the end-users to achieve negligible latency. With the spectrum being the scarce resource, mobile network service providers are tending to deploy heterogeneous networks where macro and micro base station cells coexist with small base-station cells (pico- and femto-cells). Heterogeneous networks enhance the spectrum utilization to achieve a higher data rate for the end-users (user equipment). Mobile edge computing (MEC) is introduced to minimize the latency of serving data through hosting the application servers with the closest proximity to the end-users, especially data caching servers. In such networks, substantial growth of signaling traffic on the core network (CN) can be generated due to the reduced cell size and increase in user density and mobility. The signaling traffic growth is flourishing due to the emergence of new services on mobile technology platforms.

Nowadays, on-demand video streaming and social media applications are responsible for 65% of mobile data traffic, and it is expected to reach 90% by 2022 according to the Ericsson mobility report [4]. Therefore, the existence of applications and data caching servers in the mobile edge networks is essential to offload the data traffic from the core network and minimize the networking latency while serving the maximum number of users with high bitrates. In the current and legacy mobile networks, the application servers and the content data should be accessed from centralized data centers and content distribution network (CDN) nodes. The latter nodes are placed at the mobile core network and the point of presence (PoP) that constrains the backhaul networks. Given the evolution at the level of base stations, D2D, and storage technology, deploying the application and caching servers at macro, micro, pico and femto base stations become feasible. However, flexible, agile, and automated network entities should exist side by side with the MEC entities to achieve the desired application and data caching schemes for the above designs. NFV and SDN are proposed to achieve these objectives for networking entities, but so far, they are examined and researched in the context of monolithic applications. NFV and SDN-based network services and components should be proposed and provided as microservices to scale, complement with MEC, and enable advanced application and data caching deployment criterion. Implementing NFV and SDN networking microservices entities at the network edges offloads the networking orchestration traffic from the core network and enables elastic network federations that can be self-sustained while providing high bandwidth connectivity with negligible latency for the end-users. The centralized core networking entities can then synchronize and orchestrate the network federations' inter-traffic.

2.5 Challenges of NFV Implementing Microservices

NFV adopting microservices paves the way for the arrival of the telco-cloud. To ensure wider adoption of NFV by the ICT industry, NFV should overcome the challenges introduced by the softwarization of network functions and the development architecture. This would aid NFV in meeting all the expectations of hyper-scaling while satisfying the carrier-grade requirements. Prime challenges include the following issues. Table 2.1 summarizes this section.

2.5.1 VNFCs Networking Complexity

NFV microservices architecture is based on the creation of (as many as needed) small independent VNFCs that are chained together using various web-based protocols. This approach can result in complex network activities that are difficult to manage and rapidly impose a negative effect on network manageability. Real-world applications can be decomposed into hundreds of microservices and tens of thousands of running instances, as the case with Netflix and Twitter [5] [6]. VNFCs provide networking services that handle various networking traffics and latency-sensitive workloads. Therefore, networking complexity escalates further on various levels. The network chaining complexity is a challenge that NFV-microservices should overcome through intelligent networking management possibly with SDN integration [7].

2.5.2 VNFCs Service Discovery

Despite the benefits that microservices architecture introduces to NFV, VNFCs management and development are still intricate challenges. A task such as the deployment of applications is trivial with monolithic applications but with microservices architecture additional subtasks, it becomes a complicated job. Software development and information technology operations (DevOps) tools along with containers have become mature enough to automate the complicated development on remote servers, such as one-click install applications in a cloud environment [8]. However, service discovery of VNFCs is a major hurdle that impedes the scalability of the NFV application and platforms. As VNFCs scale on-demand in a cloud environment, a real-time automated service discovery mechanism should be developed to create dynamic service chains to permit dynamic scaling of VNFs.

2.5.3 VNFCs Service Monitoring, Logging, and Meta-Data Collection

Typical NFV applications are carrier-grade in nature, and they thrive on high QoS. Real-time metrics and meta-data should be collected and processed on-the-fly to facilitate the NFV service entities (VNFs and VNFCs) orchestrations that achieve the desired QoS. Therefore, guaranteeing NFV application QoS is a challenge with microservices architecture. The orchestration and management entities in the NFV platform require clear visibility of the collected system metrics data to perform versus VNFCs health checks. Further analysis of VNFCs health checks can craft the NFV provided service topology, but any variation in the performance metrics across various VNFCs or NFV infrastructure (NFVI) resources hinders this capability. With the on-demand automated scaling ability and delay-sensitive VNFCs services, collecting and analyzing the generated metrics and meta-data across NFV microservices platform to give a holistic view of services chains and networks control flows remains an open issue.

2.5.4 VNFCs Security

Implementing VNFCs with microservices architecture brings new security challenges that did not face the traditional monolithic applications. These security challenges get exacerbated due to the extensive usage of various communication channels between all the VNFCs that create more roads for data hijacks and interception while on transit. For instance, establishing mutual trust and distributing components secrets are major security concerns [9]. Implementing all the security measures on a hyper-scale microservices intensifies the security challenges.

Table 2.1: Challenges and Solutions of NFV microservices architecture adoption

<table>
<tr><th>Challenge</th><th>Description</th><th>Solutions/Recommendations</th></tr>
<tr><td>VNFCs Networking Complexity</td><td>VNFCs as microservices are chained together using various protocols, mainly web-based protocols. This approach can result in a complex network activity that can rapidly increase manageability complexity with a higher risk of network exposure to security issues.</td><td rowspan="6">VNFC application states should be extracted and reserved in data stores (Persistence Centralization).
NFV platform should utilize SDN while implementing the following functions within the controller:
Decentralized governance
Governor units
Network segmentation
Continental federations
VNFCs should be logically grouped into various functional groups and serving units.
An optimal placement of orchestration entities should be provided.
Various VNFCs structures that comply with service availability forum (SA-Forum) standards to achieve the carrier-grade high-availability requirements should be defined.
Redundancy models and automated management of the replicas at the network segments level should be provided.
An efficient SDN query collision resolution should be provided.
A virtual centralized network-provisioning layer especially for the operations support system (OSS) should be provided.</td></tr>
<tr><td>VNFCs Service Discovery</td><td>Real-world NFV applications can be decomposed into hundreds of microservices (VNFCs) and tens of thousands of running instances. Service discovery challenge is a major hurdle that can impede the scalability of the NFV applications and platforms.</td></tr>
<tr><td>VNFCs Service Monitoring, Logging, and Meta-Data Collection</td><td>NFV applications are carrier-grade in nature that thrives on high QoS. Real-time metrics and meta-data are needed to be collected and processed on-the-fly to facilitate the NFV orchestration and achieve the desired QoS.</td></tr>
<tr><td>Infrastructure Convergence</td><td>A converged infrastructure that drives software-defined infrastructure in modern DC introduces challenges for NFV microservices architecture.</td></tr>
<tr><td>Routing Convergence</td><td>Existing routing protocols cannot keep up with the hyper-scale DCs in terms of scalability and efficiency. Supporting NFV applications along with the current load of cloud applications is a challenge for all cloud service providers.</td></tr>
<tr><td>Placement of VNFCs</td><td>The criterion used to place the VMs and containers on physical servers is the main contributor to the increase in the signaling traffic between servers. Therefore, having the optimal allocation for the VNFCs is essential to satisfy the carrier-grade requirements.</td></tr>
</table>

2.5.5 Infrastructure Convergence

The convergence of infrastructure is a promising approach currently being utilized in modern DCs to allow the ICT service providers to scale their infrastructure with efficient resource utilization [10]. Converged infrastructure drives the software-defined infrastructure in modern DC, such as Google DCs [11]. However, this kind of computing infrastructure is not flawless. Some of the challenges that should be addressed in software-defined infrastructure to enable NFV microservices architecture are as follows.

a) Computing resources convergence: Converged infrastructure includes a variety of computing resources in hosts. Various standards, communication types, file system protocols, and interface buses are used to connect hosts over COTS networking equipment. DC operators have the exclusive control rights of the network leaving the users with narrow to no exposure to the control functionalities of the underlying network infrastructure. With this limitation of network control exposure, users cannot optimize VNFCs to the best performance.
b) Networking resources convergence: Converged infrastructure combines all kinds of traffics into unified infrastructure without any segregated network. This approach of unified network infrastructure imposes risks on high priority traffics. Applying QoS and traffic separation through various networking bearers occurs through network adapters and switch partitioning. Although this approach is a solution, it introduces various manageability and traffics processing challenges especially in a virtualized environment. In a virtualized environment, the physical network adapters are shared between various applications, such as VNFCs and DC management entities that should deliver their services in real-time.

Simply providing more bandwidth in a converged infrastructure is not a solution to host NFV applications. DC infrastructure orchestrators should integrate and expose various network-controlling functionalities to maintain the desired QoS and assure the interoperability of VNFCs.

2.5.6 Routing Convergence

Multiple distinct architecture choices can be used when designing a data center. Each aims at minimizing the resources required to suit the needs of the cloud service providers. It is imperative that cloud service providers are continuously striving to improve their own hardware and software networking infrastructure. Google has gone the extra mile and developed proprietary networking protocols to manage its traffic routes [11]. Existing routing protocols cannot keep up with its hyper-scaled DCs in terms of scalability and efficiency. Supporting NFV applications along with the current load of cloud applications is a challenge for all cloud service providers. They should take a step back and decide on the conflict resolution techniques that be used. In addition, the adoption of microservices architecture with NFV applications requires new approaches at the levels of network hardware and software infrastructure specifications. Previously, the use of local area networks (LAN) was sufficient for enterprises when their servers were placed in close proximity. With the wide adoption of cloud computing infrastructure, VLANs used to meet the network demands and create multiple broadcast domains. However, classic VLANs are limited to the 12-bit ID field, which does not satisfy the hyper-scaling level of cloud demands. This led to the emergence and development of generic routing encapsulation (GRE) and virtual extensible LAN (VxLAN). VxLAN and GRE provide virtual LAN connectivity on a hyperscale over Layer 3 networks. Layer 3 networking equipment (routers) is grouped into various logical groups called autonomous systems (ASs). The latter usually use open shortest path first (OSPF) protocol to exchange routing information among group members and border gateway protocol (BGP) to exchange information with other ASs. When looking closely at these two techniques, OSPF and BGP have evolved to serve the current internet networks with great success. However, the increase in the number of virtualized applications using virtual machines (VMs) and containers has imposed challenges to the current routing protocols. VMs and containers are entities added and dropped out on the fly to meet the cloud application dynamic workloads. These VMs and containers are mobile; they can migrate from one serving node to another in real-time [12]. With these properties, VMs and containers highly rely on the network traffic mobility and low-latency. Common routing protocols are yet to be proven to serve efficiently this kind of workloads because their routing convergence is measured in seconds. Adding NFV application to the existing cloud workload can disrupt the underlying network because NFV adds hyper-scale overlay networks served by VNFCs. This begs the question: how can SDN emerge as a solution to pave the way for NFV with hyper-scaling VNFCs? It is a challenge for the SDN controller. A first step would be deploying distributed SDN controllers to handle multiple network federations routing convergence, but this area requires further investigation to converge on implementation techniques.

2.5.7 Inter- and Intra-connecting VNFCs

Classical approaches for connecting network functions on-premises are achieved through direct connections or through layer 2 (L2) switches. However, in a virtualized environment, various inter- and intra- connections approaches can be held; they are illustrated in Figure 1:

a) Two VNFCs are on the same physical server and on the same virtual switch (vSwitch).

b) Two VNFCs are on the same physical server but on different vSwitches.

c) Two VNFCs are on different physical servers.

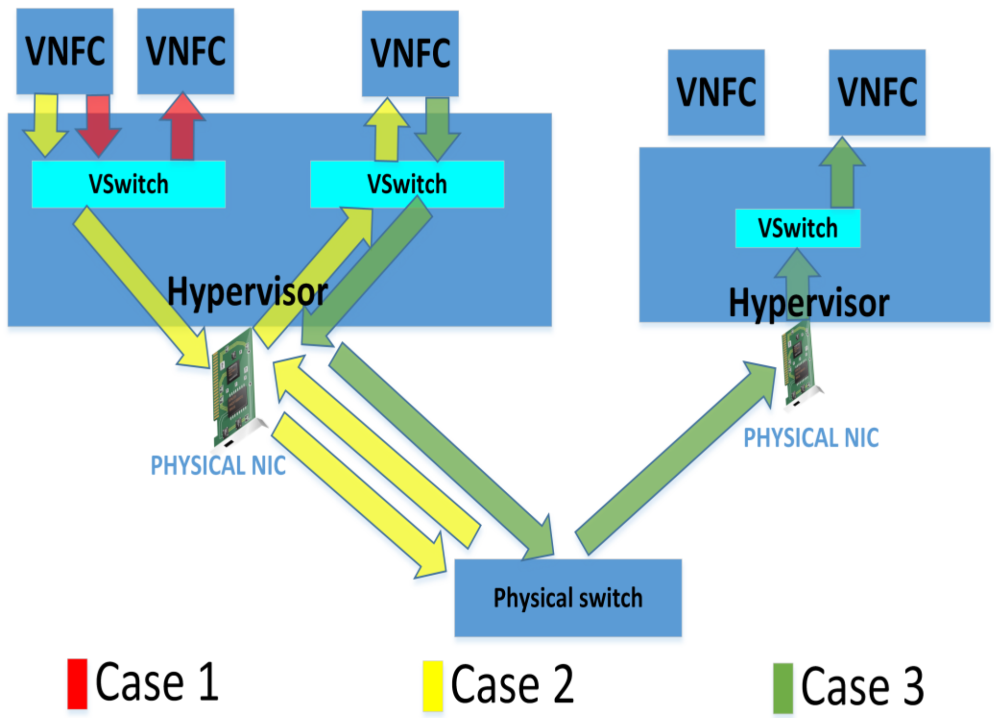

Figure 2.1: Inter- and Intra- connections of VNFCs

Each of the aforementioned cases of VNFCs connections has its own advantages and disadvantages. The VNFCs establish virtual connections through the virtual network interface controllers (vNIC), which can introduce various hops spanning tree. Optimized traffic routing and VNFCs placements should be used to monitor and minimize the network traffic latency. Single-root input/output (I/O) virtualization (SR-IOV) compliant NICs are considered as a solution to eliminate the intermediate virtual network hops, but they can hinder the VNFCs mobility in a virtualized environment.

2.5.8 Placement of VNFCs

The criterion used to place the VMs and containers on physical servers is the main contributor to the increase in the signaling traffic between servers. The VMs and containers allocation is one of the main factors that affect the carrier-grade application requirements such as QoS, reliability, and high availability. Migrating networking functions to VNFCs microservices is a challenging process because these VNFCs will be executed either within VMs or within containers running on COTS servers in DCs and should satisfy the strict carrier-grade requirements. Therefore, having the optimal (or as close to optimal as possible) allocation for the VNFCs is an indispensable step to satisfy the QoS requirements.

ETSI has defined a basic framework architecture that does not have a VNFCs placement management entity [1]. The virtualization orchestrator handles the VNFCs mapping to hosts. The orchestrator is either managed by the cloud service provider or is delegated to VNFCs owners. Furthermore, VNFCs placement directly affects the service chains' routing decisions. This can have a critical impact on the service level agreements (SLAs) in which the cloud service providers guarantee computing resources performance and availability. However, the existing SLAs do not guarantee the carrier-grade application performance with five nines (99.999%) of service availability, which is a critical requirement for virtualized carrier network functions. Therefore, cloud tenants should orchestrate the VNFCs deployment and management in order to achieve the desired QoS. For example, Amazon web services (AWS) are utilized by Netflix to serve the hyper-scale user base that is responsible for 35.2% of North America networking traffic [13]. For Netflix to achieve its desired QoS with high service availability, it has developed and contributed to various open-source software entities. Netflix use case is an example of how cloud tenants can introduce their own optimization techniques and approaches to hyper-scale their applications without sacrificing QoS.

VNFCs placement and management are more complex compared to the current cloud applications. This means that the techniques used by the leading companies who have developed the cloud application architectures are not sufficient to orchestrate the NFV platforms. VNFCs are networking function services that overlay networks and process networking packets in real-time. Therefore, any potential error or service degradation can escalate issues at various levels of the substrate and overlay networks and can disrupt any dependent services. These issues are on the horizon of the IT and DevOps pioneer enterprises. For instance, the cloud services of Apple iCloud, iTunes, and other products face disruption with an outage of 4 hours in 2015 due to an internal DNS error [14].

Having schedulers agnostic of NFV application intricacies may result in inefficient VNFCs placements. Considering this, service chained VNFCs can for some reason be scheduled on hosts where delay constraints are violated. This placement can hinder the NFV application services from scaling and offloading traffics between VNFCs. In light of the previous points, it is a necessity to associate the NFV microservices architecture with a carrier-grade NFV-aware scheduler that defines the service chain's computational paths to enhance the scalability and traffic offloading of the application service. The NFV-aware scheduler would optimally be defined as a management entity within the cloud orchestration platforms to ensure that the NFV services can serve dynamic workload while satisfying all carrier-grade requirements.

2.6 VNFCs Placement Modelling

In order to provide a scheduling solution that satisfies the SLA and QoS requirements, it is necessary to understand the cloud model. The cloud infrastructure consists of interconnected DCs distributed across different geographical areas. Racks are the building blocks of the DC, and they are intra-connected through aggregated switches. They host sets of servers with different resources capacities that are grouped in shelves. Servers belonging to the same rack are connected through the same networking device, top of the rack (TOR) switch. The topology of the network connecting the servers determines the latency constraints between them. By recognizing and modeling various delays between servers, DCs can be divided into different latency zones. As for the VNFCs instances, they are executed within VMs and containers that are mapped to the physical servers by the cloud orchestrator. As mentioned in previous sections, NFV applications typically provide their services through various chained VNFs, which are defined as several VNFCs. These chains determine the dependency relations between the VNFCs. The inherited relations are associated with delay tolerance and communication bandwidth attributes that are defined at the abstracted service representation level. The service computational path is restricted by the delay tolerance constraints, which determine the maximum allowed latency between VNFCs instances at which this path outage is declared. Therefore, maintaining the maximum number of computational paths requires optimal NFV-aware scheduling models and algorithms. Therefore, the following constraints should be satisfied:

2.6.1 Resources Capacity Constraints

These constraints are used to eliminate servers that do not satisfy the resources demands of the VNFCs.

2.6.2 Network Delay Constraints

These constraints discard the servers that violate the delay tolerance between VNFCs.

2.6.3 Availability Constraints

These constraints select the servers that satisfy the following:

a) Affinity Constraint: defines the set of VNFCs that can reside on the same hosting server.

b) Anti-affinity Constraint: defines the set of VNFCs that should reside on different servers. Usually, these VNFCs can tolerate higher outage than the co-located VNFCs.

2.6.4 Redundancy Constraints

These constraints define the number of redundant VNFCs and their redundancy model type. The redundancy models are highly correlated with the cloud environment metrics, such as the spin-up time of a VM or container.

2.6.5 Anchors Constraints

VNFCs anchors are defined by the functional dependencies that exist between the VNFCs microservices. Dependencies may introduce network hierarchy limitations between the VNFC and its anchors.

2.6.6 Orbital Area Constraints

The orbital area is defined by the region where the VNFC can be placed. This area is bounded by the VNFC anchors' constraints associated with the service chain. A VNFC can have multiple peers and dependents in a service chain. Therefore, the orbital areas and distances must be carefully calculated to enable further elastic scalability of the NFV service. Figure 2.2 illustrates the conceptualization of the VNFCs' anchors in relation to the VNFC orbital area. It demonstrates the placement criterion for a VNFC where the dependents placements act as anchors and dictated its placement orbital area.

2.7 VNFCs Placement Simulation

The NFV-aware scheduler should generate optimal placements of VNFCs to pave the way for a carrier-grade NFV service. For this purpose, a mixed-integer linear programming

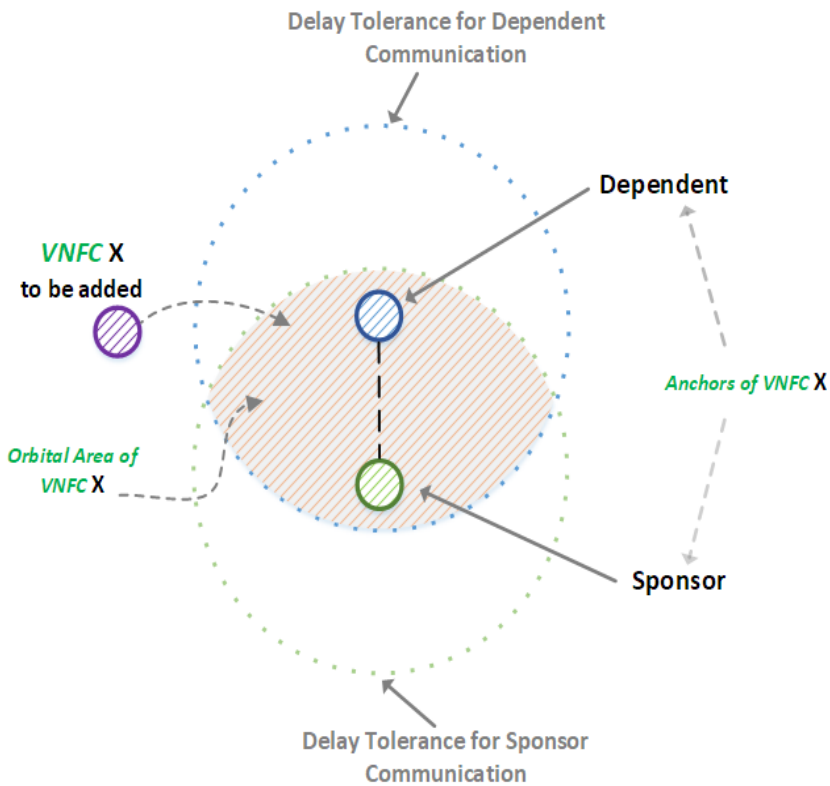

Figure 2.2: The orbital area of a given VNFC

(MILP) model is formulated based on the aforementioned constraints and with the following objective function:

$$\text{Minimize} \sum_{m=0}^{N_{MME}} \sum_{s=0}^{N_{SGW}} dMS_{ms} + \sum_{m=0}^{N_{MME}} \sum_{h=0}^{N_{HSS}} dMH_{mh} + \sum_{s=0}^{N_{SGW}} \sum_{p=0}^{N_{PGW}} dSP_{sp}$$

Where:

dMS_{ms} = Communication delay between $VNFC_m$ of type MME[1] and $VNFC_s$ of type SGW[2].

dMH_{mh} = Communication delay between $VNFC_m$ of type MME[1] and $VNFC_h$ of type HSS[3].

dSP_{sp} = Communication delay between $VNFC_s$ of type SGW[2] and $VNFC_p$ of type PGW[4].

N_{MME} = Total number of VNFC instances of type MME.

N_{HSS} = Total number of VNFC instances of type HSS.

N_{SGW} = Total number of VNFC instances of type SGW.

N_{PGW} = Total number of VNFC instances of type PGW

Virtualized evolved packet core (vEPC) is used as a use case in the simulation. vEPC has been introduced by 3GPP as a simplified all-internet-protocol (IP) core network architecture [15]. vEPC is developed to unleash the full potential of radio access technologies. It combines the leading IP infrastructure and mobility to enable mobile broadband services and applications. Table 2.2 summarizes the input data of the model. Given the available

---

[1] MME is the mobile management entity in the EPC.

[2] SGW is the serving gateway in the EPC.

[3] HSS is the home subscriber server in the EPC.

[4] PWG is the packet gateway in the EPC.

computing processing power and the computational complexity of the MILP model, the dataset is defined to generate the simulation results within a reasonable time. The delay tolerances between entities are based on data center network latency measurements as defined in [16].

The MILP model is implemented using the IBM ILOG CPLEX optimization studio and the greedy algorithm is implemented using Java. A virtual machine with 12 vCPU cores and 64 GB of memory is used to run the simulation environment. We have compared the NFV-aware scheduler with a greedy algorithm. The corresponding results are shown in Figure 2.3 and Table 2.3.

Table 2.2: The model input data

Input Variable	**Value**
Physical servers	20 servers
MME VNFC	3 Instances
HSS VNFC	2 Instances
SGW VNFC	2 Instances
PGW VNFC	3 Instances
Delay tolerance between MME and HSS	320 µs
Delay tolerance between MME and SGW	400 µs
Delay tolerance between SGW and PGW	120 µs

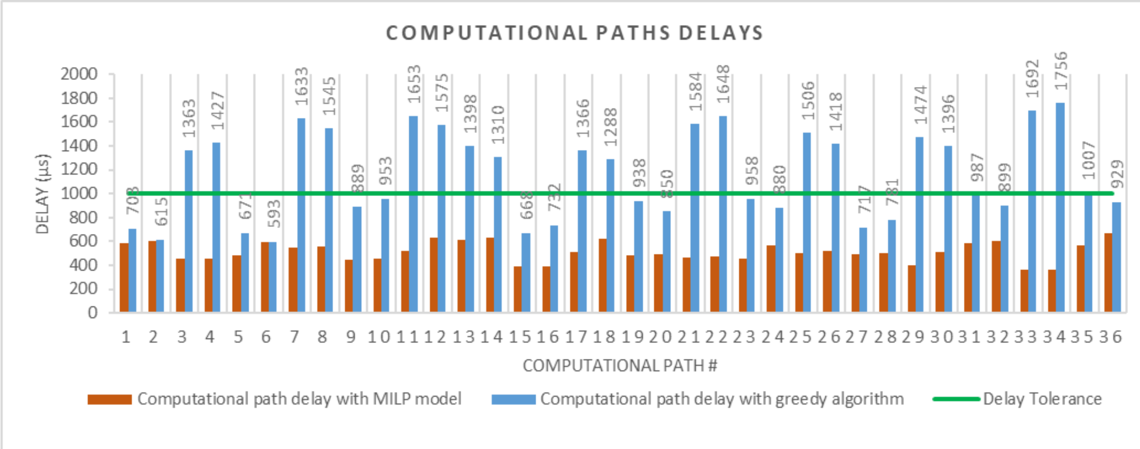

Figure 2.3: Computational paths delays

The MILP model generates the optimal placements that satisfy all the aforementioned constraints while minimizing network delays. These placements maximize the number of the available computational paths that represent the VNFCs service chains.

