Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

A Performance Evaluation Model for Network Function Virtualisation on 5G Networks

Cristoffer Leite da Silva

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientadora
Prof.a Dr.a Priscila Solis Barreto

Brasília
2019

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

A Performance Evaluation Model for Network Function Virtualisation on 5G Networks

Cristoffer Leite da Silva

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Prof.a Dr.a Priscila Solis Barreto (Orientadora)
CIC/UnB

Prof. Dr. Jacir Luiz Bordim
CiC/UnB

Prof. Dr. Paulo Roberto de Lira Gondim
ENE/UnB

Prof.a Dr.a Genaina Nunes Rodrigues
Coordenadora do Programa de Pós-graduação em Informática

Brasília, 16 de dezembro de 2019

Dedication

I dedicate this work to my wife, Juliana. Thank you for your unconditional support and for helping me through this path. This work is as yours as it is mine.

Acknowledgments

Thanks for my advisor, Dr Priscila América Solís Mendez Barreto, for whom I had the privilege of being oriented. Thanks to Dr Jacir Luiz Bordim and Dr Paulo Roberto de Lira Gondim for your reviews and considerations. Also, thanks to the teachers which helped on this project, especially Dr Marcos Fagundes Caetano and Dr Geraldo Pereira Rocha Filho. Thanks to the partners at the 5G-RANGE project for enabling this research. And finally, my big and sincere thank you for two fellow students, Rafael Amaral Soares and Gabriel Ferreira, for your help with the synthetic workload and the insights of how to develop the experiments, respectively.

This work was carried out with the support of the Higher Education Personnel Improvement Coordination - Brazil (CAPES), through the Access to the Portal de Periódicos.

Resumo

Virtualização de funções de redes é uma tecnologia promissora para reduzir custos, melhorar a configurabilidade e dinimizar infraestruturas de rede. A tecnologia facilita a criação e gerenciamento de redes para casos de uso das novas gerações de redes móveis, motivo que a fez ser integralmente adicionada à nova arquitetura do 5G. Porém, o uso de virtualização traz para as redes alguns problemas relacionados àquela tecnologia. Os maiores desafios têm como base a competição por recursos computacionais, gerando impedimentos de relocação, compartilhamento de estruturam, camadas adicionais de encaminhamento de pacotes e, principalmente, degradação de máquinas virtuais, o que pode diminuir consideravelmente o desempenho das redes e impactar o cumprimento de acordos feitos com os usuários. A nova geração de redes móveis também terá de lidar com esses problemas e, por isso, este trabalho propõe o uso de funções novas de rede integradas à arquitetura padrão do 5G que avaliam a performance dos recursos utilizados para virtualização de funções de rede. O trabalho apresentado se concentra na avaliação da eficiência do tráfego de rede adicional criado pela infraestrutura. Especificamente, aplica-se uma arquitetura distribuída de funções de observação que monitoram switches virtuais e reportam os dados obtidos para uma função central de avaliação. Para assegurar a viabilidade da arquitetura proposta, testes foram realizados utilizando uma infraestrutura virtualizada que segue os padrões propostos por institutos e empresas, tudo com o objetivo de garantir a viabilidade da arquitetura proposta em diferentes casos de uso. Os resultados mostram que é possível monitorar o ambiente sem causar interferências em transmissões realizadas entre as funções monitoradas. Essas descobertas também são cruciais para novos esforços na virtualização de componentes de rede, que tem sido um assunto muito discutido em redes móveis.

Palavras-chave: 5G, virtualização de funções de rede, arquitetura baseada em serviços

Resumo Expandido

Título: Um Modelo de Avaliação de Desempenho para Virtualização de Funções de Rede em 5G

A quinta geração de redes móveis (5G) integra o uso de telecomunicações e sistemas de computador. Com os últimos avanços nessa área, a virtualização de componentes de rede (NFV) está incluída como um dos principais facilitadores para o 5G, desenvolvendo seu núcleo em torno da ideia de funções virtualizadas que fornecem serviços através de uma plataforma unificada. Sua adaptabilidade e abstração de hardware motivam o uso de funções virtualizadas para implantação da infraestrutura de rede, mas também porque seu uso permite reduzir custos para criação e manutenção de ambientes de rede. No entanto, alcançar ambientes virtualizados completos ainda é uma tarefa difícil, que pode ser possível usando os padrões criados pelo setor de redes e institutos de pesquisa.