This objective is achieved by placing the VNFCs instances on the hosts with minimum connection delays, which provides valid connections for the computational paths.

Increasing the number of computational paths can be quantified by the number of participating members in a functional group of a VNFC instance. All the functional group members should share the same VNFC instance type and reside in the same orbital area. The higher the number of participating members in a functional group, the better it becomes. Table 2.3 shows the count of the functional groups' members that are generated from the MILP model and the greedy algorithm placements. The MILP model achieves higher functional group members count compared to the greedy algorithm.

The NFV provided service can achieve better performance and availability using the MILP model placement algorithm than the greedy algorithm. From the perspective of performance, the MILP model allows the functional group to offload traffic between higher

VNFCs members; however, it is not the case with the greedy algorithm. From the perspective of availability, the MILP model provides better availability to the functional group compared to the greedy algorithm because the MILP model has higher members count; these members act as redundant components that can take over the workload upon a failure of a VNFC instance.

Table 2.3: Functional group members

VNFC Instances	**MILP Model** **Functional Group Members Count**	**Greedy Algorithm** **Functional Group Members Count**
MME #1	3	2
MME #2	3	1
MME #3	3	2
HSS #1	2	1
HSS #2	2	1
SGW #1	2	1
SGW #2	2	0
PGW #1	3	1
PGW #2	3	2
PGW #3	3	2

In addition to the increase in the count of the VNFCs functional group members using the proposed MILP model, the results show that the computational paths' delays are minimized compared to the greedy algorithm as depicted in Figure 2.3. Minimizing the VNFCs computational paths' delays is paramount for the VNFCs management entities. The difference between the delay tolerance and the computational paths' delays allow the management entities to apply various policies on the systems. These policies vary according to the intentions of the network service providers, such as green or advanced security-based analysis policies.

2.8 Conclusion

NFV is the technology revolutionizing the ICT industry by implementing network functions as software-based applications running on COTS hardware. It adopts the IT virtualization platform benefits and innovations. The industry and academic researchers are exploiting virtualization technology to simplify and enhance the NFV platforms in order to pave the way for ICT industry wider adoption. To unleash all the advantages of NFV, various challenges should be overcome. Therefore, the leading ICT service providers, equipment vendors, and academic researchers should be aware of NFV challenges and explore new approaches to overcome them.

This chapter discussed the possibility of adopting microservices architecture in NFV to enable hyper-scaling services. To this end, various challenges were identified and discussed. Anticipated solutions for these issues were provided as well. The chapter introduced a detailed VNFCs placement challenges study and proposed an NFV-aware scheduler design. The latter scheduler was evaluated in terms of the MILP model to show the potential advantages of optimized VNFCs placement in a virtualized environment.

References

[1] ETSI, "Network Functions Virtualisation (NFV); Virtualisation Technologies; Report on the Application of Different Virtualisation Technologies in the NFV Framework," ETSI GS NFV Eve 004 Version 1.1.1, 2016.

[2] ETSI, "Mobile-Edge Computing – Introductory Technical White Paper," Solutions Reference Architecture, 2014.

[3] Intel, "Evolved Packet Core (EPC) For Communications Service Providers," Issue 1, White Paper, 2016.

[4] Ericsson, "Ericsson Mobility Report," https://Www.Ericsson.Com/Assets/Local/Mobility-Report/Documents/2017/Ericsson-Mobility-Report-June-2017.pdf, Report, 2017.

[5] Netflix, "First Netflixoss Meetup," Http://Techblog.Netflix.Com/2013/02/First-Netflixoss-Meetup.html, February 2017.

[6] Twitter, "The Infrastructure Behind Twitter: Efficiency and Optimization," https://Blog.Twitter.Com/2016/The-Infrastructure-Behind-Twitter-Efficiency-and-Optimization, February 2017.

[7] Heavy Reading, "Service Chaining In Carrier Networks," White Paper, 2015.

[8] Digital Ocean, "One-Click Install Apps," https://Www.Digitalocean.Com /Community/Tags/One-Click-Install-Apps?Type=Tutorials, February 2017.

[9] T. Yarygina and A. H. Bagge, "Overcoming Security Challenges in Microservice Architectures," IEEE Symposium On Service-Oriented System Engineering (SOSE), 2018, pp. 11-20.

[10] Cisco and Netapp, “Converged Vs. Hyper-Converged Infrastructures: Understanding the Impact on your Organization,” White Paper, 2016.

[11] Nutanix, “The Secret to Google’s Datacenter and Why It Matters to Every Enterprise,” https://Www.Nutanix.Com/2014/02/20/The-Secret-To-Googles-Datacenter-And-Why-It-Matters-To-Every-Enterprise/, February 2017.

[12] M. Jammal, H. Hawilo, A. Kanso, And A. Shami, “Mitigating The Risk Of Cloud Services Downtime Using Live Migration And High Availability-Aware Placement,” IEEE International Conference On Cloud Computing Technology And Science (Cloudcom), December 2016, Pp. 578-583.

[13] Sandvine, “2016 Global Internet Phenomena: Latin America and North America,” Report, 2016.

[14] CNBC, “Apple: Services Back After Major Outage,” Http://Www.Cnbc.Com/ 2015/03/11/Some-Apple-Services-Suffering-Outages.Html, February 2017.

[15] ETSI, “Digital Cellular Telecommunications System; Universal Mobile Telecommunications System (UMTS); LTE; Network Architecture,” 3GPP TS 23.002 Version 11.6.0, Release 11, 2013.

[16] C. Guo Et Al., “Pingmesh: A Large-Scale System for Data Center Network Latency Measurement and Analysis,” ACM Conference on Special Interest Group on Data Communication, August 2015, pp. 139-152.

Chapter 3

3 NFV/SDN-based vEPC Solution in Hybrid Clouds

3.1 Introduction

The demand for high-bandwidth network connectivity has been growing significantly over the past few years. It has gained further momentum with the surge in the number of Internet-connected mobile devices ranging from smartphones, tablets, and laptops, to sensor networks and machine-to-machine (M2M) connectivity. This amplification of networking traffic has exceeded the capacities of mobile operators' networks. Because network traffic is expected to intensify further [1] in the coming years, network service providers have little choice but to invest in bandwidth-oriented infrastructure to satisfy the demand. However, while studies show that the return on capital with such investments is minimal [2], network upgrading highly depends on the infrastructure upon which the network relies. This dependency regarding the exponential cost of network equipment diminishes the revenue margin of the network operators when an upgrade or new service is released. Network operator challenges are not bound by the cost of expensive hardware devices alone but are also impacted by increasing energy costs coupled with the lack of personnel with expertise to design, implement, and orchestrate a progressively complex hardware-based infrastructure. Moreover, maintenance of the network infrastructure is another primary concern of operators. The scope of these issues is not limited merely to revenue loss; ripple effects manifest through lags in time-to-market, as well as in general hindrances to innovation within the telecommunications industry. Therefore, network operators seek to reduce—or even forfeit—their dependency on proprietary hardware.

To achieve these targets, network service operators are investigating the integration of virtualization technology within the telecommunications industry. Virtualization technology emerged as a mean for information technology (IT) specialists to achieve more effective capital investments with higher returns on capital. Virtualization also facilitates the hardware and software decoupling process [3]; for example, multiple isolated software programs can share the underlying hardware. Virtualization enhances resource utilization, reduces capital expenditures (CAPEX) and operating expenditures (OPEX), and yields many advantages offered by the cloud service. As an initial step, a group of seven telecom operators established an industry specifications group for network functions virtualization (NFV) under the European Telecommunications Standards Institute (ETSI). They revealed their solution in October 2012 [4], which prompted several telecom equipment providers and IT specialists to subsequently join the group.

NFV is the concept of migrating the network functions from dedicated hardware equipment to software-based applications and is the technology that can best take advantage of the IT virtualization evolution. Equipment and software components will be consolidated on standardized IT platforms (high volume servers, switches and storage), while network functions within the proprietary hardware can be simultaneously decoupled. Through NFV, network functions can be substantiated in various locations, such as data-centers, network nodes, and end-users premises as the network requires [2]. Virtualized network functions (VNFs) will be mainly hosted within cloud environments by utilizing cloud computing services [2]. However, data-center operators are facing a tremendous increase in the number of servers and virtual machines, which in turn increases server-to-server communication traffic. In order to tackle these challenges, data-center operators require a network that is efficient, flexible, agile, and scalable to the servers connections. Software-defined networking (SDN) has been suggested as a solution to the above-mentioned challenges. SDN operates on an aggregated and centralized control plane and has been used to resolve network management and control problems [5]. The main idea behind SDN is to separate the forwarding/data plane from the control plane while providing programmability on the control plane.

In this chapter, we have assessed the next generation of mobile core network entities with the intent to virtualize and adapt them for cloud deployment. Several approaches exist in the literature (as discussed in Section III), but these approaches address solutions through private cloud interfaces, which are completely owned and controlled by the network service providers. Furthermore, the previous literature does not consider the co-existence of different applications hosted with the virtualized mobile core network entities in the cloud platform. In order to allow a mobile core network to exploit all the advantages provided by cloud services, we have proposed a solution based on two technologies: NFV and SDN. The proposed solution will facilitate the co-existence of VNF with the existing IT apps in multi-tenant private and public cloud interfaces, as shown in Figure 3.1.

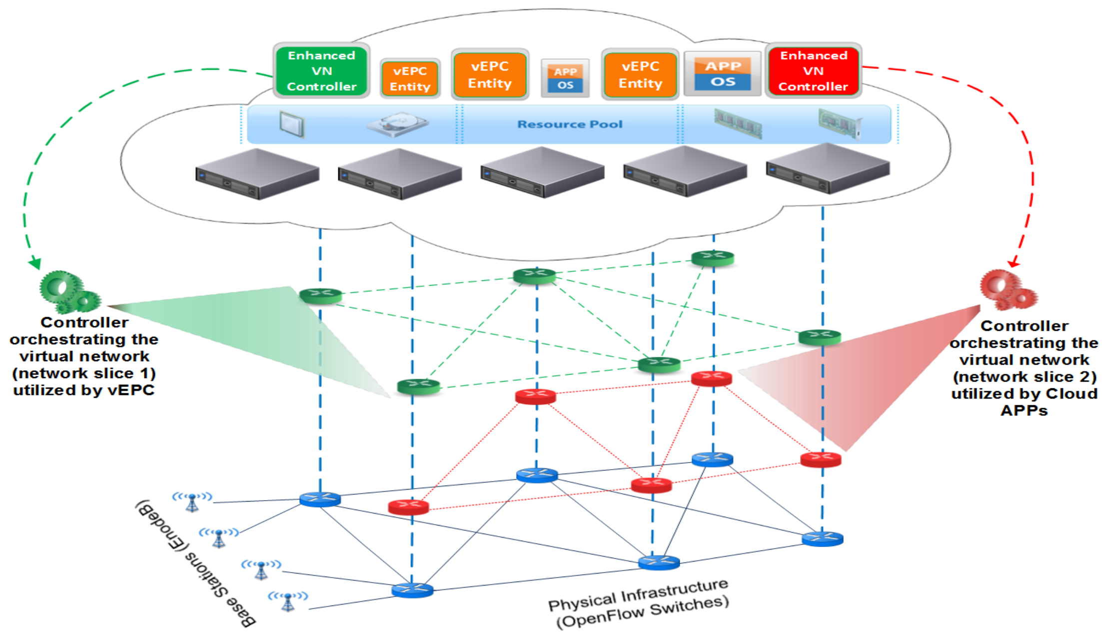

Figure 3.1: Co-existence of vEPC with cloud applications

Moreover, it will ensure that the VNF will meet the desired carrier-grade quality of service without sacrificing the scalability of the VNF.

3.2 Evolved Packet Core

Mobile broadband networks are needed to exploit the advantages of Internet connectivity and mobile services. For service providers to offer a mobile connectivity platform that supports broadband speeds and high data traffic capacity, major changes need to be undertaken in mobile networks. Granted, mobile networks have evolved to support higher data rates, Internet Protocol (IP), and packet-switching protocols. Long Term Evolution (LTE) and High-Speed Packet Access (HSPA) have provided further high data rate radio access technologies. As for the core network, evolved packet core (EPC) has been introduced by 3GPP in release 8 as simplified all-IP core network architecture. EPC has been developed to unleash the unrestrained potential of these advanced radio access technologies; it is designed to allow complete mobile broadband services and applications by combining the leading IP infrastructure with mobility. Moreover, EPC is designed to support different radio access technologies such as 2G, 3G, WLAN, WiMAX, and fixed access networks (Ethernet, DSL, cable and fiber) [6].

EPC's main architecture operates according to two main principles. The first guiding principle involves decoupling the control plane from the user data plane. This concept was introduced to satisfy various requirements, such as facilitating the control plane to scale according to the number of users being served, while at the same time the data plane scales according to applications and services being accessed. Also, the decoupling is required to implement optimization techniques independently for control signaling and user data. Moreover, decoupling the control and data planes allows service providers to implement their instrumentations more flexibly according to the network architecture [7]. The second principle: EPC has to maintain a flat architecture whereby user data traffic can be processed using as few nodes as possible. Because data traffic demands are growing rapidly with new service releases, the flat architecture permits a cost-efficient scaling of the data plane [8].

The basic entities of EPC to support IP connectivity over LTE access are the following:

1- Mobility Management Entity (MME):

MME is the main control plane entity in the LTE network. It manages the mobile users' access to the LTE network. MME is responsible for assigning network resources like the serving gateway and manages mobility conditions that support roaming, paging, and handovers. It is also responsible for the identification and authentication of the user by interacting with the home subscriber server.

2- Serving Gateway (SGW):

SGW is responsible for routing and forwarding user data packets from and to the base station. It also acts as a mobility anchor for the user data plane during inter-mobility handovers, and between LTE and other access technologies. SGW decides which packet data network gateway will serve the user.

3- Packet Data Network Gateway (PDN-GW) (PGW):

PGW ensures the connectivity of the user data plane to the external networks. PGW contains the policy enforcement functions, packet filtering, and can implement online and offline charging functions.

4- Home Subscriber Server (HSS):

HSS is the central user information database. It provides information about user authentication and authorization for different network functions.

5- Policy and Charging Rules Function (PCRF):

PCRF is responsible for applying and deciding the policies and charges in real-time for each service and user. It automatically decides policies and charges that should be enforced.

3.3 Virtualizing EPC

LTE technology has revolutionized the telecommunication industry. It has enhanced mobile data traffic by significantly improving the quality of the user experience. This increase in mobile data traffic has presented challenges in expanding and enhancing network services to satisfy the demands of users. Users, for example, expect to have a similar (or better) experience from fixed network access. Furthermore, machine-to-machine communication service reliance on mobile networks has intensified network demands. This issue has led service providers to make investments (CAPEX and OPEX) that defy financial sense. From this perspective, it has becoming essential to have a flexible, robust, and easily manageable network—a network that could be scaled on-demand in real-time, and be capable of automated management. Virtualizing EPC will offer these solutions for service providers and allow them to exploit control and data plane decoupling for better resource utilization and higher network performance. Virtualization permits scaling the control plane independent of the data plane to achieve optimized configuration for their networks. The result is that service providers will be able to instantiate multiple EPC entities, with different resources allocated to the control and data plane, at lower financial expenditure. Furthermore, providers will benefit from higher network service resilience, automated failure recoveries, and automated management systems supported by the NFV orchestrators [2]. Virtualization is indeed the solution that will unleash the potential of EPC.

Virtualized EPC (vEPC) has captured the attention of researchers as a solution to provide support for next-generation mobile networks. Some pioneering solutions, such as moving the evolved packet core into the cloud, are expressed in the literature. As in [9][10], they provide suggestions, from design to implementation phases, on how to migrate SDN into mobile networks. In [9], the authors present EPC in conjunction with SDN, which would permit the movement of the control plane into a data-center. In essence, they extended the scope of OpenFlow to permit the GPRS tunneling protocol (GTP) control plane to be executed through OpenFlow, which can then be implemented into a data-center. In [10], the authors proposed an SDN architecture solution for mobile core networks. This approach delineates two entities: the mobile flow forwarding engine (MFFE) responsible for the user plane, and the mobile flow controller (MFC) responsible for the control plane. The MFFE is advocated as a fully software-defined entity that differs from the OpenFlow-based network. Furthermore, they have described how the aforementioned architecture can be integrated within the legacy network. As NFV is also considered to be the technology that will enable the virtualization of network functions, it is further explored in [11][12] to be a key support system for vEPC in cloud interfaces. In [11], the authors presented a proof-of-concept implementation of the routing network function utilizing an OpenFlow-enabled network, and these VNFs provide the intelligence for routing decisions. In [12], the authors considered virtualizing EPC entities MME and HSS while trying to integrate SGW and PGW functionalities within OpenFlow SDN.

3.4 NFV/SDN Based vEPC Solution

So far the proposed solutions for virtualizing EPC are applicable mostly to small networks within a private cloud. Private clouds are groups of data-centers owned by the network service providers, and these providers have full control over the entire infrastructure (physical servers, underlying core networks, virtual environments, and orchestrators). These solutions do not address multi-tenant support and co-existence with variant cloud applications that are already utilizing the cloud. For vEPC to take full advantage of NFV technology, it should have the capacity to be deployed in a hybrid cloud. A hybrid cloud is a composite of different cloud types (private, public, and community clouds); its architecture requires both on-premises (private) and off-site (public) cloud infrastructure. Within this architectural network, service providers can host user-critical information applications in private clouds while hosting computationally demanding applications in public clouds in different geographical locations. (Refer to [3] for a comprehensive study of the cloud computing environment.) Implementing vEPC in a public cloud is challenging because the infrastructure will be controlled by cloud service providers (Google, Amazon, and so on), and be shared with different types of applications. ETSI NFV group proposed a management and orchestration framework that could be integrated with the current IT virtualization environment to enhance VNF lifecycle management and orchestration [2]. This framework was only intended to describe which entities are required to allow VNF management by the orchestrators, and how they might be integrated within operations and business support systems (OSS/BSS). ETSI NFV group does not clarify in the framework how NFV will support multi-tenant and on-demand scalability, which are objectives specified by NFV.

To increase the adoption of vEPC in the telecommunication industry, vEPC should meet all carrier grading requirements with respect to performance, fault resilience, scalability, and quality of service. To achieve the desired requirements, we propose a solution based on integrating NFV and SDN.

As for the NFV portion of the proposal, we have chosen the grouping criterion suggested by [2]. We selected this grouping criterion with the intention to minimize the signaling traffic that is generated by the EPC control plane, as signaling traffic is expected to grow 50 percent in excess of data traffic growth, according to Nokia Siemens Networks [13]. Signaling growth is flourishing owing to the multitude of new services emerging on mobile technology platforms—for example, voice-over LTE (VoLTE), and on-demand video streaming. In [2], the authors provided a detailed analysis of applying a novel VNF grouping criterion for vEPC entities. This grouping criterion significantly decreases the total signaling traffic by 70 percent in the core network, which results in the minimization of computation and networking transactions of the VNFs in the virtualized environment. The vEPC entities are divided into four virtual appliances: the first appliance combines MME with HSS frontend (FE); the second appliance groups SWG with PWG; the third appliance groups user data repository (UDR), PCRF, on-line charging system (OCS), and off-line charging system (OFCS); and the fourth appliance groups the serving general packet radio service (GPRS) support node (SGSN) with the home location register frontend (HLR FE). As for the SDN part of the solution, we selected the approach proposed by [14]. In [14], the authors introduce a solution—that is, the distributed FlowVisor (DFVisor)—that leverages multi-tenant with fine-grained QoS capabilities without sacrificing the scalability of cloud applications. FlowVisor [15] is a network virtualization solution currently being studied in the SDN community with fine-grained QoS management support for OpenFlow network fabric. It uses an OpenFlow flow matching mechanism to enable both fine-grained QoS management and network virtualization [16][17]. Although FlowVisor is efficient and widely used in current OpenFlow-based networks (particularly for virtualization with fine-grained QoS support), it has scalability issues caused by its centralized slice control model and the lack of native mechanisms for network virtualization.

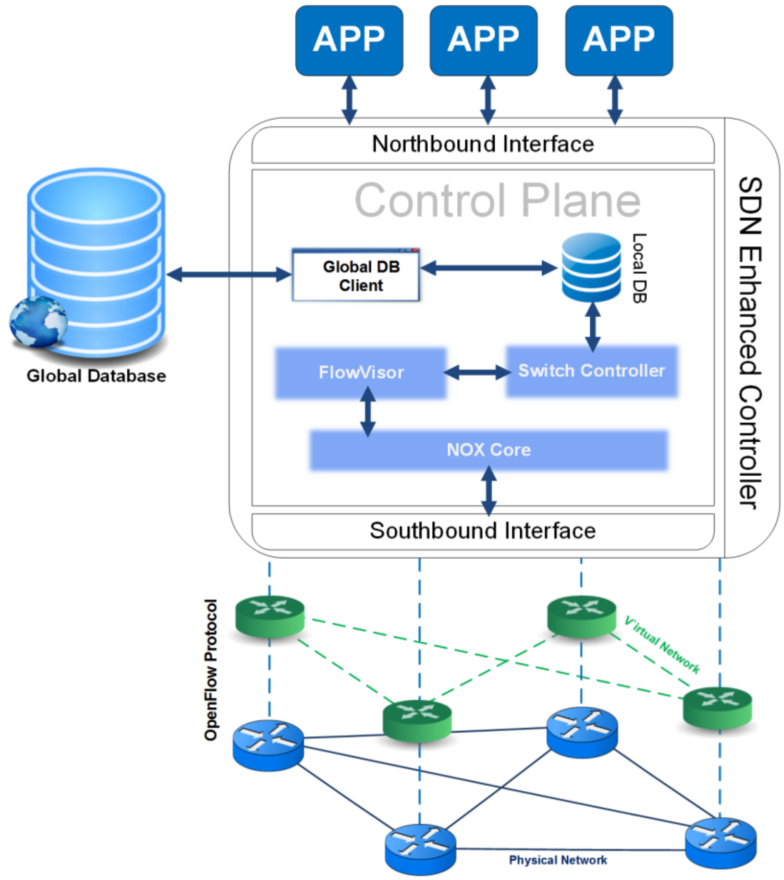

Figure 3.2: Enhanced Controller

More importantly, current pull-based flow setup and statistics-gathering, the network statistics competing with the southbound interface bandwidth with OpenFlow control flows, and the centralized slice control model in current FlowVisor cause flow setup and statistics-gathering latencies. These latencies not only limit the flow setup rate for each switch and the management granularity for each controller but also affect the virtual network scalability and QoS management. Consequently, the network requirement for latency-sensitive applications such as vEPC cannot be achieved.

DFVisor addresses these issues by providing a fully distributed SDN architecture with an enhanced OpenFlow protocol. Specifically, DFVisor improves the current OpenFlow protocol explicitly to support tunneling, such that a much larger network address space can be provided. On top of the enhanced tunneling, DFVisor also constructs multiple OpenFlow tunnels so that the flows in each OpenFlow tunnel can be forwarded using OpenFlow flow matching with fine-grained QoS support.

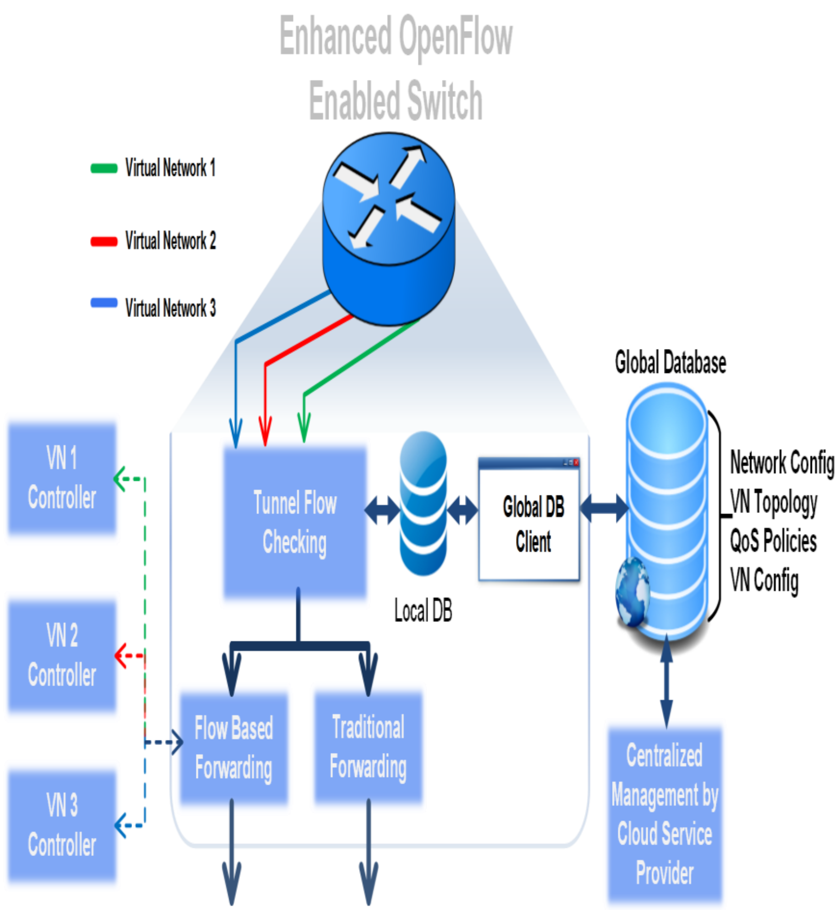

Figure 3.3: Enhanced Switch

The enhanced OpenFlow controller and switch are illustrated in Figures 3.2 and 3.3. DFVisor constructs a wholly distributed architecture, which includes a distributed synchronized two-level database that consists of a global database and multiple local databases in each switch and controller, and a distributed slice control module in each controller. This synchronized, two-level database enables a new data channel so that the network configuration and statistics information can be transferred through data synchronization via this new data channel rather than through the OpenFlow protocol over a southbound interface. The proposed distribution method not only bypasses the competing southbound bandwidth with OpenFlow control flow but also facilitates a push-based flow setup and statistics gathering mechanism to avoid the scalability limitation caused by the current pull-based flow setup and statistic gathering latencies. Distributing the centralized slice controller to be a slice control module within each controller also mitigates network failure potential and reduces the extra latencies caused by it.

3.5 vEPC Entities Placement in DFVisor

EPC consists of multiple entities with different functionalities. Virtualizing EPC by instantiating its VNFs in the cloud may have a serious effect on the performance and the quality of service. In this chapter, we propose to use the vEPC VNFs entities with the DFVisor to meet all the desired carrier-grade requirements. The placement procedure is based on analyzing the VNFs interconnections and functionalities to achieve superior performance and QoS offerings without affecting the vEPC scalability in a multi-tenant cloud. The VNFs entities' placement is illustrated in Figure 3.4.

3.5.1 MME and HSS FE

MME and HSS FE are joint components within one virtual application. HSS FE is an application that leverages all of the logical functionality towards MME for user authentication and authorization processes, but without retaining the user information database. By implementing the HSS FE with the MME, authentication, and authorization processes are carried out internally—without any data transactions through the network—which minimize the generated signaling traffic. Moreover, this grouping will allow a single MME to manage multiple SGWs, as shown in figure 3.4. Typically, The HSS FE issues a query for user information data that will be hosted in the local controller database, and stores these data temporarily in cache memory. After querying for user information, the HSS Front End (FE) acts as a complete user database and performs all authentication and authorization processes with the MME entity. After completing these processes, the resulting information will be sent to the local database hosted by the local controller. When the local controller notices the changes, new policies and configurations will be generated and enforced within SGW and PGW. Furthermore, this grouping also allows the dynamic scaling of MME without affecting the user-data plane, which is one of the EPC principles.

3.5.2 UDR, PCRF, OCS, and OFCS

The UDR, the PCRF, the on-line charging system (OCS), and the off-line charging system (OFCS) will be extending the enhanced controller proposed in [14]. The enhanced controller consists of improved Open-Flow protocol to support virtual overlay networks using tunneling procedures and a local FlowVisor controller module for local network orchestration (as illustrated in figure 3.2). It also has a global database client to synchronize the data between the local and global databases.

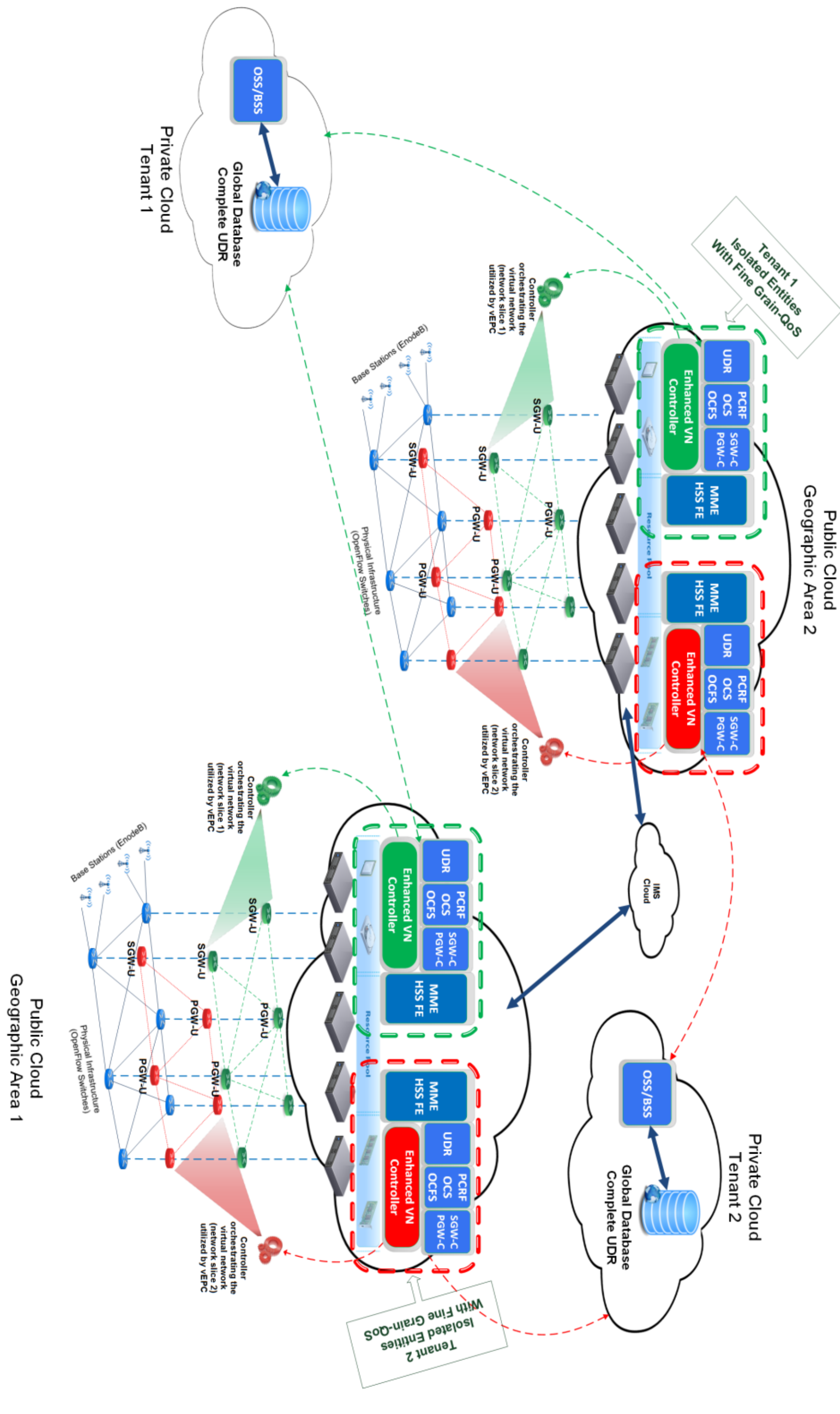

Figure 3.4: vEPC entities placement with DFVisor in public and private cloud with multi-tenant support

The local and global databases will be extended to host user information as well. DFVisor delivers a distributed, synchronized, two-level database, which facilitates high availability and fast data retrieval and processing. The global database is a distributed database sourced in Zookeeper [18], which has built-in configuration and watch services to maintain synchronization between the global and local databases, and to orchestrate the centralized network configuration and management. Zookeeper-based global databases can be deployed in a cluster form, consisting of a couple of nodes with replicated data, to avoid the single point of failure. Zookeeper can scale horizontally with scaled read throughput to support a large number of reading clients that specifically fit the data visiting model in UDR, PCRF, OCS, and OFCS entities. A Zookeeper-based global database also supports hierarchical deployment, which means that it can deploy a couple of nodes closer to users so that a global database cache—between the global database and local databases—can be enabled and the network connectivity and data reading latencies can be reduced [19]. The synchronized local databases in each controller and switch can automatically update their data from the global database through Zookeeper watch service, the result being that they can cache part of UDR to facilitate faster data processing.

PCRF, OCS, and OFCS entities also will be included as applications utilizing the enhanced controller. Having the PCRF amalgamated within the same VM with local databases that cache the user information leads to an efficient way of generating the policy functions, as the PCRF requires the user information to generate the adequate policies for each established bearer. Because the user information is pro-actively pushed to the local databases through Zookeeper watch service, this approach prevents information exchange from overwhelming the network node, minimizes the latency of policy-function generation, and speeds policy enforcement to the PGW via the controller. As for the OCS and OFCS, the OCS is used to charge users accounts in real-time (e.g., through a pre-paid credit system), whereas the OFCS charges users after the session is ended (as in billing services known as “pay as you go”). The OCS and the OFCS interact with the PCRF and controller local databases to gather information about the session and enforce charging policies to the PGW, such as terminating the communication session when the credit limit has been exceeded. PCRF, OFCS, OCF, and UDR entities will be utilizing the enhanced controller northbound interface as an internal transaction between processes, which offloads the network statistics transferring from the control Channel (the southbound interface of SDN), and improves its performance.

3.5.3 SGW and PGW

As for the user-data plane that contains the SGW and PGW, we designated them as an OpenFlow switch while hosting control plane functionalities in the enhanced controller. The control plane will host the functions-related control signaling messaging and resources management logic. The data-plane functions such as data-plane forwarding rules and tunnels matching should be supplied by OpenFlow switches. As DFVisor supports OpenFlow 1.3 with tunneling enabled, it can easily manage the tunneling mapping within a switch, which in turn maximizes the system performance. Furthermore, each type of bearer can be identified through OpenFlow flow matching and linked to a specific QoS level (defined by QoS policy in the global and local databases). This process allows the vEPC (running on DFVisor) to slice the resource of each switch logically, and assigns it to each type of bearer to make sure that the QoS is guaranteed at the switches.

3.6 Functionality Analysis

The proposed vEPC is an NFV solution based on a distributed SDN architecture and DFVisor. Compared to other SDN architectures, as shown in Table 3.1, DFVisor is the only solution that can support more than 2^{12} virtual networks in a system with thousands of servers, while at the same time supporting centralized network configuration and management without sacrificing the fine-grained QoS capabilities.

Specifically, DFVisor enhances the OpenFlow 1.3 protocol to support tunneling. With a 32-bit tunnel identification field, DFVisors can support 2^{32} virtual networks. Moreover, with the distributed synchronized two-level database system, DFVisor enables push-based flow-setup and statistics-gathering, which mitigates the 4 milliseconds of delay from a new flow setup, and the more than 10 seconds of delay from the network statistics-gathering for a ToR switch in current OpenFlow network [20], the latter of which is considered too long to support latency-sensitive applications such as vEPC. DFVisor can also reduce the network latencies by hierarchically deploying the global database such that a global database cache can be formed in a location close to users.

DFVisor provides higher scalability and availability than other SDN solutions because it is fully distributed, and no single network point of failure and performance bottleneck exists. Although the global database that is logically centralized may cause a performance bottleneck in DFVisor, the Zookeeper-based database itself can be physically distributed and scaled horizontally. A Zookeeper cluster with 13 nodes can support thousands of clients through accessing, with scaled read throughput, 250 simultaneous clients; furthermore, the number of clients supported can be increased in advance to tens of thousands or more by adding observer nodes to the Zookeeper cluster [19], which meets the read-intensive data model of UDR, network configuration, and management policy in the proposed vEPC approach. To meet the write-intensive data model in network statistics-gathering in a big scale network, DFVisor can combine a NoSQL data store (such as Cassandra) with Zookeeper, and thereby store network statistics to Cassandra to enable the fast read and write operations and scaled read and write throughputs [21].

Table 3.1: SDN architecture comparison

SDN architecture	Network config. & management	Controller	Network virtualization mechanism	Number of Virtual networks	Number of servers	QoS
FlowVisor[15]	undefined	Centralized FlowVisor controller + distributed virtual network controller	VLAN	$< 2^{12}$ =4096	< 1500	Fine-grained
AdVisor [23]	undefined	Centralized FlowVisor controller + distributed virtual network controllers	VLAN	$< 2^{12}$ =4096	< 1500	Fine-grained
VeRTIGO [24]	centralized	Centralized FlowVisor controller + distributed virtual network controllers	VLAN	$< 2^{12}$ =4096	< 1500	Fine-grained
DFVisor [22]	centralized	Distributed FlowVisor controller + distributed virtual network controller	tunneling	$< 2^{32}$	Tens of thousands	Fine-grained
SPARC [25]	centralized	Centralized master controller + distributed virtual network controllers	VLAN	$< 2^{12}$ =4096	undefined	Fine-grained
Midokura [26]	centralized	Distributed virtual controllers	tunneling	$< 2^{32}$	Tens of thousands	Best effort

Therefore, DFVisor can provide higher scalability and availability without sacrificing the fine-grained QoS, centralized configuration, and management.

3.7 Conclusion

Cloud computing is the technology that revolutionized the IT industry. It has mitigated the application's dependency on the underlying hardware and provided a virtual environment where computing resources, platforms, and services are delivered dynamically on-demand [3]. Cloud computing benefits are not limited to the IT industry alone, but rather extend to other industries to enhance business models through three concepts: availability of services wherever you go, accessibility from multiple devices, and durability of applications. The purpose is to extend cloud computing advantages to the telecommunication industry, especially with respect to mobile networks. Ultimately, we have proposed in this chapter a system architecture solution to migrate the mobile core network to the cloud. The solution is based on two technologies NFV and SDN to leverage the QoS and scalability of the VNF. Furthermore, the solution introduces placement techniques regarding vEPC entities in comparison to DFVisor, which is an enhanced approach for network overlaying in SDN environments. With the use of DFVisor the VNF co-exists with the IT applications in private and public clouds without violating the carrier-grade service quality.

References

[1] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018,” White Paper, Feb. 2014.

[2] H. Hawilo, A. Shami, M. Mirahmadi, R. Asal, , “NFV: State of the Art, Challenges and Implementation in Next Generation Mobile Networks (vEPC)” to appear in IEEE Network Magazine, Nov. 2014.

[3] M.A. Sharkh, M. Jammal, A. Shami, and A. Ouda, "Resource allocation in a network-based cloud computing environment: design challenges," IEEE Communications Magazine, vol. 51, no.11, pp. 46-52, Nov. 2013.

[4] ETSI, “Network Function Virtualization: An Introduction, Benefits, Enablers, Challenges, & Call for Action,” portal.etsi.org/NFV/NFV_White_Paper.pdf, 2012.

[5] M. Jammal, T. Singh, A. Shami, R. Asal, Li, Y., “Software Defined Networking: State of the Art and Research Challenges,” Computer Networks Journal, vol. 72, pp.74-98, Oct. 2014.

[6] ETSI, ”LTE; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (EUTRAN) access,” 3GPP TS 23.401 version 11.8.0 Release 11.

[7] J. Kempf, B. Johansson, S. Pettersson, H. Luning, T. Nilsson, "Moving the mobile Evolved Packet Core to the cloud," IEEE Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 784-791, Oct. 2012.

[8] W. Hahn, "Flat 3GPP Evolved Packet Core," IEEE Wireless Personal Multimedia Communications (WPMC), pp. 1-5, Oct. 2011.

[9] J. Kempf, B. Johansson, S. Pettersson, H. Luning, T. Nilsson, "Moving the mobile Evolved Packet Core to the cloud," IEEE Wireless and Mobile Computing, Networking and Communications (WiMob),pp.784-791, Oct. 2012 .

[10] K. Pentikousis, Y. Wang, W. Hu, "MobileFlow: Toward Software-Defined Mobile Networks," IEEE Communications Magazine, vol. 51, no. 7, pp. 44-53, Jul. 2013.

[11] J. Batalle, J. Ferrer Riera, E. Escalona, J.A. Garcia-Espin, "On the Implementation of NFV over an OpenFlow Infrastructure: Routing Function Virtualization," IEEE SDN Future Networks and Services (SDN4FNS), pp.1-6, Nov. 2013.

[12] W. Kellerer, M. Hoffmann, K. Hoffmann, E.-D. Schmidt, "A Virtual SDN-Enabled LTE EPC Architecture: A Case Study for S-/P-Gateways Functions," IEEE SDN Future Networks and Services (SDN4FNS), pp.1-7, Nov. 2013 .

[13] Nokia Siemens Networks, "Signaling is Growing 50% Faster than Data Traffic." http://nsn.com/system/files/document/signaling_whitepaper_onl ine_version _final.pdf, 2012.

[14] L. Liao, V. Leung, P. Nasiopoulos, "DFVisor: Scalable Network Virtualization for QoS Management in Cloud Computing," 10th International Conference on Network and Service Management, Nov. 2011.

[15] R. Sherwood, G. Glen, Y. Kok-Kiong, A. Guido, C. Martin, et al., "Flowvisor: A network virtualization layer." OpenFlow Switch Consortium, Tech. Rep , 2009.

[16] OpenFlow Switch Consortium, "OpenFlow Switch Specification Version 1.0. 0." archive.OpenFlow.org/documents/OpenFlow-spec-v1.0.0.pdf, 2009.

[17] OpenFlow Switch Consortium. ”OpenFlow Switch Specification, version 1.3.0” https://www.opennetworking.org/images/stories /downloads/sdn-resources/onf-specifications/OpenFlow/OpenFl ow-spec-v1.3.0.pdf, 2012.

[18] P. Hunt, M. Konar, F.P. Junqueira, B. Reed, "ZooKeeper: Wait-Free coordination for Internet-scale systems," USENIX Annual Technical Conference, Jun. 2010.

[19] H. Robinson, "Observers: Making ZooKeeper Scale Even Furthe", http://blog.cloudera.com/blog/2009/12/observers-making-zooke eper-scal e-even-further/, 2009.

[20] A. R. Curtis, C. M. Jeffrey, J. Tourrilhes, P. Yalagandula, P. Sharma, S. Banerjee, "Devoflow: scaling flow management for high-performance networks." ACM SIGCOMM Computer Communication Review, vol. 41, no. 4, pp. 254-265, 2011.

[21] A. Cockcroft, D. Sheahan, "Benchmarking Cassandra scalability on aws-over a million writes per second." http://techblog.netflix. com/2011/11/benchmarking-cassandra-scalability-on.html, 2011.

[22] L. Liao, A. Shami, V. Leung, “DFVisor: A Distributed FlowVisor Platform for QoS aware Cloud Network Virtualization,” under reviewed by IET Network Special Issue: Software Defined Networking, Oct. 2014.

[23] E. Salvadori, R. Doriguzzi Corin, A. Broglio, M. Gerola, "Generalizing Virtual Network Topologies in OpenFlow-Based Networks," IEEE Global Telecommunications Conference (GLOBECOM), pp. 1-6, Dec. 2011.

[24] R. Doriguzzi Corin, M. Gerola, R. Riggio, F. De Pellegrini, E. Salvadori, "VeRTIGO: Network Virtualization and Beyond," Software Defined Networking (EWSDN),pp.24-29, Oct. 2012.

[25] W. John, A. Devlic, Z. Ding, D. Jocha, A. Kern, M. Kind, et al., "Split Architecture for Large Scale Wide Area Networks." arXiv preprint arXiv:1402.2228, 2014.

[26] M. Akane, "Introduction to network virtualization for IaaS Clouds", LinuxCon, May 2013.

Chapter 4

4 Network Function Virtualization-Aware Orchestrator for Service Function Chaining Placement in the Cloud

4.1 Introduction

The demand for high-bandwidth network connectivity has been growing significantly over the past few years. It has gained further momentum with the surge in the number of internet-connected mobile devices ranging from smartphones, tablets, laptops to sensor networks and Machine-to-Machine (M2M) connectivity. The network traffic has exceeded the capacities of the existing mobile service providers' networks [1]. Since the network traffic is expected to increase in the near future, Network Service Providers (NSPs) should invest in bandwidth-oriented infrastructure to satisfy the demand [2]. While studies show that the return-on-capital with such investments is minimal [3], the network upgrading highly depends on the network infrastructure. This dependency along with the exponential cost of the network equipment may lessen the revenue margins of the NSPs when an upgrade or new service is released. NSPs' challenges are not only bounded to the cost of expensive hardware devices, but they are also affected by the increase in the energy costs coupled with the shortage of personnel with expertise to design, implement, and orchestrate a progressively complex hardware-based infrastructure. Moreover, maintenance of the network infrastructure is another primary concern of the service providers. The scope of these issues is not limited merely to the revenue loss but also to the ripple effects that manifest through lags in time-to-market as well as in the general hindrances to innovation within the telecommunications industry. Therefore, network operators seek to reduce or even forfeit their dependency on proprietary hardware. To achieve these targets, network service providers are investigating the integration of virtualization technology within the telecommunications industry. Virtualization technology emerges as a mean for Information Technology (IT) specialists to enhance capital investments with higher returns-on-capital. Virtualization also facilitates the hardware and software decoupling process where multiple isolated software programs can share the underlying hardware [4]. As an initial step, a group of seven telecommunication operators established an industry specification group for Network Function Virtualization (NFV) under the European Telecommunications Standards Institute (ETSI). Once they proposed their solution in October 2012, several telecommunication equipment providers and IT specialists subsequently have joined the group [5]. NFV is the concept of migrating the network functions from dedicated hardware equipment to software-based applications. NFV is the technology that can exploit the advantages of the IT virtualization evolution. Equipment and software components are consolidated on standardized IT platforms (e.g., high volume servers, switches, and storage) while network functions within the proprietary hardware can be simultaneously decoupled. Through NFV, Virtual Network Functions (VNFs) can be instantiated at various locations, such as Data Centers (DCs), network nodes, and end-users' premises depending on the network requirements [3]. Exploiting the advantages of the cloud computing services, Software Defined Networking (SDN), and NFV facilitates the opportunity to design and implement scalable, elastic, and programmable next-generation networks [6], [7]. However, the latter desired networks introduce various deployment and orchestration challenges that should be resolved to realize their benefits and pave the way for wider commercial adoption by the industry [8], [9]. ETSI defines the basic architecture standards for the NFV Management and Orchestration (NFV-MANO) framework. Each NFV networking service consists of one or more VNF [10]. VNFs implement various functionalities that provide the networking services defined by the Network Service Descriptor (NSD). According to the NSD VNF Forwarding Graph (VNFFG), the logical path connecting the VNFs is defined as a Service Function Chain (SFC). Having well-defined standard interfaces for the VNFs provides the NSPs with the freedom to design and implement their proprietary services to meet the customers' needs while avoiding vendor lock-in of their NFV platforms. Moreover, it drives the innovation and evolution of the NFV networking services and provides the capability of flexible management and orchestration of the VNFs lifecycle based on functional/non-functional constraints. Despite all the significant literature studies on NFV, VNFs deployment and orchestration still need to be further investigated and exploited to satisfy the carrier-grade requirements for the networking services [11]–[14]. Researchers have been addressing various aspects of NFV challenges. For instance, VNFs orchestration and management challenges have been addressed in many literature studies [15]–[22]. They propose different optimization models and heuristic solutions for managing the VNFs placement problem. Besides, other researchers direct their efforts to realize the development of NFV management platforms [23]–[26]. However, the literature studies discard the fact that the VNFs are running as software applications on commodity servers that provide them not only with the flexibility and programmability of a distributed software application but with the benefits of the microservices architecture as well. Although the majority of the research projects have considered the carrier-grade nature of the NFV, their solutions do not reflect the carrier-grade requirements of cloud-based application, such as performance, fault resilience, high availability, scalability, QoS, VNF Components (VNFCs) structure, and governments' geo-restrictions [27]–[30]. VNFs are the building block of

NFV and are constructed by chaining various VNFCs to provide the desired services. The VNFCs take advantage of microservices architecture and the emerging implementation of Service-Oriented software Architecture (SOA). Each VNFC is foreseen as a microservice by itself, which enables heterogeneous VNF structures and allows more flexibility in terms of hosting environment and manageability. However, the intra-connections of VNFCs are directly affected by their placements, which affect and define the performance of a VNF service. Moreover, the interconnections of the VNFs that represent the logical container of the VNFC are directly affected by the VNFs' logical placements, which in return affect the service chain performance. With this in mind, VNFs' placement and service chaining are still important challenges that need further investigation to achieve the anticipated benefits of NFV, such as lower Operation and Capital Expenditure (OPEX and CPEX), on-demand scaling, and real-time network programmability while satisfying the above carrier-grade requirement. To address the inadequacies of VNFs placement and SFCs orchestration, this chapter introduces a novel VNF placement orchestration using a Mixed Integer Linear Programming (MILP) optimization model and associates it with a heuristic solution, Betweenness centrality Algorithm for Component Orchestration of NFV platform (BACON). The VNF placement orchestration is based on capturing all the carrier-grade requirements of an NFV application, such as the functionality, latency, and availability constraints. The main objective of the orchestration is finding the VNFs placements that satisfy the functional and non-functional constraints while minimizing the intra-communication delays between the VNF instances and enhancing the Quality of Service (QoS) of the computational path (SFC). The main contributions of this work can be summarized as follows: i) Propose an intelligent orchestrator that selects the best placement for the VNFs in a given NFV application to minimize the intra-communication delays between the VNF instances and enhance the QoS of the computational path (SFC). The optimized placement achieves a higher number of VNF instances participating in a service chain with different serving components. This outcome generates more active redundant computational paths that can be optimally used to achieve the desired QoS in terms of performance and high availability of service chains per request. ii) Capture the carrier-grade functionality constraints that affect the SFCs of the NFV application, such as the application's availability. iii) Capture the VNFs' dependencies constraints to generate a successful interacting SFCs. iv) Minimize the end-to-end delay of the SFC.

4.2 Background and Related Work

NFV is the technology that promises to revolutionize the telecommunication industry by providing substantial benefits to the next-generation networks. As NFV captures the interest of the leading telecommunication industrial equipment/service providers and academic researchers, intensive research projects are focusing on this technology.

Evolved Packet Core (EPC) is one of the basic network entities that are considered for virtualization. Taleb et al. [31] implement virtualized EPC (vEPC) using the cloud computing environment and demonstrate the feasibility of providing vEPC as a service. The authors also propose a comparative analysis of various architectures. Baba et al. [32] present and implement a vEPC architecture based on the VNFs. The architecture satisfies the requirements of the machine-to-machine service computing with reduced resources. The authors achieve 27% CPU time reduction with the proposed architecture. A smart VNF placement to deploy multi-tier cloud applications is proposed by PACE [33]. However, PACE overlooks many of the requirements that affect the VNF placement to achieve the desired

QoS in multi-tier cloud-based applications. These requirements include the VNF dependency hierarchy, delay tolerance, and anti/co-location constraints. An efficient and scalable VNF provisioning framework is proposed in E2 [34]. E2 is a framework that manages the VNFs by combining traffic engineering and the best VNF placement. It is suitable for a private cloud that serves a single type of applications and provides specific functionalities, such as traffic offloading to proprietary switches. E2 has discarded the various placement constraints, such as the instances' inter and intra-dependency and the delay tolerance between components. Bari et al. [35] propose an optimization algorithm for the VNF placement with a simplified set of constraints. The latter only considers the deployment cost, the resources requirement, and the processing delay. This optimization algorithm discards the placement constraints that satisfy the carrier-grade requirements of the VNF applications, such as the VNF chaining, reliability, and delay tolerance constraints. Mohammad Khan et al. [36] formulate a mixed-integer linear programming optimization model for VNFs placement and traffic flow routing while minimizing resource utilization. However, the proposed solution has focused on minimizing computational resources while ignoring non-functional constraints such as redundancy, dependency, and availability. Sahel et al. [37] focus on the network service chaining problem by formulating an integer linear programming model and a heuristic algorithm. The proposed solution is based on two segments: a decomposition selection with a backtracking phase and a mapping phase; leading consequently to suboptimal solutions. Nguyen et al. [38] formulate a quadratic programming model and propose a heuristic solution for the VNF placement and routing problems. However, the latter does not consider the VNF chaining and dependencies in their solution. The authors also consider that the networking service is provided by one VNF. Gadre et al. [39] propose an agile VNF placement solution based on a divide-and-conquer algorithm. The formulation considers that the VNFs are hosted on network switches. Hosting VNFs on virtual switches could accelerate the processing of the user's service chain request, but it contradicts the principles of SDN and NFV. Eramo et al. [20] propose an integer linear programming model for VNF migration and placement that minimizes the total expenses and revenue loss. The proposed work has overlooked various constraints in their considerations, such as the delay tolerance and dependencies between components. Ahvar et al. [40] formulate an integer linear programming for the VNF placement to minimize the cost of the NSP. However, the proposed ILP has considered the resource constraints, such as the decision variables without including other functional and non-functional constraints. Gupta et al. [41] introduce "COLAP", a predictive framework to place the participating VNFs of an SFC in a cloud environment while optimizing the service latency. In summary, this work has considered the service latency as the main metric while overlooking the VNF instances' dependencies and availability metrics. Zhang et al. [42] formulate the VNF placement problem as bin-packing and open Jackson network problems to achieve better resource utilization. The proposed solution has considered computational utilization as the main metric while ignoring non-functional constraints such as redundancy, dependency, and availability. Ayoubi et al. [21] propose a cut-and-solve approach for the VNF placement problem. The approach consists of two sub-problems and maximizes the policy-aware traffic flows count. This work has considered the service chain latency as the main metric while overlooking the dependencies and availability metrics. Qu et al. [43] formulate a MILP model and a heuristic approach to overcome the scalability of an optimization model while maximizing the reliability and minimizing the SFC end-to-end delays. The authors have proposed an algorithm that selects a subset of VNFs that are needed to generate an SFC and its redundant. The user traffic in the proposed algorithm is managed through the main SFC and its redundant, simultaneously, which results in a costly SFC deployment. The redundant path has a longer SFC leading to higher delay and thus affecting the QoS, in the case of SFC request's migration or failure. Despite the high demand for resource allocation for the proposed algorithm, the authors have discarded the delay tolerance between components. Hantouti et al. [44] have discussed SDN architectures for SFC and provided an analysis of the traffic steering techniques in the context of SDN-based SFC approaches. The work has presented a comprehensive analysis while identifying relevant research challenges and classifying the traffic steering techniques according to their efficiency in real-life networks. Bagaa et al. [45] have proposed an algorithm to define the optimal number of core network virtual elements to meet the demand of the mobile traffic while maintaining the QoS and maximizing the profits of the cloud operators. Furthermore, the authors have developed an algorithm to place the core network virtual instance in a federated cloud. Benkacem et al. [46] have formulated a VNF placement algorithm to minimize the cost while maximizing the Quality of Experience (QoE) of the virtual streaming service. The authors have applied the bargaining game theory to achieve an optimal tradeoff between the cost efficiency and QoE in the proposed solution. Laghrissi et al. [47] have addressed the problem of non-uniform distribution of signaling messages in irregular network topologies. They have proposed a solution to map the non-uniform distribution of signaling messages in the physical domain into a new uniform environment through the utilization of Schwartz-Christoffel conformal mappings. Taleb et al. [48] have proposed a VNF placement algorithm to cope with the surging mobile traffic while minimizing the cost in terms of the total number of instantiated VNFs to build a Virtual Network Infrastructure (VNI) in a cloud environment. The proposed algorithm objective functions are minimizing the path between users and their respective data anchor gateways and optimizing their sessions' mobility. Bagga et al. [11] proposed a placement algorithm for the mobile network functions over a federated cloud. The proposed algorithm instantiate the Packet Data Network Gateways (PDN-GW) virtual instances and select the adequate virtual PDN-GWs for user equipment receiving specific application service. Laghrissi et al. [49] developed a tool that facilitates the development of spatio-temporal models of mobile service usage over a particular geographical area. Furthermore, the tool help in defining mobile users' behavior in terms of mobility patterns and service consumption. Most of the aforementioned approaches propose solutions through private cloud interfaces, which are completely owned and controlled by cloud service providers. Also, the previous literature studies discard the fact that different applications can be hosted within the VNF entities in the cloud platform. So far, the proposed solutions for the NFV-SDN framework are mostly applicable to small-scale networks within a private cloud. Private clouds are groups of data centers owned by network service providers. The latter has full control over the entire infrastructure (physical servers, underlying core networks, virtual environments, and orchestrators). These solutions overlook multi-tenant support and co-existence with variant applications that are already using the cloud. Additionally, most of the above literature studies have focused on the VNF functionalities and placements from the perspective of single-tier applications (services) where a single type of VNFs is responsible for serving the users' requests (traffic). However, most NFV applications (services) are multi-tier applications (services) where a set of different types of VNFs work collaboratively to serve users' requests (traffic). The majority of the stated research has discarded various carrier-grade requirements, such as performance, fault resilience, high availability, scalability, QoS, and governments' geo-restrictions. In order to achieve the desired objectives of NFV, further studies should be conducted on the VNF's functionalities and placements from the perspective of multi-tier applications orchestration while satisfying the carrier-grade requirements. To mitigate the above inadequacies and pave the way for advancing NFV, SFC realization, and wider adoption within NSPs, this chapter proposes an intelligent VNF placement orchestrator. The latter proposes a MILP model and a heuristic solution, BACON, and satisfies various carrier-grade requirements of NFV platforms. The MILP model acts as a solver for small-scale NFV platforms and a benchmark for BACON that addresses large-scale NFV platforms.

4.3 Motivation

VNFs are hosted in a cloud environment where they are executed either within Virtual Machines (VMs) or within containers. The allocation of the VNFs' execution environment on the hosting servers in data centers directly affects the quality of service provided by these VNFs [50]–[53]. Therefore, having an optimal allocation for the VNFs is essential to satisfy the carrier-grade requirements.

4.3.1 VNF Placement Requirements

The ETSI defined framework does not provide a definition for the VNFs' placement management entity. Mainly, the mapping of the VNFs to their hosts is managed by the cloud service provider or is delegated to the users (VNFs' owners). Furthermore, NFV is associated with service function chains that are directly affected by the VNF placement. At the

Infrastructure as a Service (IaaS) level, the cloud service provider may offer a certain level of guaranteed resources performance and availability of the VMs assigned to the tenants. However, this approach does not guarantee the QoS of the VNFs deployed on these VMs. In fact, tenants would have to deploy and manage their VNFs in an efficient manner to achieve the desired quality of service. Netflix utilization of the Amazon Web Services (AWS) is an example of how tenants deploy and manage their cloud applications to meet the QoS requirements [54]. Netflix has contributed to various open-source software entities that integrate with AWS and other cloud services to enhance and achieve the desired quality of service. VNF schedulers that are agnostic of the intricacies of the tenant's application may result in inefficient placements.

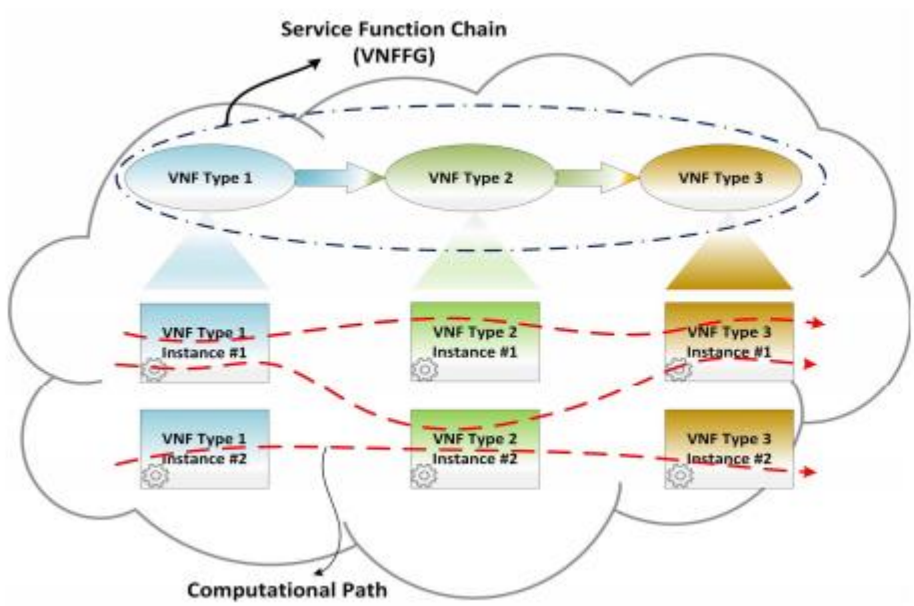

Figure 4.1: Service function chain and computational path of NFV of different

In these placements, computationally chained VNF components may be placed where the delay constraints can be violated, which hinders the application's functionality in terms of scalability and traffic offloading. A carrier-grade-aware (NFV-aware) application architecture that defines the computational paths, the participating components (VNFs) and the prospected service function chains are needed to enhance the scalability and traffic offloading of the application components (VNFs) [55]. It is necessary to note that the prospected service chain represents the path that should be generated to process the users' requests. The main objective of designing a carrier-grade application-aware (NFV-aware) architecture is to ensure that the system and its services are capable of serving various workloads with insignificant or zero degradation in QoS while maintaining the carrier-grade requirements with minimal SFC delay.

4.4 Problem Formulation

In order to take advantage of NFV technology, it is necessary to understand the architecture of its VNFs, their corresponding SFCs, and QoS requirements. This section describes the VNFs architecture and proposes the constraints to satisfy the requirements of QoS and meet the SLA.

4.4.1 VNF Architecture

NFV services (applications) are typically developed using a VNF-based architecture where each service consists of one or more VNFs. These VNFs are chained logically to create the service chain as described in the VNFFG. The VNFs' functionalities are combined to provide high-level abstracted services. As described by the VNFFG, the participating VNFs in the service function chain are configured to represent the functional dependencies and form the service computational paths. Fig. 4.1 illustrates the VNFs' service function chain. The dependency relation is captured at the service representation level where the delay tolerance and communication bandwidth attributes are defined. The delay tolerance determines the maximum latency at which a VNF instance can maintain communication with its dependent ones without declaring any service or computational path outage or degradation.

4.4.2 Requirements of VNFs Scheduler

Each VNF instance of the service is scheduled on a server in the cloud using VMs mappings. Each VM can be hosted on one server and can have at least one VNF instance running on it. Sudden demand spark or failure events can occur in the cloud, such as natural disasters, run-time failures, and global broadcasting events. In order to deal with these events, users' requests/traffic is balanced between various computational paths, or soft failovers to the redundant computational paths groups are triggered. Therefore, increasing the number of computational paths is translated into a better quality of service. The number of computational paths can be increased by adding VNFs on various tiers of the NFV service. However, adding more VNF components can overwhelm the OPEX and CAPEX of the users' investment. Besides, increasing the number of VNF components while overlooking their optimal placements can result in underutilized VNFs. To address these challenges, this chapter proposes a novel NFV-aware scheduling technique to achieve the carrier-grade QoS of an NFV service. The scheduler finds the optimal physical server to host the VNF component while minimizing the delay between the VNFs' components of the service function chain. This technique allows the maximum number of the VNFs to communicate without violating the functional and non-functional constraints. In other words, this technique generates the maximum number of computational paths to serve the users' requests while satisfying the quality requirements.

To consider a successful generation of computational paths, VNFs should be hosted on servers that can satisfy their computing requirements (CPU, memory, storage and networking resources) in the service chain without violating the delay tolerance among their dependent ones. In order to achieve the optimal count of the computational paths, this chapter proposes a mixed-integer linear programming model to schedule the VNFs while minimizing the traffic delays between the VNFs constituting the service chain. The MILP model provides an NFV-aware placement solution that generates mappings between the cloud physical servers and the VMs on which the tenants' VNFs are hosted while satisfying the following constraints:

(a) Capacity constraints: These constraints generate a servers' list that satisfies the resource demands of each VNF to meet the Service Level Agreement (SLA). In the proposed scheduler, the computational resources consist of CPU and memory.

(b) Network-Delay constraints: These constraints prune the above list to generate other servers' sub-list that satisfy the latency requirements to avoid any service degradation between the communicating VNFs.

(c) Availability constraints: These constraints prune the candidate servers generated by the capacity and delay requirements according to the following constraints:

i) Co-location constraint: It requires that the dependent VNFs should be placed on the same server of their sponsor if the delay tolerance of these dependent VNFs is ephemeral.

ii) Anti-location constraint: It requires that the dependent VNFs should be placed on different servers if their delay tolerances can compensate for the communication cost.

iii) Redundancy constraint: With this constraint, VNFs of the same type cannot reside on the same server. In this case, these VNFs should be placed as far as the delay tolerance allows.

(d) Dependency constraints: These constraints define the structure of the computational path between the defined VNFs.

4.5 Mathematical Formulation

In the MILP model, the set of VNFs participating in the SFC is denoted as V. V^A denotes a subset of V where its VNFs should satisfy the anti-location constraint. V^C denotes a subset of V where its VNFs should satisfy the co-location constraint. For each VNF, a subset of V is defined as dependent VNFs and denoted as V^D. v and v' represent a single VNF instance that belongs to a given VNF set. V_V^D is defined as the set of dependent VNFs of VNF v. The available set of servers in a given DC is denoted as S while the total number of servers in this set is denoted as N^S. s and s' represent a single server that belongs to a given server set. R denotes the set of computational resources types (CPU and memory). r represents a resource type in the computational resources set (CPU or memory). The computational resources r of a specific VNF v are denoted as V_{vr}^{Res} . The available resources r of a server s are denoted by S_{sr}^{Res} . The communication delay tolerance between the VNF components v and v' is defined as $T_{vv'}$. The communication delay between servers s and s' is denoted by $D_{ss'}$. The delay between two dependent VNFs v and v' is defined as D_{vv}. P_{vs} is the binary decision variable that defines the placement state of a VNF v on server s as follows:

$$P_{vs} = \begin{cases} 1 \ if \ V\ N\ F \ instance \ v \ is \ placed \ on \ server \ s \\ 0 \ otherwise \end{cases} \quad (1)$$

4.5.1 Model Formulation

The objective function and the constraints of the proposed MILP model are formulated as follows: Objective function:

$$Minimize \quad \sum_{v'}^{V^D} D_{vv'} \quad \forall\, v \in V \tag{2}$$

Subject to:

Availability/Dependency Constraints:

$$0 \le P_{vs} \le 1 \quad \forall\, v \,\epsilon V, \quad \forall s \,\epsilon\, S \tag{3}$$

$$\sum_{s=0}^{N^S} P_{vs} = 1 \quad \forall\, v \,\epsilon V \tag{4}$$

$$P_{vs} + P_{v's} \le 1 \quad \forall\, v\,, v' \,\epsilon\, V^A, \quad \forall\, s \,\epsilon\, S \tag{5}$$

$$P_{vs} + P_{v's} \ge 2 \quad \forall\, v\,, v' \,\epsilon\, V^c, \quad \forall\, s \,\epsilon\, S \tag{6}$$

Capacity Constraints:

$$\sum_{v=0}^{V} P_{vs} \times V_{vr}^{Res} \le S_{sr}^{Res} \quad \forall r \,\epsilon R, \quad \forall s \epsilon S \tag{7}$$

Network Delay Constraints:

$$D_{ss'} \times \left(P_{vs} + P_{v's'} - 1\right) - D_{vv'} \le 0 \;\; \forall\, v \in V, v' \in V_v^D, \quad \forall s, s' \in S \tag{8}$$

$$D_{vv'} \le T_{vv'} \quad \forall v \in V, v' \in V_v^D \tag{9}$$

As shown above, the NFV-aware placement constraints are grouped into availability, dependency, capacity, and network connection constraints. Constraint (3) defines the decision variable of the VNFs placement as a binary variable. Constraint (4) ensures that the defined VNF instance can only reside on one server at most. The anti-location constraint is defined in (5) and the co-location constraint is defined in (6). The capacity constraint (7) ensures that the candidate servers should have enough resources to host the assigned VNFs. Constraint (8) is defined as a network connection constraint. The latter ensures that a counted connection is established after the successful placement of the connected VNFs. Constraint (9) reflects the delay tolerance between the VNF types and maps the delay of the hosting servers to their VNFs instances.

4.5.2 Model Complexity

In order to determine the complexity of the proposed MILP model, we use the reduction method. In this section, we reduce the problem to a bipartite matching one in order to build our model accordingly [56]. Any scheduling problems can be interpreted as a triplet $a \mid b \mid c$, where a represents the problem environment, b represents the problem constraints, and c represents the objective function of the problem [57]. These triplet fields vary depending on the scheduling problem nature. Since the proposed placement approach addresses the allocation problem of VNF components set *(V)* on the available servers *(S)* with an objective function to minimize the communication delay between the dependent components, it can be formulated as a special case of the transportation problem. The formulation for the problem can be represented as $S_s \mid V_v \mid \sum D(x)$ where the S_s is the problem environment consisting of s different parallel servers, V_v defines the VNF job v that can proceed on a single server s, and $D(x)$ represents the objective function to be optimized. In this special case, the problem is known as a constrained bipartite matching problem. $G = (V, S, a)$ represents the bipartite graph that consists of VNF components nodes as set V, server nodes as set S, and arc a connecting the two sets. The arc $a = \{v, s\}$ assigns the VNF component *v* of set *V* to server *s* of set *S*, and it represents the decision variable P_{vs} defined in the previous section. Said that and using the Hopcroft-Karp algorithm, the bipartite maximal matchings are determined in polynomial time to the number of edges and vertices [58]. Thus, this type of bipartite matching problem that is formulated using linear programming models is categorized as an NP-hard problem, and by reduction, the proposed MILP model is NP-hard. Therefore, the proposed MILP model would be solvable for small-scale DC networks [59]. With this in mind, this chapter proposes a heuristic approach, BACON, to address the large-scale DC networks.

4.6 BACON: NFV-Aware Placement Algorithm

Due to the computational complexity of the proposed MILP model (NP-hard) and given the available computing processing power, the optimization model imposes a limitation on scaling to large-scale data center networks. Therefore, this section proposes a novel heuristic solution, Betweenness centrality Algorithm for Component Orchestration of NFV platform (BACON). BACON is based on the betweenness centrality of a node in a graph that works around the complexity and the time-consuming execution of the MILP model. Given a set of servers S and a set of VNFs participating in an SFC, BACON finds a feasible near-optimal VNF placement solution compared to the MILP optimal solution. The generated solution satisfies the previous constraints while relaxing the objective function. BACON executes different subroutines to find the placement solutions. Prior to the placement subroutine, BACON analyzes the types of the participating VNF in a given SFC. The VNF types are then divided into sub-groups according to their inherited dependency from the VNF Forwarding Graph (VNFFG). Each sub-group consists of three VNF types and is assigned a criticality attribute based on the communication delay tolerance of the participating VNF types. If BACON finds an undercount group, it shares VNF types from another subgroup. It is necessary to note that a group is considered as an undercount one when it contains less than three VNF types. After the grouping step, BACON builds a graph to represent the model system. The graph is built while considering that all the available servers in the data center are connected through a logical communication link in a mesh topology. BACON constructs the weighted graph *G(V,E,w)* where the vertices V represents the set of available servers in a given data center, the edges *E(v, v)* represents the logical communication link between the servers, and the weights *w(v, v)* represents the data communication delay between the servers. Since the SFC is divided into sub-groups of three components, the count of the vertices is triple the number of the servers. Thus, BACON covers all the placement possibilities of a sub-group. Once the graph is built, BACON calculates the Betweenness Centrality *(BC)* of the vertices (servers).

4.6.1 Calculation of Betweenness Centrality

The calculation of the betweenness centrality is based on the number of the shortest paths from the source node (s) to sink node (t) that passes through a specific node. Betweenness centrality:

$$B(v) = \sum \frac{\alpha_{st}(v)}{\alpha_{st}} \quad \forall v \neq s, v \neq t$$

$$where \begin{cases} \alpha_{st}(v) = Number\ of\ shortest\ paths\ from\ s\ to\ t\ passing\ through\ v \\ \alpha_{st} = Total\ number\ of\ shortest\ paths\ from\ s\ to\ t \end{cases} \tag{10}$$

Calculating the *BC* identifies the servers that can be anchors for the median nodes in the defined subgroups. Median nodes are the VNF instances of the mediator VNF type in a given sub-group. For example, in Fig. 4.1, the mediator VNF type in the given sub-group is VNF type 2. The placement of the median nodes of the sub-group is based on the critically attribute. BACON starts by placing the most critical VNF components of the sub-groups' median VNF types on the servers with the highest *BC* while satisfying the functional constraints. This placement criterion guarantees that the highest critical VNF components in a sub-group are placed in the most branched servers with minimal communication delays. It also guarantees that the critical component has the maximum count of the computational paths between the sub-group members without violating the communication delay tolerance. Once the median VNF components of the sub-group are placed on the servers, BACON hosts the members of the other sub-group on the servers. The group members that interconnect the sub-groups are placed on the servers with the highest *BC*. These servers belong to the intersection subset of the candidate servers of the interconnected sub-groups median as follows:

$$S_m = S_{SG'} \cap S_{SG''}$$

$$Where \begin{cases} S_m = A\ set\ of\ candidate\ servers\ to\ place\ the\ sub-group\ members \\ S_{SG'} = A\ set\ of\ candidate\ servers\ to\ place\ the\ sub-group\ SG'median\ members \\ S_{SG'} = A\ set\ of\ candidate\ servers\ to\ place\ the\ sub-group\ SG''median\ members \end{cases} \quad (11)$$

Algorithm 1 BACON

INPUT: $V = (V_1, V_2, ..., V_v)$

$V^D = (V_1, V_2, ..., V_n)$

$S = (S_1, S_2, ..., S_k)$

OUTPUT: S^P

where $S^P \subset S$

```
begin:
C = VNFComponentCriticalityRank(V)
SubGroups = DivideIntoSubGroups(V, V^D, 3)
SubGroups.AssociateCriticality(C)
G = BuildGraph(S, S.Delays)
for s_i ∈ S do
    s_i.Centrality = G.BetweennessCentrality(s_i)
end for
S.DescendingSort(Centrality)
SubGroups.DescendingSort(Criticality)
for g_i ∈ SubGroups do
    v = g_i.GetMedainVNF
    VNFType = v.GetType
    for v_i : VNFType.InstancesCount do
        for s_i ∈ S do
            if s_i.AvailableResources >= v_i.Resources then
                s_i.Host(v_i)
                S^P.Add(s_i)
                Break
            end if
        end for
    end for
end for
for g_i ∈ SubGroups do
    S' = g_i.CandidateServers ∩ g_{i+1}.CandidateServers
    M = g_i.GetVNFMembers
    for m_i ∈ M do
        VNFType = m_i.GetType
        for v_i : VNFType.InstancesCount do
            for s_i ∈ S' do
                if s_i.AvailableResources >= v_i.Resources then
                    s_i.Host(v_i)
                    S^P.Add(s_i)
                    Break
                end if
            end for
        end for
    end for
end for
return : S^P
end
```

Figure 4.2: BACON: The proposed heuristic algorithm.

BACON ensures that the group members have the best-fit servers with the most branching communication paths without violating the communication delay tolerance not only between the members of a sub-group but also between the interconnected members of the other sub-groups. Finally, BACON returns the VNF components set where each component is associated with a host. The generated placement is considered the best effort to achieve the minimum delay between the VNF components while maximizing the count of the possible computation paths. BACON is represented in Fig. 4.2. The highest order of magnitude in BACON is the subroutine that calculates the betweenness centrality of the vertices nodes. Examining the subroutine closely, the worst-case scenario can be calculated by finding all the combinations of the sub-groups while holding the median node then calculating the betweenness centrality of the median nodes. The results in order of magnitude are as follows:

$$O(\frac{n!}{(n-2)!\times 2!}) = O(\frac{n^2 - n}{2}) \tag{12}$$

Iterating n times over the median node, the worst case is then:

$$O(\frac{n^2 - n}{2}) \times n = O(\frac{n^3 - n^2}{2}) \tag{13}$$

Given n as the total number of available servers *"S"* in a given data center then:

$$O(\frac{S^3 - S^2}{2}) \tag{14}$$

4.7 NFV-Aware Placement Simulation

At the root level, the cloud consists of data centers distributed across various geographical areas. Each data center consists of multiple racks communicating through aggregated switches. Each rack has a set of shelves hosting servers, which can have different resources capacities. Servers residing on the same rack are connected with each other through the same network device, such as the Top Of the Rack (TOR) switch. Finally, the VMs/containers are hosted on the servers. This tree structure determines the network delay constraints and consequently, the delay between the communicating VNFs. This architecture divides the cloud into different latency zones. For the simulation, we have considered a 3-tier data center with:

- Access Switches or TOR Switches: Connecting the servers in the same rack.
- Aggregation Switches (ASw): Connecting the TOR switches.
- Core Switches: Connecting the ASw and acting as gateways to the external networks.

In order to generate the delay data-set of the servers in the simulation, we distribute the servers among the DC's racks and their data flow throughout the 3-tier DC network. Each DC network tier represents a specific delay with each unique server-to-server connection. The delays are generated randomly and follow a normal distribution with a specific predefined 99th percentile latency for each tier [60-62].

4.7.1 A. Simulation Results and Evaluation

The proposed MILP model and BACON are compared to two greedy algorithms. The first greedy algorithm is an NFV-agnostic algorithm. The other one is an NFV-aware algorithm, the "Greedy-k-NFV algorithm" which is proposed by Qu et al. [43]. This comparison shows the impact of NFV-aware placement on the computational paths' delays that affect the validity of these paths. It also evaluates the performance of BACON. During the simulation, we have used the vEPC as the simulation use case [12]. The 3rd Generation Partnership Project (3GPP) group introduces the EPC as all Internet-Protocol (IP) core network architecture [63]. It is designed to unleash the full potentials of mobile networks to provide broadband services. In the simulation, the four major components of the EPC have been considered; Mobile Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (SGW), and Packet Data Network Gateway (PGW or PDN-GW). Each component represents a VNF type in the input data-sets of the simulation. The simulation testbed is implemented and deployed on the SharcNet computing platform [64]. Wobbie-142 computing server is used to execute the simulation. Wobbie-142 computing server has 24 core-48 thread Intel Xeon E5-2690 v3 (2x sockets configuration) and 768.0 GB of memory. The simulation is executed in two phases:

- Phase 1: Small-scale DC network simulation In this phase, the data-set of a small-scale DC network is the input of the MILP model, BACON, and the greedy algorithms in the testbed. The input data is shown in Table 4.1, and the evaluation results are shown in Fig. 4.3, Fig. 4.4, Fig. 4.5, and Fig. 4.6.
- Phase 2: Large-scale DC network simulation In this phase, the data-set of a large-scale DC network is the input of BACON and the greedy algorithms in the testbed. The input data is shown in Table 4.2, and the evaluation results are shown in Fig. 4.7.

Table 4.1: Small-Scale DC Network Dataset

Set	Count
Available servers in DC	30
VNF of type MME	2
VNF of type HSS	3
VNF of type SGW	2
VNF of type PGW	3

Table 4.2: Large-Scale DC Network Dataset

Set	Count
Available servers in DC	300
VNF of type MME	20
VNF of type HSS	23
VNF of type SGW	25
VNF of type PGW	30

4.7.2 Component Intra-Communication Delay Comparative Analysis

This section provides a comparative analysis between the proposed NFV-aware MILP model, BACON, and the other greedy placement algorithms for small- and large-scale DC networks.

a) Small-Scale Network Simulation:

Fig. 4.3 shows the connection delays between the VNF instances of types MME and HSS. Fig. 4.4 shows the connection delays between the VNF instances of types MME and SGW. Fig. 4.5 shows the connection delays between the VNF instances of types SGW and PGW. As shown in the figures, the MILP model generates connections with the optimal minimum delay of the intra-connectivity between the entities. BACON achieves a near-optimal minimum delay where it deviates slightly from the MILP results. However, BACON has the lowest delays when compared to the other two greedy algorithms especially the "greedy-k-NFV" algorithm [43], which minimizes the communication delay of the SFC entities. The proposed MILP model and BACON do not only minimize the communication delay of the intra-links, but they also provide the best count of the links that satisfy the delay tolerance constraints between the VNFs instances. However, the other greedy algorithms generate placement decisions that violate the delay tolerance constraints between the VNFs instances. Any violation of the delay tolerance constraints terminates the connection between the VNF instances, and the link is considered as an invalid one for a computational path. The computational paths delays are shown in Fig. 4.6. The benefits of increasing the number of computational paths can be quantified by assessing how many members are participating in a functional group of a VNF instance. All group members should share the same VNF type and reside in the same orbital area. The orbital area is defined by the area where the functional group members can maneuver without violating any of the previous constraints. Fig. 4.8 shows the VNF orbital area. The boundaries of an orbital area are defined by the delay tolerance constraints of the dependent VNF instances. The higher the number of participating members in the functional group, the better its performance and reliability. The SFC performance and availability can be enhanced by the functional group members. From a performance perspective, user data traffic can be offloaded between the functional group members. The traffic offloading process is mainly managed by the health check entities in a system. The health check entities constantly monitor and collect various metrics from the active VNFs instances and balance the traffic to achieve the desired performance. From an availability perspective of the SFC, the functional group members are considered as redundant components that can mitigate the failure of the VNF instances due to a sudden interruption that affects the QoS of the SFC.

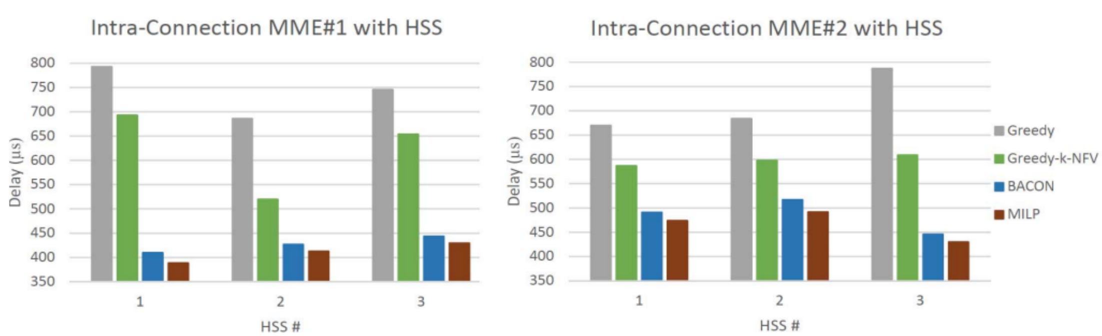

Figure 4.3: Intra-connection delay between VNF instances of types MME and HSS.

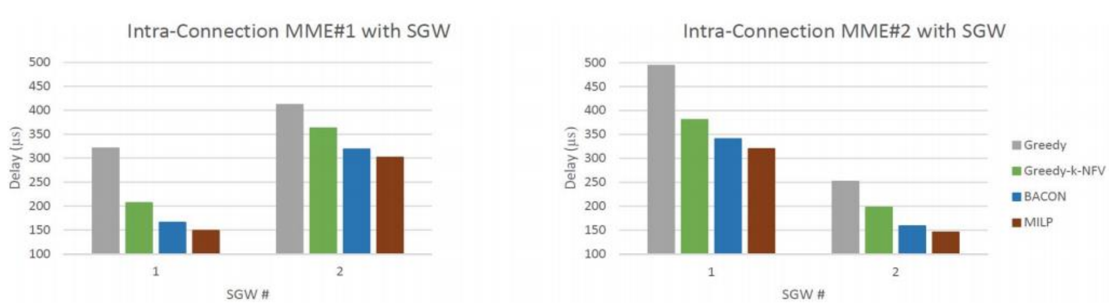

Figure 4.4: Intra-connection delay between VNF instances of types MME and SGW.

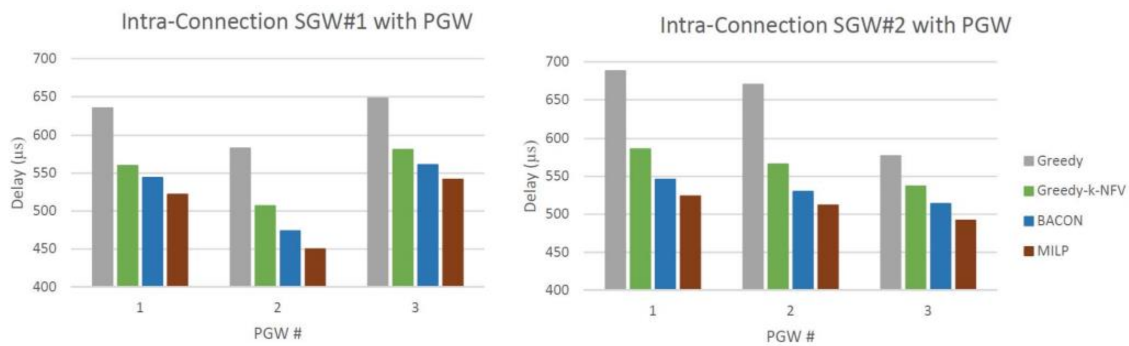

Figure 4.5: Intra-connection delay between VNF instances of types SGW and PGW.

The proposed MILP model and BACON generate the best count of functional group members. The results of Table 4.3 represent the VNF instances count in each VNF-type functional group for the small-scale DC simulation. The results show that the proposed heuristic "BACON" and the MILP model have achieved the best count of VNF members in each VNF-type functional group. BACON and the MILP model have achieved a count of two, three, two, and three group members for the following VNF-types; MME-VNF-type, HSS-VNF-type, SGW-VNF-type, and PGW-VNF-type, respectively. On contrary, the Greedy-k-NFV algorithm has achieved one, two, one, and one and the Greedy algorithm has achieved one, one, one, and one for these VNF-types; MME-VNF-type, HSSVNF-type, SGW-VNF-type, and PGW-VNF-type, respectively. Achieving higher member counts (higher VNF count of different types) in a specific functional group enhances the QoE for the service users. QoE is determined by the perception and evaluation of service from the user viewpoint. With the increase in the member counts in a functional group, the number of possible computational paths increments accordingly. These paths can be optimally used by services to facilitate the migration of data traffic between different computational paths in case of any degradation in performance or to migrate any errors while providing a seamless service to the user and maintaining the desired level of QoE.

b) Large-Scale Network Simulation

The MILP model has a high order of magnitude that hinders the results generation within a reasonable time given the available computing processing power. Therefore, it is not evaluated on the large-scale network simulation. BACON, the greedy NFV-agnostic, and the greedy-k-NFV algorithms are evaluated on the large-scale network. The simulation results are shown in Fig. 4.7, and the functional group counts are represented in Table 4.4. Similar to the small-scale network simulation, BACON achieves the lowest delays of the SFC computational paths and the highest count of the functional group members when compared to the other two greedy algorithms. The results in Table 4.4 show that the proposed heuristic “BACON” has achieved the best count of VNF members in each VNF-type functional group. BACON has achieved members’ count of 18, 22, 24, and 30 group members for the following VNF-types MME-VNF-type, HSS-VNF-type, SGW-VNF-type, and PGW-VNF-type, respectively. However, the Greedy-k-NFV algorithm has achieved 12, 13, 16, and 21 and the Greedy algorithm has achieved 3, 7, 4, and 9 for the following VNF-types MME-VNF-type, HSS-VNF-type, SGW-VNF-type, and PGW-VNF- type, respectively. BACON outperforms the other two greedy algorithms especially the greedy-k-NFV. The greedy-k-NFV is proposed to overcome the scalability of an optimization model while maximizing the reliability and minimizing the SFC end-to-end delays [43]. When compared to the greedy-k-NFV algorithm, BACON has a lower order of magnitude, which allows better scalability of the algorithm. The greedy-k-NFV has the following order of magnitude:

$$O(kN(M + N \log N))$$

$$Where \begin{cases} k = Initial\ set\ of\ paths \\ M = Number\ of\ edges \\ N = Number\ of\ nodes\ in\ network \end{cases} \qquad (15)$$

The simulation environment consists of DCs with multiple commodity servers to host the NFV applications. Given this simulation setup, the greedy-k-NFV algorithm variable can then be represented as follows:

- k = S, the number of servers in a given DC since the VNF instances can be hosted on any server in the DC.
- M = S2, since all servers are connected to each other with a logical mesh network.
- N = S, since the node in a network represents a server in the DC.

To this end, the order of magnitude of the greedy-k-NFV algorithm can be represented as:

$$O(S^4 + S^3 \log S)) \qquad (16)$$

where S = Number of servers in a given data center This shows that BACON has a lower order of magnitude:

$$O(\frac{S^3 - S^2}{2}) \qquad (17)$$

Thus, BACON outperforms the greedy-k-NFV algorithm as shown earlier.

Table 4.4 Small-scale DC network functional group members count

DC	Greedy	Greedy-k-NFV	BACON	MILP
MME functional group	1	1	2	2
HSS functional group	1	2	3	3
SGW functional group	1	1	2	2
PGW functional group	1	1	3	3

Table 4.3 Large-scale DC network functional group members count

DC	Greedy	Greedy-k-NFV	BACON
MME functional group	3	12	18
HSS functional group	7	13	22
SGW functional group	4	16	24
PGW functional group	9	21	30

4.7.3 SFC End-to-End Delay Comparative Analysis

The proposed MILP model and BACON do not only increase the count of the functional group members, but their placements' results show that the computational paths' delays are minimized when compared to the other two greedy algorithms. The computational paths' delays are shown in Fig. 4.6 and Fig. 4.7 for small- and large-scale networks respectively. Minimizing the computational paths' delays is a necessity for the SFC orchestration and management entities because the time difference between the delay tolerance and computing paths' delays allow the orchestration and the management entities to apply various policies on the systems. These policies vary according to the intent of the network service providers. For example, network service providers can introduce policies to achieve green or security analysis networks.

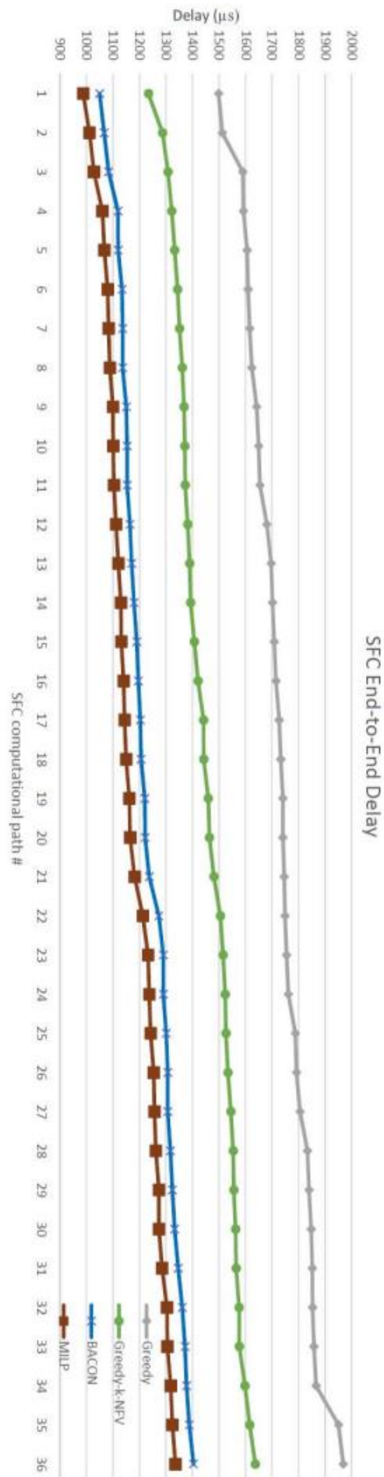

Figure 4.6: The end-to-end delays of SFCs in small-scale DC network.

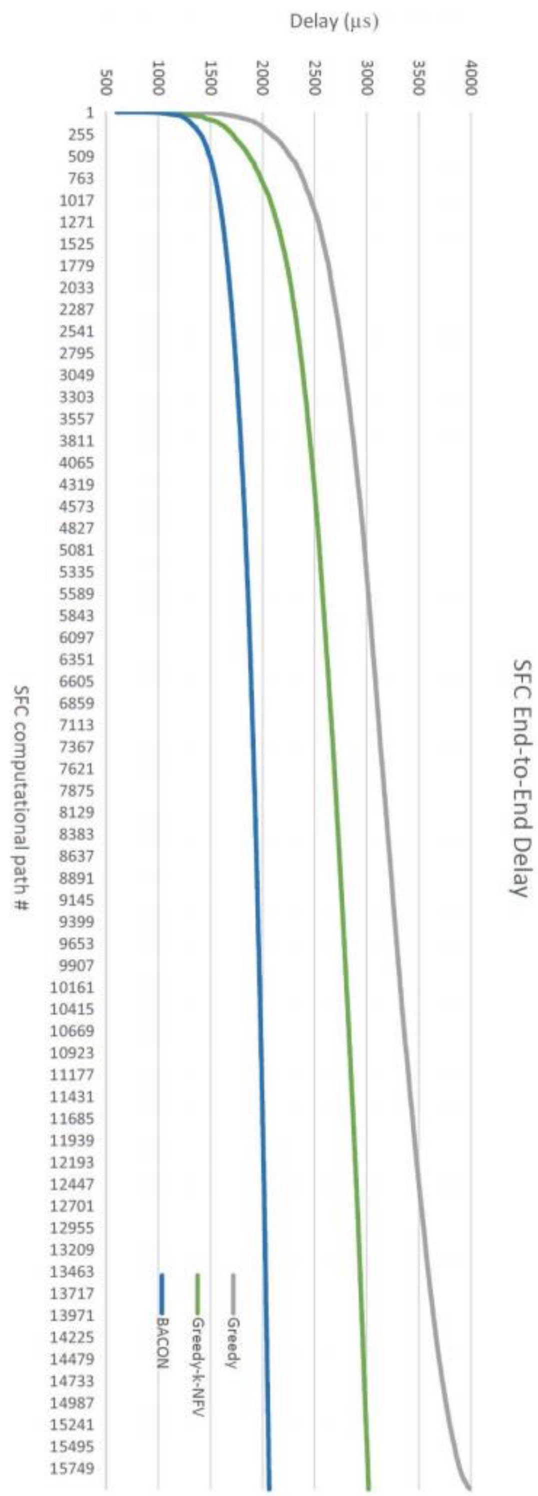

Figure 4.7: The end-to-end delays of SFCs in large-scale DC network.

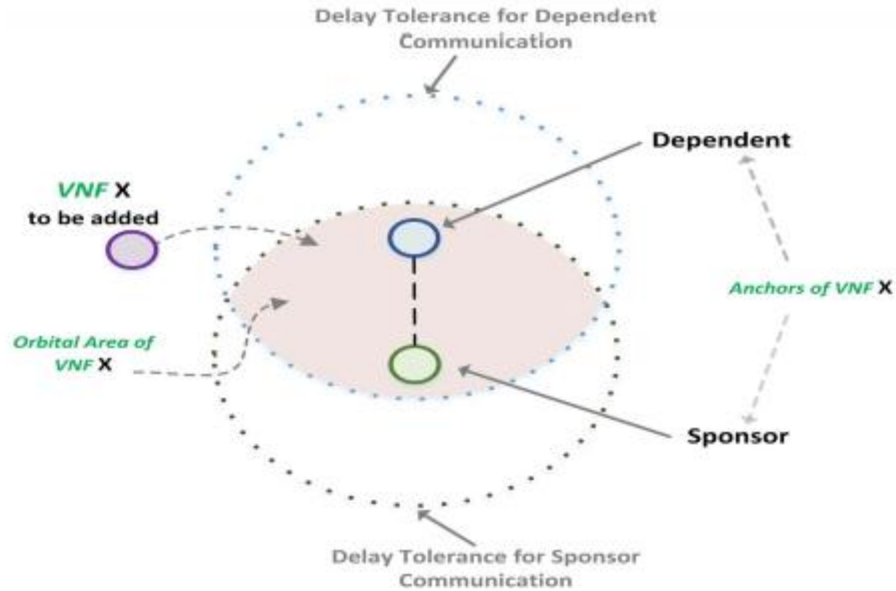

Figure 4.8: The placement zones of VNFs depending on their sponsors and

4.8 Conclusion

NFV has been introduced by the leading NSPs as a technology to revolutionize the information and communications technology industry. It has transformed the network functions from proprietary hardware to software-based applications where virtualization can be exploited. The academic and industrial researchers are investigating the possibilities of integrating NFV with the virtualization platforms. This step paves the way to unleash the full potentials of the NFV technology. Therefore, various NFV challenges should be resolved to achieve wider adoption of this technology. In this chapter, we presented a novel approach to address the placement problem of VNFs and their associated SFCs. A MILP model and a heuristic algorithm, BACON, were proposed to minimize the communication delay between the VNF instances and enhance the end-to-end QoS of the SFC. The proposed MILP model and BACON are implemented to capture the carrier-grade requirements of an NFV application. They are also evaluated on small- and large-scale DC networks data-set. In both cases, the proposed MILP model and BACON outperform the greedy NFV-agnostic and NFV-aware algorithms.

References

[1] Ericsson, "Ericsson mobility report on the pulse of the network society," Ericsson, Stockholm, Sweden, White Paper EAB-16:006659 Uen, Nov. 2016.

[2] Cisco, "Cisco visual networking index: Forecast and methodology, 2016–2021," Cisco, San Jose, CA, USA, White Paper C11-481360-01, Jun. 2017.

[3] IDC, "Is your network ready for digital transformation?" White Paper, IDC#US41912917, Indianapolis, IN, USA, Jan. 2017.

[4] M. A. Sharkh, M. Jammal, A. Shami, and A. Ouda, "Resource allocation in a network-based cloud computing environment: Design challenges," IEEE Commun. Mag., vol. 51, no. 11, pp. 46–52, Nov. 2013.

[5] ETSI, "Network function virtualization: An introduction, benefits, enablers, challenges, & Call for action," ETSI, Sophia Antipolis, France, White Paper 1, Oct. 2012.

[6] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," Comput. Netw. J., vol. 72, pp. 74–98, Oct. 2014.

[7] A. Ksentini, M. Bagaa, and T. Taleb, "On using SDN in 5G: The controller placement problem," in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2016, pp. 1–6.

[8] A. Ksentini, M. Bagaa, T. Taleb, and I. Balasingham, "On using bargaining game for optimal placement of SDN controllers," in Proc. IEEE Int. Conf. Commun. (ICC), May 2016, pp. 1–6.

[9] ETSI, “Network functions virtualisation (NFV); management and orchestration,” ETSI GS NFV-MAN, White Paper 001 V1.1.1 (2014-12), Dec. 2014.

[10] ETSI, “Network functions virtualisation (NFV); ecosystem; report on SDN usage in NFV architectural framework,” ETSI GS NFV-EVE, White Paper 005 V1.1.1 (2015-12), Dec. 2015.

[11] M. Bagaa, T. Taleb, and A. Ksentini, “Service-aware network function placement for efficient traffic handling in carrier cloud,” in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Apr. 2014, pp. 2402–2407.

[12] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, “NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC),” IEEE Netw., vol. 28, no. 6, pp. 18–26, Nov. 2014.

[13] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latré, M. Charalambides, and D. Lopez, “Management and orchestration challenges in network functions virtualization,” IEEE Commun. Mag., vol. 54, no. 1, pp. 98–105, Jan. 2016.

[14] H. Hawilo, L. Liao, A. Shami, and V. C. M. Leung, “NFV/SDN-based vEPC solution in hybrid clouds,” in Proc. IEEE Middle East North Africa Commun. Conf. (MENACOMM), Apr. 2018, pp. 1–6.

[15] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, “Service function chaining in next generation networks: State of the art and research challenges,” IEEE Commun. Mag., vol. 55, no. 2, pp. 216–223, Feb. 2017.

[16] S. Mehraghdam, M. Keller, and H. Karl, “Specifying and placing chains of virtual network functions,” in Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet), Oct. 2014, pp. 7–13.

[17] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, “Near optimal placement of virtual network functions,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), May 2015, pp. 1346–1354.

[18] R. Mijumbi et al., “Design and evaluation of algorithms for mapping and scheduling of virtual network functions,” in Proc. IEEE Conf. Netw. Softwarization, Apr. 2015, pp. 1–9.

[19] T. Taleb, A. Ksentini, M. Chen, and R. Jantti, “Coping with emerging mobile social media applications through dynamic service function chaining,” IEEE Trans. Wireless Commun., vol. 15, no. 4, pp. 2859–2871, Apr. 2016.

[20] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, “An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures,” IEEE/ACM Trans. Netw., vol. 25, no. 4, pp. 2008–2025, Aug. 2017.

[21] S. Ayoubi, S. Sebbah, and C. Assi, “A cut-and-solve based approach for the VNF assignment problem,” IEEE Trans. Cloud Comput., to be published.

[22] S. Dutta, T. Taleb, and A. Ksentini, “QoE-aware elasticity support in cloud-native 5G systems,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2016, pp. 1–6.

[23] J. Batalle, J. F. Riera, E. Escalona, and J. A. Garcia-Espin, “On the implementation of NFV over an OpenFlow infrastructure: Routing function virtualization,” in Proc. IEEE SDN Future Netw. Services (SDN4FNS), Nov. 2013, pp. 1–6.

[24] I. Cerrato, M. Annarumma, and F. Risso, “Supporting fine-grained network functions through Intel DPDK,” in Proc. 3rd Eur. Workshop Softw. Defined Netw. (EWSDN), Sep. 2014, pp. 1–6.

[25] J. Martins et al., “ClickOS and the art of network function virtualization,” in Proc. 11th USENIX Conf. Netw. Syst. Design Implement., Apr. 2014, pp. 459–473.

[26] T. Taleb, A. Ksentini, and R. Jantti, “‘Anything as a service’ for 5G mobile systems,” IEEE Netw. Mag., vol. 30, no. 6, pp. 84–91, Dec. 2016.

[27] M. Jammal, A. Kanso, and A. Shami, “High availability-aware optimization digest for applications deployment in cloud,” in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2015, pp. 6822–6828.

[28] M. Jammal, A. Kanso, P. Heidari, and A. Shami, “Availability analysis of cloud deployed applications,” in Proc. IEEE Int. Conf. Cloud Eng. (IC2E), Apr. 2016, pp. 234–235.

[29] M. Jammal, H. Hawilo, A. Kanso, and A. Shami, “Mitigating the risk of cloud services downtime using live migration and high availability-aware placement,” in Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom), Dec. 2016, pp. 578–583.

[30] M. Jammal, A. Kanso, P. Heidari, and A. Shami, “Evaluating high availability-aware deployments using stochastic Petri net model and cloud scoring selection tool,” IEEE Trans. Services Comput., to be published.

[31] T. Taleb et al., “EASE: EPC as a service to ease mobile core network deployment over cloud,” IEEE Netw. Mag., vol. 29, no. 2, pp. 78–88, Mar. 2015.

[32] H. Baba, M. Matsumoto, and K. Noritake, "Lightweight virtualized evolved packet core architecture for future mobile communication," in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Mar. 2015, pp. 1811–1816.

[33] L. E. Li et al., "PACE: Policy-aware application cloud embedding," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2013, pp. 638–646.

[34] P. Shoumik et al., "E2: A framework for NFV applications," in Proc. 25th Symp. Oper. Syst. Princ., Oct. 2015, pp. 121–136.

[35] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," IEEE Trans. Netw. Service Manage., vol. 13, no. 4, pp. 725–739, Dec. 2016.

[36] A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. K. Ramakrishnan, and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," in Proc. 21st IEEE Int. Workshop (LANMAN), Apr. 2015, pp. 1–6.

[37] S. Sahhaf et al., "Network service chaining with optimized network function embedding supporting service decompositions," Comput. Netw. J., vol. 93, pp. 492–505, Dec. 2015.

[38] T. M. Nguyen, S. Fdida, and T. M. Pham, "A comprehensive resource management and placement for network function virtualization," in Proc. IEEE Conf. Netw. Softwarization (NetSoft), Jul. 2017, pp. 1–9.

[39] A. Gadre, A. Anbiah, and K. M. Sivalingam, "A customizable agile approach to network function placement," in Proc. Eur. Conf. Netw. Commun. (EuCNC), Jun. 2017, pp. 1–6.

[40] S. Ahvar, H. P. Phyu, S. M. Buddhacharya, E. Ahvar, N. Crespi, and R. Glitho, "CCVP: Cost-efficient centrality-based VNF placement and chaining algorithm for network service provisioning," in Proc. IEEE Conf. Netw. Softwarization (NetSoft), Jul. 2017, pp. 1–9.

[41] L. Gupta, M. Samaka, R. Jain, A. Erbad, D. Bhamare, and C. Metz, "COLAP: A predictive framework for service function chain placement in a multi-cloud environment," in Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC), Jan. 2017, pp. 1–9.

[42] Q. Zhang, Y. Xiao, F. Liu, J. C. S. Lui, J. Guo, and T. Wang, "Joint optimization of chain placement and request scheduling for network function virtualization," in Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jun. 2017, pp. 731–741.

[43] L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks," IEEE Trans. Netw. Service Manag., vol. 14, no. 3, pp. 554–568, Sep. 2017.

[44] H. Hantouti, N. Benamar, T. Taleb, and A. Laghrissi, "Traffic steering for service function chaining," IEEE Commun. Surveys Tuts., to be published.

[45] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, "Coalitional game for the creation of efficient virtual core network slices in 5G mobile systems," IEEE J. Sel. Areas Commun., vol. 36, no. 3, pp. 469–484, Mar. 2018.

[46] I. Benkacem, T. Taleb, M. Bagaa, and H. Flinck, "Optimal VNFs placement in CDN slicing over multi-cloud environment," IEEE J. Sel. Areas Commun., vol. 36, no. 3, pp. 616–627, Mar. 2018.

[47] A. Laghrissi, T. Taleb, and M. Bagaa, “Conformal mapping for optimal network slice planning based on canonical domains,” IEEE J. Sel. Areas Commun., vol. 36, no. 3, pp. 519–528, Mar. 2018.

[48] T. Taleb, M. Bagaa, and A. Ksentini, “User mobility-aware virtual network function placement for virtual 5G network infrastructure,” in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2015, pp. 3879–3884.

[49] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, “Towards edge slicing: VNF placement algorithms for a dynamic & amp; realistic edge cloud environment,” in Proc. IEEE Global Commun. Conf., Dec. 2017, pp. 1–6.

[50] H. Hawilo, A. Kanso, and A. Shami, “Towards an elasticity framework for legacy highly available applications in the cloud,” in Proc. IEEE World Congr. Services, Jun./Jul. 2015, pp. 253–260.

[51] M. Jammal, A. Kanso, P. Heidari, and A. Shami, “A formal model for the availability analysis of cloud deployed multi-tiered applications,” in Proc. IEEE Int. Conf. Cloud Eng. Workshop (IC2EW), Apr. 2016, pp. 82–87.

[52] M. Jammal, H. Hawilo, A. Kanso, and A. Shami, “ACE: Availability-aware CloudSim extension,” IEEE Trans. Netw. Service Manage., vol. 15, no. 4, pp. 1586–1599, Dec. 2018.

[53] M. Jammal, H. Hawilo, A. Kanso, and A. Shami, “Generic input template for cloud simulators: A case study of CloudSim,” Softw., Pract. Exper., pp. 1–28, 2018.

[54] Netflix. (2016). Completing the Netflix Cloud Migration. Accessed: Sep. 22, 2017. [Online]. Available: https://media.netflix.com/en/ company-blog/completing-the-netflix-cloud-migration

[55] H. Hawilo, M. Jammal, and A. Shami, "Orchestrating network function virtualization platform: Migration or re-instantiation?," in Proc. IEEE 6th Int. Conf. Cloud Netw. (CloudNet), Sep. 2017, pp. 1–6.

[56] N. Mustafa. (2013). Reductions to Prove NP Hardness. [Online]. Available: https://perso.esiee.fr/~mustafan/TechnicalWritings/ Complexityslides/lec6.pdf

[57] M. L. Pinedo, "Deterministic models: Preliminaries," in Scheduling— Theory, Algorithms, and Systems. Boston, MA, USA: Springer, 2011.

[58] Cornell University. (2017). Matchings. [Online]. Available: http:// www.cs.cornell.edu/courses/cs6820/2016fa/handouts/matchings.pdf

[59] R. Coric, M. Dumic, and D. Jakobovic, "Complexity comparison of integer programming and genetic algorithms for resource constrained scheduling problems," in Proc. Int. Conv. Inf. Commun. Technol., Electron. Microelectron., 2017, pp. 1182–1188.

[60] C. Guo et al., "Pingmesh: A large-scale system for data center network latency measurement and analysis," in Proc. ACM Conf. Special Interest Group Data Commun. (SIGCOMM), Aug. 2015, pp. 139–152.

[61] Plexxi, "Latency in Ethernet switches," Plexxi, Nashua, NH, USA, White Paper, 2016. [Online]. Available: http://www.plexxi.com/wpcontent/uploads/2016/01/Latency-in-Ethernet-Switches.pdf

[62] VMware, "Network I/O Latency on VMware vSphere 5," Technical White Paper, Palo Alto, CA, USA, 2012. [Online]. Available: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/ techpaper/network-io-latency-perf-vsphere5-white-paper.pdf

[63] Digital Cellular Telecommunications System (Phase 2+); Universal Mobile Telecommunications System (UMTS);LTE; Network Architecture, document 3GPP TS 23.002 version 11.6.0, ETSI, 2013.

[64] SHARCNET. (Apr. 2017). Cluster Wobbie.Sharcnet.Ca. Accessed: Aug. 5, 2017. [Online]. Available: https://www.sharcnet.ca/my/ systems/show/29

Chapter 5

5 Orchestrating Network Function Virtualization Platform: Migration or Re-Instantiation?

5.1 Introduction

During the period spanning from 2012 – 2018, mobile data traffic has increased by more than 1800% [1], [2]. With the emergence of 4G and 5G technologies, Network Service Providers (NSPs) are facing high connectivity demands. The shift towards smart and inter-connected devices, such as smartphones, wearables, and machine to machine (M2M) connections is placing an increased burden on NSPs as well. Currently, the number of mobile-connected devices is greater than eight billion, which is to exceed eleven billion by 2021 [1], [3]. The introduction of smart devices has had a profound and indisputable effect on network traffic globally as more than 89% of the total traffic is currently attributed to these devices[1]. This percentage continues to increase each year with the continual introduction and adoption of the latest devices and technologies. In order to meet these unprecedented and growing demands, NSPs should enhance the portability, interoperability, performance, reliability, security, and management of their networks while reducing their capital expenditures (CAPEX) and operational expenditures (OPEX) [4]. One way to address these needs is through Network Function Virtualization (NFV). NFV is a technology proposed by the European Telecommunications Standards Institute (ETSI) in 2012 to solve the challenges mentioned above as well as those associated with service availability, scalability, and resilience of current networks [5]. NFV can be defined as the decoupling of network functions from their underlying proprietary hardware. When decoupled, the network functions (NFs) are virtualized and are hereafter referred to as Virtual Network Functions (VNFs). These VNFs are entirely software-based and are executed as applications on commercial, off-the-shelf equipment (COTS). Several benefits arise from the virtualization of NFs including network scalability and flexibility, improved operating performance, reduced development cycles and time to market, as well as CAPEX and OPEX savings [5]. NFV alongside cloud computing and software-defined networking (SDN) is revolutionizing current and conventional networks and providing a framework for the networks of the future. In order to ensure that the service provided by an NSP is reliable, they are held to certain carrier-grade requirements and quality of service (QoS) guarantees. There are several critical applications such as emergency services, medical services, and financial services, which require high availability (HA) in the realm of five nines (99.999%) or greater. Quantified, this minimum guarantee of five nines translates to less than five and a half minutes of down-time during a calendar year [6]. Furthermore, the HA must be combined with a low latency guarantee to truly and holistically provide a reliable and available service. As with any application dealing with hardware and software, VNFs can experience both scheduled (maintenance) and unscheduled (natural events, overload, equipment failure) outages. The challenge NSPs face is to provide a service that can overcome these planned and unplanned events and still abide by the HA and QoS guarantees. One way to do this is to introduce resiliency into the network; that is, the ability to recover a service after an outage in a timely manner while maintaining a seamless user's service. This network resiliency can be achieved by migration and re-instantiation [7]. However, the introduction of migration and re-instantiation must be accompanied by an intelligent orchestrator to select between the two in order to maintain service performance and availability. The need for an intelligent orchestrator is evident when considering the execution of network functions in a given network. The placement of VNFs plays a major role in terms of overall service quality as there have been observed cases where the underlying network infrastructure is underutilized however, delays are experienced due to high latency and low throughput [8]. Additionally, there are certain functions (ex. monolithic applications) in which the preservation of application states is essential to ensure service continuity; such functions would benefit much more from the migration technique as it preserves the state while transferring to the target server. Contrastingly, other functions (ex. stateless microservices) do not require the preservation of the application states to ensure service continuity; such functions would benefit from the re-instantiation technique since it does not preserve application states when it gets re-instantiated on the target server. Since the migration and re-instantiation techniques each have their respective costs, selecting the most appropriate technique for a given VNF is an essential component of service preservation and continuity. While the type of VNF is important in orchestrator's selection of migration or re-instantiation, it is not the only factor that must be considered. Additional delays are introduced when considering the rebuilding and governance registration of the recently migrated or re-instantiated VNF. Furthermore, NFV applications are presented as a group of interconnected VNFs known as Service Function Chains (SFCs). This means that it is not enough to only consider one specific VNF when selecting to migrate or re-instantiate, but rather the requirements of the entire SFC should be considered. To address the lack of carrier-grade level resilient elasticity in NFV enabled networks, this chapter introduces an intelligent NFV-aware orchestrator that manages the VNFs placements and executes migration or re-instantiation policy while minimizing the impact of the VNF's outage on the network. For this purpose, we propose a novel placement and policy-aware selection techniques that look into the service level priority or criticality, VNFs' interdependencies, their communication delay tolerance and computational resources requirements. When an outage occurs, the orchestrator translates the above requirements into constraints to execute the policy-aware selection technique accordingly. The latter executes a placement algorithm and decides on migration or re-instantiation while minimizing the SFC latency and total experienced downtime. To this end, a mixed-integer linear programming (MILP) model is developed as an optimal solution for VNFs placement and policy-aware selection in small-scale networks. For largescale systems, the MILP model is associated with a latency-aware heuristic solution that is based on the graph-related algorithms. The work of this chapter is an extension of another chapter [6]. Also, proposed a MILP model to decide on migration or re-instantiation, it discarded the complexity of the MILP model, the service levels of the NFV applications and the need for a heuristic solution that manages large-scale networks. Therefore, we extend this work with the following: Model the carrier-grade's functionality constraints that affect the SFCs of the NFV application, Model the VNFs' interdependency constraints to build successful SFCs, Model the criticality/priority of the different service levels of NFV applications, Develop an intelligent orchestrator that selects whether to migrate or re-instantiate for the VNFs in a given NFV application to minimize the latencies between the VNF instances and improve the QoS of the SFC.

5.2 Related Work

The importance of state migration or re-instantiation and its applications to the need for elasticity in NFV enabled networks has been a widely researched topic. Gember-Jacobson et al. identify operational challenges in several of the proposed network function state transfer frameworks such as safety, scalability, and efficiency and propose methods including packet reprocessing and Peer-to-Peer (P2P) transfers in an effort to reduce latency and state transfer time [9]. Rajagopalan et al. propose Split/Merge, a system that enables the dynamic scaling in and out along with distributed load elasticity [10]. Woo et al. propose a framework that meets performance thresholds and allows for the elastic scaling of VNFs [11]. These studies deal with the migration mechanism however, they do not consider the practical implementation of this mechanism in a virtualized network since constraints surrounding the migration such as availability and resources have been discarded. Xia et al. discuss the optimal VNF migration problem while considering the constraints of both computational and network resources [12]. They propose a heuristic solution, which achieves comparable results to the optimal model as proven simulation. However, this model overlooks additional constraints including availability and service discovery delay and. Xia et al. also discuss the migration of a Virtual Machine (VM) as opposed to an individual VNF to address the need for protocols required for internal state transfer [13]. The sole constraint on the formulated problem pertains to the link bandwidth and the objective is to minimize the duration of all migrations. This approach overlooks constraints on availability, computational resources, and delay tolerances between interconnected VFNs belonging to the same SFC. Cho et al. study the problem of VNF migration for low latency networks [14]. They have formulated a model with the objective of minimizing the number of migrations a given VNF undergoes as well as maximizing the reduction of network latency post-migration. The proposed model considers computational constraints (CPU capacity) and network resource constraints (link bandwidth); however, the model does not consider availability, service discovery, or SFC delay tolerance constraints. Furthermore, this model does not take into consideration the concept of re-instantiation as an NFV management technique. Minimization of the migration and consolidation energy is considered as an ILP formulated by Eramo et al. and multiple heuristic solutions to solve the NP-hard problem are suggested [15]. The results suggest that there can be significant energy savings when performing VNF migration. The authors have considered constraints on computational resources, network resources, and server utilization; however, availability and network delay constraints are not captured. Zhang et al. formulate the VNF migration and rule update problem with the joint objective of minimizing the cost of migrating a VNF as well as the delay associated with updating the network [16]. This model considers constraints on the link capacity and flow table size; however, computational, availability and delay constraints are not considered. Gumaste et al. aim to simplify VM migrations and simultaneously reduce their cost by using virtual programmable optics control functions in conjunction with virtual migration functions[17]. The objective of the proposed model is to maximize the network traffic provisioned by the datacenter. The proposed model considers network resource, overprovisioning, and robustness constraints; however, it doesn't address affinity constraints and network delay constraints between interconnected VNFs in a given SFC. Additionally, their work fails to consider re-instantiation as a post-failure recovery option. Carpio et al. propose a model that compares the tradeoff between VNF replication and VNF migration regarding the resource requirements, traffic management, and the QoS impacts [18]. The problem is formulated as a linear program with the weighted objective of minimizing server, link, and migration costs. The model took into consideration migratory and replicative constraints; however, it discarded constraints regarding availability or network delays. Sun et al. have identified three main challenges, buffer overflow avoidance, migration cost calculation, and migration flow selection in their design of a flow migration controller [19]. This work attempts to formulate an optimization problem taking the above challenges into consideration for three elasticity control scenarios (VNF scaling in, scaling out, and load balancing). The proposed model considered constraints to avoid buffer overflow and hotspot creation; however, it does not address delay, availability, or computational resource constraints. Lin et al. posit a programmable buffer (PB) that would manage the location, manner, and timing of the buffering of a given flow [20]. The authors assert that the network must actively buffer live traffic during the pre-migration process of transferring state information. Evaluation of PB has determined that it was capable of realizing near-optimal throughput speeds in excess of those suggested by the current 5G standards. Most of the above-related work addresses the VNF migration issue without taking into consideration the technique of VNF re-instantiation. VNF migration and re-instantiation should be considered simultaneously in an effort to provide a holistic carrier-grade model. Furthermore, many of the above studies constrain their models in terms of computational and network resources without taking into account availability, SFC delay tolerance, SDN convergence delay, and service discovery delay. To overcome these deficiencies, this chapter proposes a MILP optimization model and a heuristic solution, constrained to address the aforementioned constraint inadequacies, which highlights the impact of selecting between VNF migration and re-instantiation on VNF downtime and SFC delays.

5.3 Background and Motivation

NSPs worldwide are rapidly trying to adopt, implement, and improve the idea of an NFV-enabled network. In recent years, several members of the telecommunications and entertainment industries have been using the platforms made available through CSPs such as Software, Platform, and Infrastructure as a service (SaaS, PaaS, and IaaS) to develop and manage their own NFV applications. Perhaps the most notable example of this process is Netflix who through the use of Amazon Web Services serve their extensive user base and are responsible for 15% of the world's downstream internet traffic as of 2018 [21]. The following section discusses NFV-related concepts, system modeling requirements, migration and re-instantiation in the NFV environment.

5.3.1 ETSI MANO Framework

With the decoupling of software for the underlying hardware, additional steps must be taken to ensure proper management of the NFV enable network. ETSI has proposed a Management and Orchestration (MANO) framework to address the transition from conventional networks to NFV enabled ones. NFV MANO is comprised of three functional components, the virtualized infrastructure manager (VIM), the VNF manager (VNFM), and the VNF orchestrator (VNFO) [22]. The combination of these three entities under the MANO framework is responsible aspects of the network including but not limited to service management, carrier-grade requirements, performance, service availability, and VNF placement. This section offers further insight into the role of MANO in the aforementioned network aspects. An essential part of an NFV-enabled network is the management and orchestration of all the various entities and components present in the network. The following discusses the main components and building blocks of NFV orchestration and management and how their respective functionalities are exhibited and used in the overall network framework.

a) Virtualized Infrastructure Manager (VIM)The virtualized infrastructure manager (VIM) is responsible for the management of the operational records of both the physical and virtualized resources of the NFV infrastructure. In terms of virtualized resource records, the VIM manages records such as the reservation and allocation of virtual resources. In terms of physical resource records, the VIM possesses the mapping of virtualized to physical resources such as computational, storage and network capacity [23]. It is necessary to note that operations such as migration, instantiation, and scaling are functionalities of the VIM. Some functions available to the manager include VNF instantiation, feasibility checking, instance modification, instance scaling, and instance termination.

2) VNF Manager (VNFM): The VNF Manager (VNFM) is tasked with the management of VNF instance lifecycles. Under MANO, the VNFM is able to manage several heterogeneous VNF instances simultaneously and is capable of applying general functions to all of its managed VNF instances or, unique functions targeted towards specific types of VNF instances [23].

3) NFV Orchestrator (NFVO): NFVO is responsible for two main tasks, the fulfillment of resource orchestration functions through the management of resources across several VIMs and the fulfillment of network service orchestration functions through the management of network service lifecycles[23]. Furthermore, the orchestrator, through the use of the resource orchestration functionality is able to support the access of network resources and

manage the sharing of resources across VNF instances. The work presented in this chapter is an orchestrator-aware solution that collects information from the infrastructure manager and communicates it with the VNFM to decide on the instances.

5.3.2 System Modeling and Requirements

In order to accurately capture all aspects of the VNF platform in the cloud, the main components and inner workings must be defined. The following describes the various entities and requirements which critically affect the carrier-grade metrics associated with the network.

1) Service Management: In a cloud environment, the Cloud Service Provider (CSP) is responsible for providing Virtual Machines (VMs) and containers given the resource (computational, memory, etc.) requirements of the NFV whereas the NSP is responsible for the deployment and orchestration of a given VNF as well as its adherence to the standards of a carrier-grade service. Furthering the responsibilities described above, the CSP, during operation, provides a plethora of metrics relating to the service however, it is the NSP who is responsible for their interpretation [6].

2) Carrier-Grade Requirements for NFV Applications: Taking into consideration the aforementioned metrics relating to the service provision, the following describes how these collected metrics are transformed into QoS aware constraints. Through the implementation of these constraints, an NSP is able to ensure the service delivered to the end-user meets all the predefined performance guarantees.

a) Performance-Aware Constraints Through the intelligent management of VNF instances and entities, the designing of an NFV application which is performance-aware is realized. Furthermore, intelligent management can incorporate characteristic attributes of the cloud environment such as vertical and horizontal resource scaling whereby performance and availability are increased respectively. Vertical scaling is the process by which additional resources are assigned to a VM or container to improve its computational ability; these resources include a virtual central processing unit (vCPU), storage, and memory. Contrastingly, horizontal scaling is the process by which additional instances of a given VNF are instantiated in the network. It must be noted that there are several drawbacks to horizontal scaling including placement strategies for the newly instantiated VNF instances, the need to manage additional interdependencies between components, ensuring that each instance has redundancy in case of a failure, and the potential of VNF sprawl whereby many instances of a given VNF are severely underutilized.

b) Service Availability Requirements There are two main components required to ensure the service availability requirements of a VNF enabled network. First, it is important to identify system outages and faults and evaluate their impact on the overall network operation. Second, it is important to define a resiliency plan outlining the various strategies and approaches aimed at mitigating the impact of said outages and faults.

i) System Outages There are several faults, which can occur in an NFV enabled network due to internal or external factors. These faults have severe consequences when considering the reliability and the availability of the network. Malicious attacks performed against the network can target virtual components such as the network functions, orchestrators and managers. These attacks can cripple service and access protected information thus compromising the operation and security of the network. Disasters (natural and unnatural) can directly affect an NFV enabled network by targeting its infrastructure components or indirectly by targeting the infrastructure of systems it depends on such as electricity distribution grids. In order to abide by QoS guarantees, NFV networks must possess resilience to ensure that the recovery time after an active fault does not compromise these guarantees. Selecting between migration and re-instantiation after an active fault or failure occurs is a component of the failure management system and contributes to the remediation and recovery of the network.

ii) Resiliency Concepts The NFV ISG have outlined a set of tradeoffs and behavior's relating to resiliency principles in NFV-enabled networks. When converting a traditional network to an NFV-enabled network, certain resiliency measures should be implemented to ensure that the same level of service availability is preserved in the virtualized networks. However, with the transition from physical to virtual, several new resiliency measures must be adopted and their implications on the entirety of the network must be considered. The main goal when placing VNFs is to minimize the recovery time, thereby ensuring the availability of the service to the end-user. In the case of recovering from a fault or failure event, the number of components (VNF instances) should be minimized; however, minimizing the number of components, could have an adverse effect on the resiliency and inherently the availability of the service. If the resiliency is maximized, several other tradeoffs occur. Firstly, the cost of the network operation increases which might violate cost constraints. Second, during the process of virtualization, there is an inherent increase in complexity, something which counteracts the objective of maximizing resilience as it requires a reduction in overall system complexity. Finally, since state management is an essential component of NFV resiliency, and its implementation is a poly-dimensional problem, its implementation can also have adverse effects on the overall resiliency of the network due to its inherent complexity addition. The methods of migration and re-instantiation act as a solution for managing this inherent complexity as they work towards the improvement of overall system resilience.

iii) Service Availability Constraints In an effort to abide by and meet carrier-grade requirements NFV enabled networks should be resilient towards failure events and should aim at providing uninterrupted, continual service to the end-user during these events. Several techniques exist to ensure the above two requirements are met. The on-demand recovery of a VNF instance provides resiliency after a failure event and service continuity can be achieved through VNF migration as it provides instant data recovery while ensuring state preservation. Furthering the notion of resiliency, a healthy network with a carrier-grade service should offer several computational paths (which provide acceptable performance and delay) in the event of a component failure occurs.

iv) Stages of Failure Management Taking the information presented in this section into consideration, Fig 1 illustrates the stages of a failure management system. State S0 defines the normal operating conditions whereby the measured service parameters are acceptable. During this state, the failure management system is attempting to prevent and failures from happening by mitigating the impact of active faults and preventing them from developing into the system. In the case of a failure, state S1 is entered whereby the measure service parameters are unacceptable and the delivered service is severely degraded. In this state, the failure management system attempts to detect the failure.

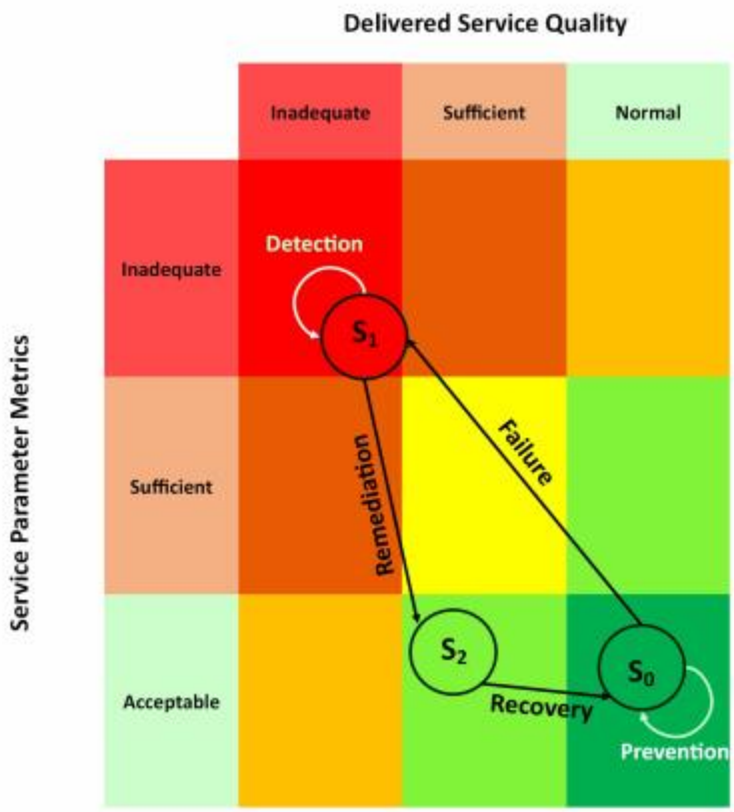

Figure 5.1: Stages of Failure Management

Once the failure is detected, the systems attempt to instantly remediate the issue and enter state S2. This state attempts to recover the service to normal operating conditions such that the measured service parameters are once again acceptable. Migration and re-instantiation techniques would contribute to the recovery stage whereby the system experiences a transition from state S2 back to the original state S0.

5.3.3 Service Levels and their Requirements

In general, service availability is the availability of all virtual and physical elements involved in end-to-end service. ETSI has classified service availability requirements into three levels depending on the customer. Level 1 customers are classified as the most critical and require high availability guarantees. Customers such as government, emergency services, and network operations are categorized as Level 1. Similarly, Level 2 customers also require high availability service, however, they are deemed to be less critical than the customers found in Level 1. Examples of Level 2 customers include enterprises, corporations, and large educational institutions. Finally, Level 3 customers such as ISP traffic can be categorized as the lowest priority service availability. When designing an NFV-enabled network, the service availability levels must be taken into consideration. This means that depending on the network conditions, priority must be given to the most critical customers to ensure their services achieve the high availability requirement. Therefore, the recovery process must be biased towards the critical services and ensuring their restoration is a priority. When selecting to migrate or re-instantiate, the nature of the service and customer are considered when determining the recovery order.

5.3.4 VNF Placement Considerations in a Cloud Environment

Today's cloud environments are composed of numerous geographically distributed and interconnected data centers (DCs). Within each data center, the topology is composed of racks of hosted servers intra-connected through the top of the rack (TOR) switches. This topology directly impacts the server to server latency experienced in data centers. As previously mentioned, it is the responsibility of the CSP to provide VMs and containers to host the VNFs. Through the use of a cloud orchestrator, the CSP provides a mapping between cloud infrastructure (VM, container) and physical infrastructure (server). Once mapped to servers, VMs are able to host VNF instances and through the chaining of several VNF instances, a VNF application. When chaining VNF instances, dependencies are generated between various instances and these dependencies each has their respective delay tolerance thresholds and bandwidth requirements. Both the allocation of cloud infrastructure to physical infrastructure as well as the allocation of VNF infrastructure to cloud infrastructure can impact the QoS and carrier-grade requirements of a given application. It must be noted that the placement of VNFs on VMs is more challenging than the placement of VMs on servers therefore, additional care must be taken to ensure the optimal placement for VNFs in a network. Taking this into consideration, there is a clear correlation between the placement of the VNF and the functions executed by the orchestrator (migration or re-instantiation). After invoking the process of migration or re-instantiation, the new VNF instance will affect delays of the overall VNF application through the new delays between the components of the service chain.

5.3.5 Migration and Re-Instantiation

One of the identified challenges of virtualization arises in the case of a catastrophic event. If such an event were to occur, several components of the VNF infrastructure would be impacted and numerous VNFs would be affected. Since the impact of such an event would be widespread through the network, the recovery effort would significantly increase in magnitude compared to the situation of a single VNF failure. It is possible that recovering the service after such an event would require the migration of VNFs to traverse into the jurisdiction of foreign administrative and regulatory bodies. In other (noncatastrophic) outages cases, we are left to either migrate or re-instantiate while simultaneously trying to maximize the number of service chains preserved. In an effort to preserve service continuity in the case of a hardware failure, regression and pre-emption are two strategies that can be used when there is an insufficient number of available resources in the remaining network. Regression attempts to maximize the number of VNFs being relocated to the new destination without altering the current state of the destination. Pre-emption, on the other hand, suggests that all the VNFs of the failed hardware component should be relocated to the new destination hardware and VNFs currently running on the destination should be suspended to free resources such that the destination possesses enough available capacity to support the relocation in its entirety. In the event that a VNF is impacted by a failure or outage, a live-migration or re-instantiation routine is invoked by the orchestrator in an effort to ensure the continued adherence to the carrier-grade requirements. Live migration can be defined as the uninterrupted relocation of a VNF from its current server to a selected destination server while preserving information about the VNF's states. Re-instantiation, on the other hand, can be defined as the initialization of a VNF (which mirrors the type of the affected VNF) on a new server without the preservation of internal states. With these concepts and requirements in mind, the remainder of the chapter discusses the proposed approach in terms of the MILP model and heuristic solution.

5.4 Optimization Model Formulation

The proposed model is solved using the IBM ILOG CPLEX optimization tool [14]. The model aims at minimizing the downtime of migrating or re-instantiating a VNF while satisfying different placement, availability, and re-instantiation/migration constraints.

5.4.1 Approach Model

When developing the optimization model, the first stage was to define the VNF enabled network. The network definition consists of the infrastructure (physical and virtual), the VNFs, as well as the SFCs. The following describes each of these components and their implementation in the model.

1) Network Infrastructure Topology: A NFV-enabled network consists of forwarding devices such as programmable switches and IP routers along with NFV nodes which, depending on their allocated resources, can host several VNFs. Multiple VNFs running on the same node is possible through the use of independent clusters of virtual machines (containers). The topology also contains SDN controllers which are tasked with managing traffic through the various VNF instances while ensuring that the performance requirements of the SFC are met.

2) VNFs: VNFs are network functions which have been decoupled from the underlying physical hardware through the process of network function virtualization. These virtual functions are deployed by the network operator on infrastructure provided by the CSP (VMs, containers). Some examples of virtualized network functions include firewalls and load balancers.

3) SFCs: A SFC is a defined sequence of VNFs which enables a service to be provided to the end-user. SFCs are subject to additional constraints in order to ensure functionality such as the delay between VNF instances, the delay between components, bandwidth, and availability.

5.4.2 Problem Constraints

The following is an outline of the various constraints used in the formulation of the optimization problem. These constraints are implemented to ensure that QoS and SLA guarantees are met in addition to the minimization of downtime associated with migration and re-instantiation. The constraints are expressed both qualitatively and quantitatively through their mathematical notation.

a) Computational resources constraint

Using this constraint, the proposed model selects a set of servers that can satisfy the VNFs' resources demand. In this model, the resources are CPU cores and memory.

b) Network delay constraint

Using this constraint, the proposed model filters the servers to select the ones that do not violate the delay tolerance between the dependent VNFs in an SFC.

c) Availability constraints

Each VNF can be either a sponsor and/or a dependent one. In order to maintain the availability of the SFC chain, the proposed model defines the following constraints:

d) Affinity constraint

This ensures that the sponsor VNF and its dependents should be hosted on the same server if the dependents have tolerance time lower than the sponsor's recovery time.

e) Anti-affinity constraint

On the contrary, the dependent VNFs and their sponsor should be deployed on different servers if the dependents have a higher tolerance time compared to their sponsor's recovery time.

f) SDN network controller convergence constraint

Using this constraint, the model selects a set of servers that minimizes the convergence delay of the SDN network controller. This delay is the time needed by the controller to reflect the changes (such as new VNFs' placements) in the computational path of the VNFs of an SFC in case of migration or re-instantiation process.

g) Service discovery delay constraint

Using this constraint, the model selects a set of servers that minimizes the service discovery delay. The latter is generated from the VNFs' migration or re-instantiation process. It is defined as the VNF registration time with a service broker, which is responsible for collecting and maintaining meta-data information of the federated VNF cluster.

5.4.3 Notations and decision variables:

In this model, the set of VNFs is denoted as V, the total number of VNFs is denoted as N_v, the set of servers is denoted as S, the total number of servers is denoted as N_s, the computational resources are denoted as Res, the set of computational resources types is denoted as R, the SDN controllers set is denoted as C, and the set of dependent VNF is denoted as V^D. The original placement of the VNFs is denoted by $X^{original}$. Also, the tolerance time and recovery time are denoted as T^T and T^R respectively. SO and CO represent the hosting server's delay overhead and the network convergence delay overhead of the selected SDN controller respectively. As for delays, the delay generated from the VNF placement is denoted by D^p, the delay between server S and S' is denoted by $D^{SS'}$, the delay between the hosting server and the SDN controller is denoted as D^{CS}, and the delay resulting from the overhead of migration or re-instantiation decision is denoted as D^{Dec}. Note that Dec represents either migration or a re-instantiation decision. As for the binary decision variables, they are defined as follows:

$$X_{vs} = \begin{cases} 1 \;\; if \;\; VNF \;\; "v" \, is \;\; placed \;\; on \;\; server \;\; "\text{s}" \\ 0 \;\; otherwise \end{cases}$$

$$Y_v^{Dec} = \begin{cases} Y_v^{Migration} = 1 \;\; if \;\; VNF \;\; "v" \, will \, be \; migrated \\ Y_v^{Migration} = 0 \;\; otherwise \\ Y_v^{Re\text{-}Instantiation} = 1 \;\; if \;\; VNF \;\; "v" \, will \, be \; re\text{-}instantiated \\ Y_v^{Re\text{-}Instantiation} = 0 \;\; otherwise \end{cases}$$

5.4.4 Mathematical Formulation

The objective function is:

$$min \quad \sum_{v}^{N_V} DownTime_v \qquad (1)$$

It is subjected to the following constraints:

- *Boundary Constraints:*

$$X_{vs}, Y_v^{Dec} \in \{0,1\} \quad \forall \, v \in V, \, s \in S \qquad (2)$$
$$Dec \in \{Re\text{-}Instantiation, Migration\}$$

$$DownTime_v \geq \quad 0 \quad \forall v \in V \qquad (3)$$

- *Placement Constraints:*

$$\sum_{v}^{N_V} (X_{vs} \times Res_{vr}) \leq Res_{sr} \quad \forall \, s \in S, r \in R \qquad (4)$$

$$\sum_{s}^{N_S} X_{vs} = 1 \quad \forall \, v \in V \qquad (5)$$

- *Availability Constraints:*

$$(X_{vs} + X_{v's}) \leq 2 \quad or \quad (X_{vs} + X_{v's}^{original}) \leq 2$$
$$\forall\, s \in S, v \in V, v' \in V^D, T_{v'}^T \leq T_v^R \tag{6}$$

$$(X_{vs} + X_{v's}) \leq 1 \quad or \quad (X_{vs} + X_{v's}^{original}) \leq 1$$
$$\forall\, s \in S, v \in V, v' \in V^D, T_{v'}^T \geq T_v^R \tag{7}$$

- SFC *Delay & Re-Instantiation/Migration Constraints*:

$$Y_v^{Re\text{-}Instantiation} + Y_v^{Migration} = 1 \quad \forall\, v \in V \tag{8}$$

$$D_v^P = X_{vs} \times \left[\left(\sum_{d}^{N_S} X_{vd}^{original} \times D_{sd}^{SS'} \right) + D_{cd}^{CS} \right] \quad \forall v \in V, s \in S, c \in C \tag{9}$$

$$D_v^{Dec} = (SO_v^{Dec} + CO_c^{Dec}) \times Y_v^{Dec} \quad \forall c \in C, v \in V \tag{10}$$

$$DownTime_v = D_v^{Dec} + D_v^P \quad \forall\, v \in V,$$
$$Dec \in \{Re\text{ - }Instantiation, Migration\} \tag{11}$$

Constraint (2) determines that the placement and re-instantiation/migration decision variables are binary numbers. Constraint (3) determines that the VNF downtime should be a positive number. Constraint (4) determines that the servers should have enough computational resources to host the re-instantiated or migrated VNF. Constraint (5) determines that only one server can host a VNF. To maintain the interdependency relationship between different VNFs, constraint (6) determines that a VNF shares the same server with its dependent VNF(s) if the latter cannot tolerate the absence of their sponsor VNF. On the contrary, constraint (7) determines that a VNF and its dependent(s) should share different servers if the dependent(s) can tolerate the sponsor's absence. Constraint (8) determines that a

VNF can be either migrated or re-instantiated. Constraints (9) and (10) determine that a VNF should be placed on a server that satisfies the delay requirements while minimizing the migration or re-instantiation overheads. Based on the previous constraints, the model selects either migration or re-instantiation of a VNF while minimizing its downtime. Therefore, constraint (11) shows that the downtime of each VNF is calculated in terms of the placement latency and the overhead delay resulted from either the migration or the re-instantiation process.

5.5 MILP Model Complexity

In order to determine the complexity of the proposed MILP model, we use the reduction method. In this section, we reduce the problem to a bipartite matching one in order to build our model accordingly[25]. Any scheduling problems can be interpreted as a triplet $a \mid b \mid c$, where a represents the problem environment, *b* represents the problem constraints, and *c* represents the objective function of the problem [26]. These triplet fields vary depending on the scheduling problem nature. Since the proposed approach addresses the allocation problem of VNF components set *(V)* on the available servers *(S)* with an objective function to minimize the communication delay between the dependent components, it can be formulated as a special case of the transportation problem. The formulation for the problem can be represented as $S_s \mid V_v \mid D(x)$ where the S_s is the problem environment consisting of s different parallel servers, V_v defines the VNF job v that can proceed on a single server s, and $D(x)$ represents the objective function to be optimized. In this special case, the problem is known as a constrained bipartite matching problem. $G = (V, S, a)$ represents the bipartite graph that consists of VNF components nodes as set V , server nodes as set S, and arc a connecting the two sets. The arc $a = v, s$ assigns the VNF component v of set V to server s

of set S, and it represents the decision variable P_{vs} defined in the previous section. Said that and using the Hopcroft-Karp algorithm, the bipartite maximal matchings are determined in polynomial time to the number of edges and vertices [27]. Thus, this type of bipartite matching problem that is formulated using linear programming models is categorized as an NP-hard problem, and by reduction, the proposed MILP model is NP-hard. Therefore, the proposed MILP model would be solvable for small- scale DC networks [28]. With this in mind, this chapter proposes a heuristic approach to address the large-scale DC networks.

5.6 Heuristic Solution

When considering the NP-hard MILP optimization model, it is evident that as network size increases, the limitations imposed on computational resources and the time required to converge to a solution will render this model ineffective. To solve this, a heuristic solution has been developed. Given a set of servers S and a set of VNFs participating in an SFC, the heuristic algorithm finds a feasible near-optimal VNF migration or re-instantiation placement solution compared to the MILP optimal solution. The solution produced by the proposed heuristic algorithm satisfies all the constraints outlined in the MILP model however, it relaxes the objective function. In the algorithm, different subroutines are executed to advise migration or re-instantiation decision. First, an analysis of the types of participating VNF in a given SFC is conducted. The VNF types are then divided into sub-groups according to their inherited dependency from the VNF forwarding graph (VNFFG). Each sub-group is assigned a criticality attribute based on the communication delay tolerance of the participating VNF types. If an undercount group is found, shared VNF types from another subgroup is held. It is necessary to note that a group is considered as an undercount one when it contains less than three VNF types. Once the grouping has been completed, the algorithm goes on to build a weighted system model graph *G(V, E, w)* where the vertices *V* represents the set of available servers in a given data center, the edges *E(v, v')* represents the logical communication link between the servers, and the weights w(v, v') represents the data communication delay between the servers. Then the betweenness centrality (BC) is calculated for each vertex. The calculation of the betweenness centrality is based on the number of the shortest paths from the source node *(s)* to sink node *(t)* that passes through a specific node. Calculating the BC identifies the servers that can be anchors for the nodes that will be migrated or re-instantiated in the defined subgroups. The algorithm will calculate the migration and re-instantiated delays in two different sets of the identified anchor nodes while considering VNF types that will be hosted on these anchor nodes. A quick sort is conducted on the two sets to identify the anchors that result in the lowest delay placement with a tag that identifies if a migration or re-instantiation should be conducted. Then starts by migrating or re-instantiating the most critical VNF components to the new placement base on the decision conducted from the quick sort while satisfying the functional constraints. The generated decision of migration or re-instantiation placement solution is considered the best effort to achieve the minimum delay and service interruption of the SFC while maximizing the count of the possible computation paths.

Table 5.1: Small-Scale Network Topology

Servers	Instances	HSS	MME	SGW	PGW
15	6	1	2	1	2

5.7 Results and Analysis

The generation of the dataset was executed in Java along with the heuristic algorithm. The optimization model was executed using CPLEX. All results were obtained using a PC with an Intel ® Core™ i7-8700 CPU @ 3.20 GHz CPU, 32 GB RAM, and an NVIDIA GeForce GTX 1050 Ti GPU. The results presented below show the implementation of our proposed heuristic solution compared to the optimal solution across three different network topologies. The three network topologies can be classified as small, medium, and large.

5.7.1 Small-Scale Network

The small-scale network topology used is described through the server-instance ratio, as well as the type of each instance listed in Table 5.1. Taking this topology into consideration, there are 4 possible computational paths available to complete the SFC.

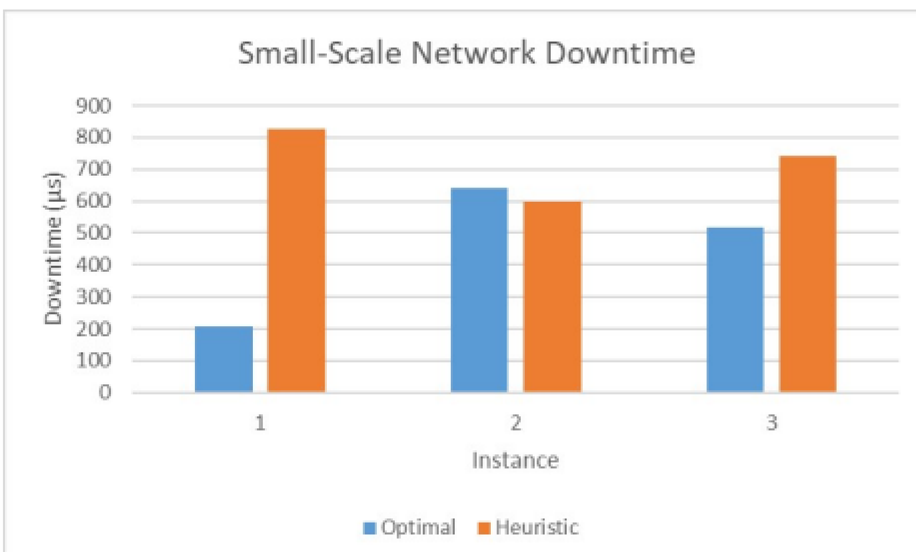

Figure 5.2: Small-Scale Network Downtime

In order to evaluate the performance of our heuristic solution, three components were selected for migration or re-instantiation. The criteria for assessing the performance of our algorithm include the downtime of service, and the resulting delay post-migration or re-instantiation. Fig. 5.2 displays the downtime associated with the migration or re-instantiation of each instance using the optimal and heuristic solutions.

As seen through these results, the heuristic achieves acceptable downtime results when selecting between the migration or re-instantiation of an instance. Fig. 5.2 displays the downtime associated with the migration or re-instantiation of each instance using the optimal and heuristic solutions. As seen through these results, the heuristic achieves acceptable downtime results when selecting between the migration or re-instantiation of an instance. Fig. 5.3 forming an SFC computational path pre and post-migration or re-instantiation. From these results, it can be stated that the heuristic solution achieves comparable performance compared to the optimal model. This statement can be further supported by considering the delay across the entire SFC. Fig. 5.4 compares the delay experienced across all computational paths initially and post-migration or re-instantiation. As seen through these results, the heuristic placement achieves acceptable delay across all computational paths which approaches the delay experienced through the optimal placement of the migrated or re-instantiated components.

5.7.2 Medium-Scale Network

The medium-scale network topology used is described through the server-instance ratio, as well as the type of each instance listed in Table 5.2. Taking this topology into consideration, there are 36 possible computational paths available to complete the SFC.

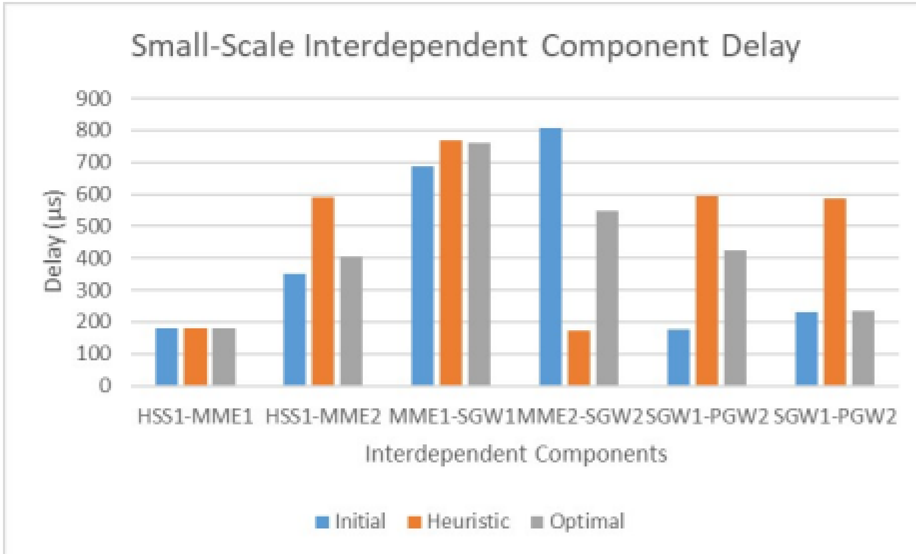

Figure 5.3: Small-Scale Interdependent Component Delay

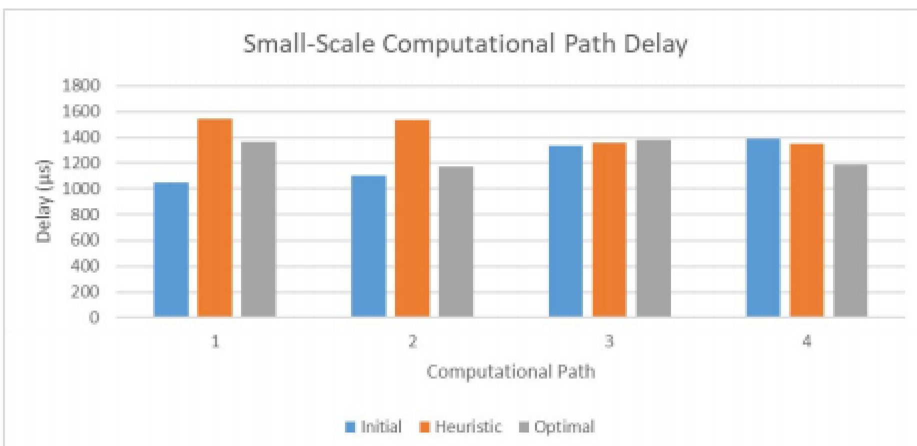

Figure 5.4: Small-Scale Computational Path Delay

Table 5.2: Medium-Scale Network Topology

Servers	Instances	HSS	MME	SGW	PGW
30	10	3	2	3	2

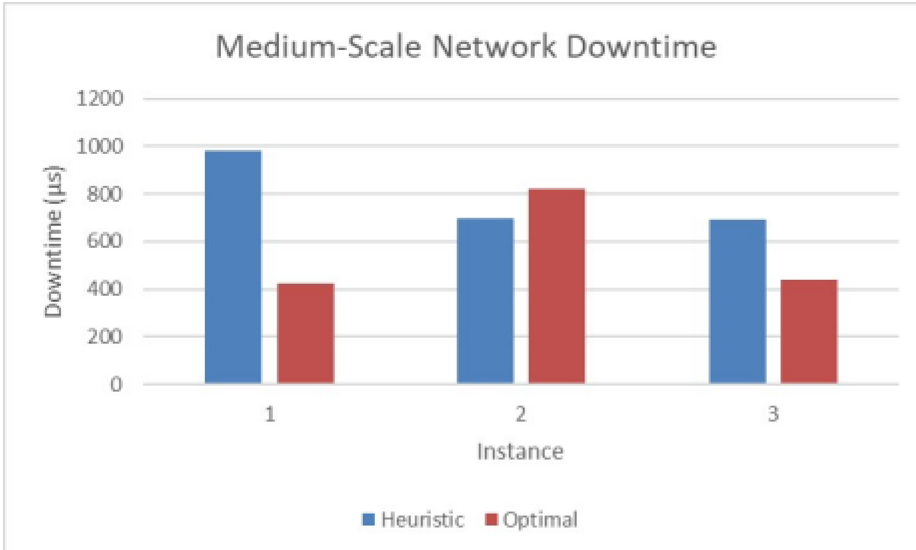

Figure 5.5: Medium-Scale Network Downtime

Three instances were selected for migration or re-instantiation to evaluate the performance of our proposed algorithm. Fig. 5.5 presents the downtime associated with the migration or re-instantiation of each instance when using the optimal and heuristic solution. As previously observed through the small-scale network results, the downtime observed in the medium-scale network using the heuristic solution is comparable to that observed using the optimal solution. Fig. 5.6 displays the initial and post-migration or re-instantiation delays observed across all SFC computational paths.

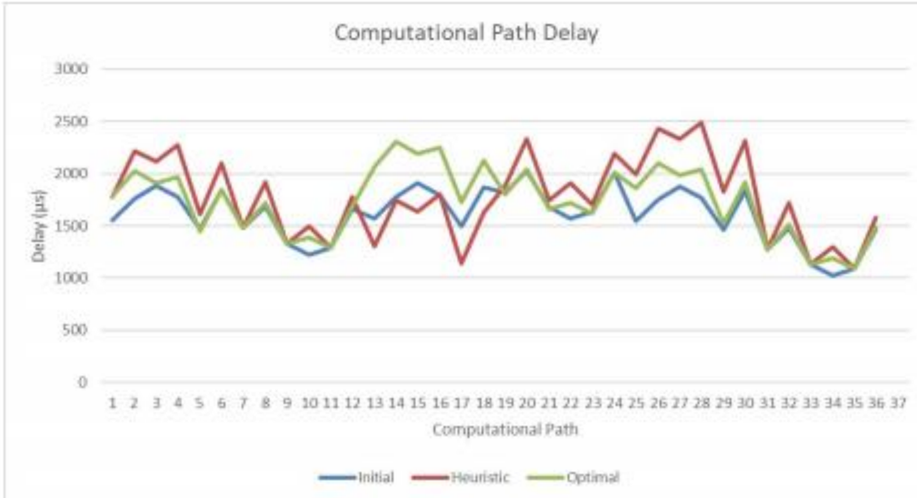

Figure 5.6: Medium-Scale Computational Path Delay

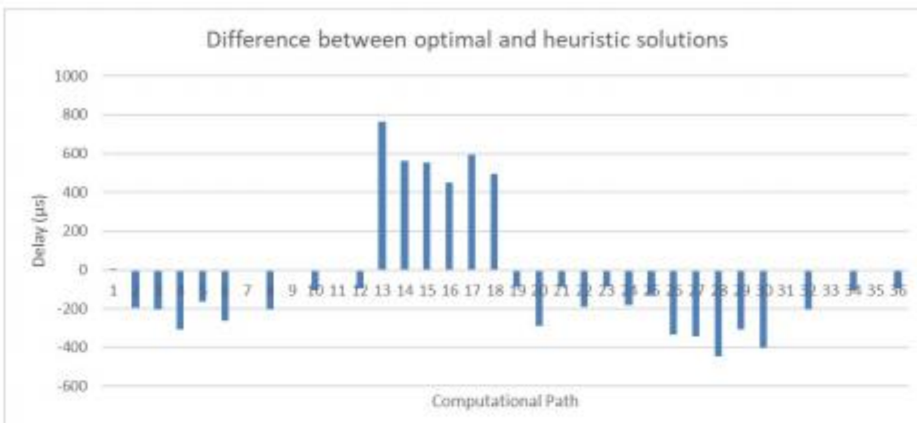

Figure 5.7: Difference between Optimal and Heuristic Solutions

These results suggest that the heuristic solution produces near-optimal results when selecting between the candidate servers for migration or the re-instantiation of components. As an illustration of the previous statement, Fig. 5.7 presents the difference between the delay experienced between the optimal and heuristic placements. The difference between the delays experienced across all computational paths is minimal, and there is a set of paths that experience significantly reduced delays when using the heuristic solution.

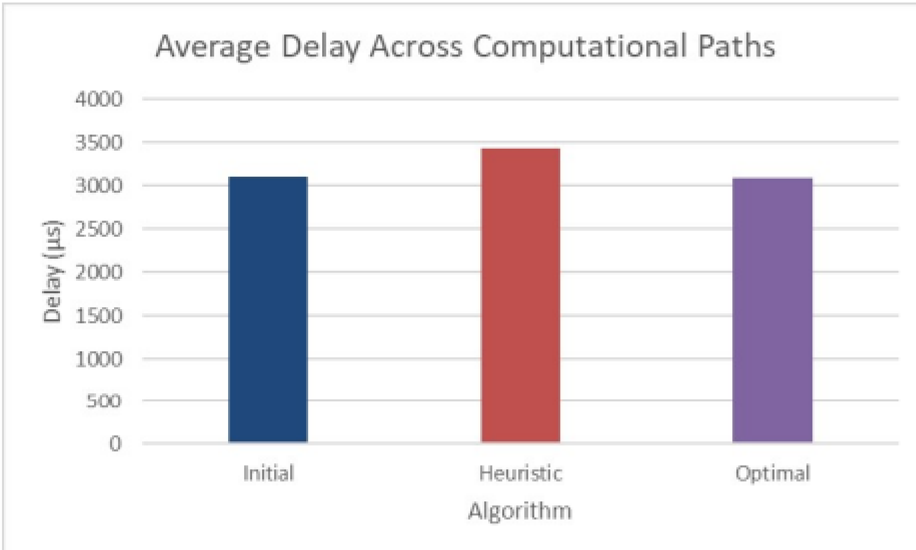

Figure 5.8: Average Delay Across Computational Paths - Large Scale

5.7.3 Large-Scale Network

The large-scale network topology used is described through the server-instance ratio, as well as the type of each instance listed in Table 5.3. Due to the nature of this topology, there are a total of 345,000 possible computational paths that can be traversed to complete the SFC. Three instances were selected for migration or re-instantiation to evaluate the performance of our proposed algorithm in this topology. Fig. 5.8 presents the average delay across all computational paths pre and post-migration or re-instantiation for both the heuristic and optimal solutions. Coinciding with the results observed in the previous network topologies, the delay observed across all computational paths when implementing the heuristic solution in the large-scale network is comparable to those obtained in the small and medium-scale networks.

Table 5.3: Large-Scale Network Topology

Servers	Instances	HSS	MME	SGW	PGW
300	98	23	20	30	25

5.8 Conclusion

In conclusion, the work presented in this chapter proposes a heuristic solution for the selection of the migration or re-instantiation of a particular VNF instance forming an SFC as well as the placement server out of a set of candidate network servers. The proposed heuristic solution has shown near-optimal performance in small, medium, and large-scale networks in terms of downtime associated with the migration or re-instantiation of a particular instance as well as the delay experienced across the various computational paths forming the SFC in the given network. The advantages of using the proposed heuristic solution instead of the optimal solution are inherently observed through the reduction in time complexity of the system as well as through the fact that the optimization problem proves to have rigid constraints which oftentimes do not produce a global optimum and therefore do not result in a solution. In contrast, the heuristic solution always produces an output which satisfies the constraints imposed during the optimization problem formulation. The main objectives of this chapter were to model the carrier-grade functionality and interdependency constraints imposed on SFCs as well as the criticality and priority of the various service levels present in NFV applications. The work presented in this chapter successfully demonstrates the implementation of the above model requirements in an intelligent orchestrator which is able to select between the migration and re-instantiations of a given VNF instance which minimizes latencies between interdependent instances and improves the overall quality of the SFC. Future work in this area would see the expansion of the functionalities of the intelligent orchestrator to address other services provided by NFV MANO.

References

[1] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022,”, 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/white-paper-c11-738429.pdf

[2] F. Jejdling, et al., “Ericsson Mobility Report June 2018”, Ericsson, Stockholm, Sweden, 2018. [Online]. Available: https://www.ericsson.com/assets/local/mobilityreport/documents/2018/ericsson-mobility-report-june-2018.pdf

[3] D. George, T. Hatt, et al., “Global Mobile Trends 2017,” GSMA Intelligence, 2017. [Online]. Available: https://www.gsmaintelligence.com/research/?file= 3df1b7d57b1e63a0cbc3d585feb82dc2&download

[4] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, “NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC),” Network, vol. Nov, no. December, pp. 18--26, 2014.

[5] M. Chiosi, et al., “Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges & Call for Action,” ETSI, Darmstadt, Germany, 2012. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper.pdf JOURNAL OF XXX 11

[6] H. Hawilo, M. Jammal, A. Shami, "Orchestrating network function virtualization platform: Migration or re-instantiation?" in Cloud Networking (CloudNet), 2017 IEEE 6th International Conference on, September 2017, Prague (Czech Republic).

[7] Network Function Virtualisation (NFV); Reliability; Report on the Resilience of NFV-MANO Critical Capabilities, ETSI Standard RGR/NFV-REL007ed112, 2017. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/NFVREL/001_099/007/01.01.02_60/gr_NFV-REL007v010102p.pdf

[8] B. Han, V. Gopalakrishnan, L. Ji et al.: 'Network function virtualization: challenges and opportunities for innovations', IEEE Commun. Mag., 2015, 53, (2), pp. 90–97

[9] A. Gember-Jacobson and A. Akella, "Improving the Safety, Scalability, and Efficiency of Network Function State Transfers," Proc. 2015 ACM SIGCOMM Work. Hot Top. Middleboxes Netw. Funct. Virtualization - HotMiddlebox '15, pp. 43–48, 2015.

[10] S. Rajagopalan, D. Williams, H. Jamjoom, and A. Warfield, "Split/merge: System Support for Elastic Execution in Virtual Middleboxes," Usenix, pp. 227–240, 2013.

[11] S. Woo, J.Sherry, S. Han, S. Moon, "Elastic Scaling of Stateful Network Functions," Nsdi '18, 2018.

[12] D. Systems, J. Xia, Z. Cai, and M. Xu, "Optimized Virtual Network Functions Migration for," 2016.

[13] J. Xia, D. Pang, Z. Cai, M. Xu, and G. Hu, “Reasonably migrating virtual machine in NFV-featured networks,” Proc. - 2016 16th IEEE Int. Conf. Comput. Inf. Technol. CIT 2016, 2016 6th Int. Symp. Cloud Serv. Comput. IEEE SC2 2016 2016 Int. Symp. Secur. Priv. Soc. Netwo, pp. 361–366, 2017.

[14] D. Cho, J. Taheri, A. Y. Zomaya, and P. Bouvry, “Real-Time Virtual Network Function (VNF) Migration toward Low Network Latency in Cloud Environments,” IEEE Int. Conf. Cloud Comput. CLOUD, vol. 2017–June, no. 1, pp. 798–801, 2017.

[15] V. Eramo, M. Ammar, and F. G. Lavacca, “Migration energy aware reconfigurations of virtual network function instances in NFV architectures,” IEEE Access, vol. 5, pp. 4927–4938, 2017.

[16] J. Zhang, D. Zeng, L. Gu, H. Yao, and M. Xiong, “Joint Optimization of Virtual Function Migration and Rule Update in Software Defined NFV Networks,” 2017 IEEE Glob. Commun. Conf. GLOBECOM 2017 - Proc., vol. 2018–Janua, pp. 1–5, 2018.

[17] A. Gumaste, A. Kushwaha, and A. Jukan, “On the Benefits of Programmable Optics for Post-Failure VM Migrations in Data-Centers,” 2018 Opt. Fiber Commun. Conf. Expo., pp. 1–3, 2018.

[18] F. Carpio, A. Jukan, and R. Pries, “Balancing the migration of virtual network functions with replications in data centers,” IEEE/IFIP Netw. Oper. Manag. Symp. Cogn. Manag. a Cyber World, NOMS 2018, pp. 1–8, 2018.

[19] C. Sun, J. Bi, Z. Meng, T. Yang, X. Zhang, and H. Hu, “Enabling NFV Elasticity Control with Optimized Flow Migration,” IEEE J. Sel. Areas Commun., vol. 8716, no. c, pp. 1–15, 2018.

[20] Y. Lin, U. C. Kozat, J. Kaippallimalil, M. Moradi, A. C. K. Soong, and Z. M. Mao, “Pausing and Resuming Network Flows using Programmable Buffers,” Proc. Symp. SDN Res. - SOSR ’18, pp. 1–14, 2018.

[21] Sandvine, “The Global Internet Phenomena Report October 2018”, 2018. [Online]. Available: https://www.sandvine.com/hubfs/downloads/phenomena/2018- phenomena-report.pdf

[22] M. Chiosi, S. Wright, et al., “Network Function Virtualizatsation (NFV): Network Operator Perspectives on Industry Progress,” ETSI, Dusseldorf, Germany, 2014. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

[23] Network Function Virtualisation (NFV); Reliability; Management and Orchestration, ETSI Standard DGS/NFV-MAN001, 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/nfvman/001_099/001/01.01.01_60/gs_nfvman001v010101p.pdf

[24] IBM, “CPLEX Optimizer,” https://www01.ibm.com/software/commerce/optimization/cplex-optimizer/, 2019.

[25] M. L. Pinedo, “Deterministic models: Preliminaries,” in Scheduling— Theory, Algorithms, and Systems. Boston, MA, USA: Springer, 2011.

[26] Cornell University. (2017). Matchings. [Online]. Available: http:// www.cs.cornell.edu/courses/cs6820/2016fa/handouts/matchings.pdf

[27] R. Coric, M. Dumic, and D. Jakobovic, "Complexity comparison of integer programming and genetic algorithms for resource constrained scheduling problems," in Proc. Int. Conv. Inf. Commun. Technol., Electron. Microelectron., 2017, pp. 1182–1188.

[28] C. Guo et al., "Pingmesh: A large-scale system for data center network latency measurement and analysis," in Proc. ACM Conf. Special Interest Group Data Commun. (SIGCOMM), Aug. 2015, pp. 139–152.

Chapter 6

Conclusion

The connectivity scale that we have reached today has exceeded the expectations of many researchers and scientists. Today, most of our tasks and applications have some sort of connectivity, technologies such as the IoT and 4/5G networking have allowed for massive expansion and immense scaling of device connectivity, this expansion has put NSPs under the pressure of adapting to the needs of their users. However, the major part of the networking environments today is rigid, and the legacy systems that operate today cannot scale as fast as the demand; this is due to the strong dependency between the network functionalities that are offered and their proprietary hardware. This results in complex scaling procedures as it becomes very costly for NSPs to add, modify, or remove functionalities from their services. this obstacle can be clearly seen from the increased maintenance and operational costs that occur when an attempt to update the network is issued. In addition, it is difficult to find the required skills set to handle these very specific hardware coupled services.

For this reason, NSPs have looked for virtualizing these services through implementing the state-of-the-art NFV/SDN approaches that decouple the software components from their dedicated hardware. SDN allows for a centralized controller to take place handling all the control plane functionalities of the network, this allows the hardware components of the network to become free from decision-making tasks and only work as devices that execute the SDN controller's commands. In parallel to this task, the SDN infrastructure works seamlessly with NFV to even add more advantages and enhance the network performance further. NFV allows for the software components in all the network devices to become virtualized and have the capability to be hosted on unified COTS servers. With this, NSPs gain immense power to quickly add, modify, terminate network services in real-time. The cost aspect of such an approach also becomes much lighter on the NSPs allowing them to invest further in innovative solutions and work towards further improving the quality of their services.

These approaches do enhance network architecture greatly. Nevertheless, they introduce new challenges to the NSPs that need to be addressed. This thesis has focused in detail on solving several of those challenges to pave the road towards a seamless transformation of network services from their current legacy infrastructure towards the NFV/SDN enabled virtualized environment. Chapter 2 has considered the microservices architecture as a platform to host an NFV based cloud application, the chapter has looked and several challenges for combining these approaches and have introduced a MILP model scheduler to mitigate the challenge of network delay by optimizing the entity scheduling in the NFV microservices environment. Chapter 3 proposed and SDN/NFV solution to transform the EPC to a completely virtualized core towards efforts of enabling the uprising of 5G networking environments. Chapter 4 has solved in detail the placement problem of the NFV instances by introducing both a MILP model and a heuristics approach that aims at minimizing the delays of the network while placing those services. Chapter 5 has looked at a key challenge in an NFV infrastructure which is failure handling. The chapter proposed both an optimization and a heuristics approach to tackle the decision of migrating or re-instantiating a network function upon failure. The summary of each chapter is described as follows.

6.1 Chapter 2 Summary

This chapter has focused on evaluating the microservices architecture to be selected as an enabler for NFV hyper-scaling in a cloud environment. The chapter discussed in detail the key challenges that face this architecture such as, VNFCs' networking complexity, service discovery, monitoring, logging, meta-data collection, and routing convergence. In addition, this chapter introduced a scheduling solution that satisfies both the SLA and QoS requirements for the VNFC placement. The proposed solution was presented as a MILP model and has evidently shown the advantages of such an approach.

6.2 Chapter 3 Summary

With the emergence of 5G networks and the extensive research the community has invested in perfecting this technology, NSPs have started to prepare their infrastructures to host 5G capabilities, with virtualization being a key enabler for dynamic networking and real-time adaptation to the end-user needs NSPs looked for NFV/SDN technologies to drive their transformation to a completely virtualized environment. This chapter proposed a novel solution for transforming the traditional EPC into becoming fully virtualized and used as a stepping stone for the 5G core network. The solution was based on harmonization between NFV and SDN to manage the vEPC. The chapter focused on creating a seamless management infrastructure for the vEPC in multi-tenant private and public clouds.

6.3 Chapter 4 Summary

When transforming into a virtualized environment NSPs now have the freedom of deploying and placing network functions on-demand, this allows to quickly generate SFCs to serve the end-user in real-time. But the ability to deploy their software functions onto any

COTS server poses many challenges, these include resource optimization, delay minimization, QoS requirements. This chapter proposed a novel solution to mitigate those challenges by optimally choosing the placement locations of the VNFs requested to generate SFCs. A MILP model was designed to optimize the placement while considering both functional and non-functional-based constraints with the goal of minimizing the end-to-end and intra-delays. In addition to the MILP model, a heuristics approach was implemented to mitigate the time complexity of the MILP model and adapt this approach to real-world scenarios in the cloud. The presented heuristics algorithm is the Betweenness centrality Algorithm for Component Orchestration of NFV platform (BACON). Both approaches enhance the reliability of the SFCs by working towards maximizing the count of the functional group members. The work presented in this chapter was also compared to two algorithms to highlight its performance, namely, an NFV-agnostic algorithm and a greedy-k-NFV approach, both the MILP and BACON outperformed the greedy algorithms and showed significant potential for real-world adaptation into cloud applications.

6.4 Chapter 5 Summary

Enhancing the portability, interoperability, performance, reliability, security, and management of networks are the main goals NSPs are working towards to ensure that their infrastructures are capable of handling the vast amounts of data and traffic that emerged with introducing the 4G and 5G networks. While working towards achieving those goals, NSPs still need to meet strict QoS and SLA requirements. One of the main challenges that face NSPs is ensuring the high availability of their services for mission-critical applications such as, emergency services, medical services, and financial services. this chapter dealt with one major aspect of these services which the decision between migrating or re-instantiating a service after its failure. The work in this chapter considered various QoS constraints while choosing the optimal placement of a new service after its failure. A MILP model and a heuristic solution were presented to solve this problem. The MILP model showed very promising results in choosing the optimal decision, and the heuristics solution has given near-optimal results but with the main advantage of much less execution time making it suitable for real-life scenarios.

Thesis Future Work

This section discusses the open challenges and future areas of research in the area of NFV/SDN based cloud applications.

6.5 Elasticity mechanisms of NFV based cloud applications

Efficient resource provisioning is a must towards creating an elastic framework. To tackle the challenge of elasticity in an NFV-based cloud application, scaling policies have to be investigated towards defining horizontal or vertical scaling techniques that allow the rapid expansion of the application, this can be approached by creating an information gathering environment to fetch and store key characteristics about the cloud application, such as energy consumption and traffic characteristics. With these mechanisms in place, storage can be considered as another key aspect of extending the optimal deployment strategies of the NFV cloud application. In this area, the failure of instances has to be closely studied to understand and place certain mechanisms to deal with critical information that each instance holds and that can be compromised when certain failures occur in the application instances.

6.6 Creation of a unified objective-aware management environment

When looking at the details of each chapter in this work, it is key to create a unified environment that considers all the discussed and solved challenges at once. Such environment would be designed in a way to harmonize all the previously discussed objectives and combine them towards creating an overall comprehensive management environment that has the goal of ensuring the QoS and SLA requirements while considering all the key aspects mentioned in this work, these aspects range from the intra-delays between instances up to advanced failure handling techniques that protect the NFV application and ensures optimality in terms of resource allocation, delays, and high availability of each service, by doing so, NSPs can significantly reduce their CAPEX and OPEX allowing them to more effectively manage their infrastructures and deliver very high-quality services to the end-users.

7 Curriculum Vitae

Name: Hassan Hawilo

Post-secondary Education and Degrees:

Beirut Arab University
Beirut, Lebanon
2007 – 2012 B.A.

The University of Western Ontario
London, Ontario, Canada
2014 - 2015 M.A.

Honors and Awards:

NSERC, The Alexander Graham Bell Canada Graduate Scholarships-Doctoral Program (CGS D)" scholarship, ON Canada, 2017-2019.

Ontario Graduate Scholarships (OGS) and QEII Graduate Scholarships in Science and Technology (QEII-GSST), ON Canada, 2016-2017.

Best Presentation Award, NSERC Symposium, Western University, ON Canada, 2017.

Graduate Student Award for Excellence in Research, Electrical and Computer Engineering Department, Western University, ON Canada, 2015,2019.

Best IEEE Computer Society and WIE Volunteer, 2017-2018.
First-place award in the IEEE ComSoc Lebanon Chapter Second Student Competition Event, 2012.

Dean Honor List Recipient for 5 years of engineering at BAU (Awarded with one time 60 % and three times with 40% tuition fee waiver scholarships), 2007-2012.

Related Work Experience

Industrial Maintenance and Automation Supervisor
SUKOMI – AVERDA Group.
2012 – 2013

Teaching Assistant.
The University of Western Ontario
2014 – 2015

Research Assistant.
The University of Western Ontario (OC2 Lab) in collaboration with Ericsson Research.
2014 - 2015

Publications:

1. 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.-Dec. 2014.
2. H. Hawilo, A. Kanso and A. Shami, "Towards an Elasticity Framework for Legacy Highly Available Applications in the Cloud," *2015 IEEE World Congress on Services*, New York, NY, 2015, pp. 253-260.
3. M. Jammal, H. Hawilo, A. Kanso and A. Shami, "Mitigating the Risk of Cloud Services Downtime Using Live Migration and High Availability-Aware Placement," *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Luxembourg City, 2016, pp. 578-583.
4. H. Hawilo, M. Jammal and A. Shami, "Orchestrating network function virtualization platform: Migration or re-instantiation?," *2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*, Prague, 2017, pp. 1-6.
5. H. Hawilo, L. Liao, A. Shami and V. C. M. Leung, "NFV/SDN-based vEPC solution in hybrid clouds," *2018 IEEE Middle East*

and North Africa Communications Conference (MENACOMM), Jounieh, 2018, pp. 1-6.

6. M. Jammal, H. Hawilo, A. Kanso and A. Shami, "ACE: Availability-Aware CloudSim Extension," in *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1586-1599, Dec. 2018.
7. H. Hawilo, M. Jammal and A. Shami, "Network Function Virtualization-Aware Orchestrator for Service Function Chaining Placement in the Cloud," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 643-655, March 2019.
8. H. Hawilo, M. Jammal and A. Shami, "Exploring Microservices as the Architecture of Choice for Network Function Virtualization Platforms," in *IEEE Network*, vol. 33, no. 2, pp. 202-210, March/April 2019.
9. M. Jammal, H. Hawilo, A. Kanso, A. Shami,” Generic input template for cloud simulators: A case study of CloudSim.” *Softw: Pract Exper*. 2019; 49: 720– 747.