O 5G usa virtualização para implantar um grupo de novas funcionalidades exigidas por muitos casos de uso da indústria e da sociedade. Para atender a esses vastos requisitos, ele visa criar um ambiente muito flexível para agregar uma infinidade de tecnologias e integrar diferentes dispositivos em uma única infraestrutura de rede móvel. A versatilidade NFV permite que o 5G crie esse ambiente, mas também traz problemas de implementação relacionados à virtualização, como competição de recursos, posicionamento de máquinas, degradação de desempenho e outros dilemas que exigem monitoramento constante dos recursos disponíveis para evitar a criação de gargalos nessas redes com alto consumo de dados. Várias pesquisas usando sistemas de monitoramento distribuído para redes virtualizadas discutiram problemas relacionados, mas nenhum tentou integrar um modelo de nativo para a arquitetura 5G e ao mesmo tempo verificar as informações adicionais transferido para permitir a implementação desse ambiente. Dado isso, enquanto as redes móveis já estão tentando implementar uma virtualização completa dos recursos de hardware, o núcleo 5G não possui um modelo de avaliação de desempenho integrado para seus ambientes implementados, especialmente na geração de tráfego adicional por parte da infraestrutura da rede virtualizada.

Este trabalho estuda a viabilidade e implementa um modelo de avaliação de desempenho distribuído integrado ao padrão de ambiente virtualizado 5G, concentrando-se em

avaliar a eficiência de infraestruturas virtualizadas e verificar quanta informação adicional é produzida para lidar com diferentes casos de uso em 5G. As arquiteturas virtualizadas usam componentes conhecidos como nós de computação para implantar máquinas virtuais e executar o trabalho computacional, enquanto um controlador centralizado atua como gerente de toda a infraestrutura, definindo o posicionamento das funções implantadas. Este trabalho implementa um modelo de avaliação de desempenho distribuído com dois modos de monitoramento. O primeiro modo usa uma função externa para analisar as métricas de rede implementadas nos nós de computação, enquanto o segundo integra a função externa proposta ao modelo 5G e mapeia a implementação estrutural do ambiente de rede para fornecer uma melhor visualização. Ambas as funções usam comutadores virtuais para monitorar o tráfego de toda a infraestrutura e o segundo modo também suporta as avaliações fornecidas pelo primeiro. O estudo propõe dois modos diferentes porque: (i) permite uma maior flexibilidade de avaliação para o provedor de infraestrutura e (ii) o primeiro modo é viável, mesmo em arquiteturas completamente desconhecidas, tornando possível uma avaliação em infra-estruturas sub-localizadas. Neste trabalho, a escolha de uma avaliação distribuída é motivada pelo fato de os nós de computação já sofrerem com escassez de recursos e concorrência, ou seja, uma função de avaliação local pode impor alta carga computacional e piorar o cenário para recursos virtuais.

Este trabalho propõe um modelo de avaliação de desempenho integrado à arquitetura do 5G para monitorar e analisar o comportamento desse ambiente virtualizado, com o objetivo de antecipar e contornar gargalos. O modelo de avaliação adiciona duas novas funções à arquitetura do 5G, uma Função de Observação distribuída (ObF) e uma Função de Avaliação de Desempenho centralizada (PEvF). ObF coleta dados de nós de computação, monitorando seus ambientes virtualizados e relatando-os para a outra função. PEvF recebe os dados coletados por todos os ObF se compila para uma avaliação completa dos recursos virtualizados, fatias e também das funções virtualizadas. Enquanto alguns trabalhos tentaram criar um sistema de monitoramento de desempenho para virtualização de funções de rede, nenhum implementou uma integração nativa com a arquitetura do 5G para avaliar o desempenho de redes móveis virtualizadas, com foco no tráfego adicional criado por um servidor de infraestrutura virtualizada.

Como forma de demonstrar sua viabilidade em grandes ambientes, os testes focaram na eficiência dos ambientes NFV. Nos testes, os casos de uso em avaliação são categorizados em três grupos, Banda Larga Móvel Melhorada (eMBB), Comunicação ultra-confiável e de baixa latência (URLLC) e Comunicação Massiva de Tipo de Máquina (mMTC). eMBB requer taxas de download moderadas a altas em conexões estáveis para usuários finais. Um teste simples de acesso baseado na Web com taxa de transferência média foi utilizado para atender a esse requisito, analisando se o aumento da taxa de transferência de dados

do usuário também aumenta os dados da infraestrutura NFV. URLLC exige baixa latência e alta disponibilidade, com foco em cenários de computação de ponta. A conversa por voz requer um atraso máximo de 150 ms para realizar uma chamada aceitável, 50 ms para fornecê-lo com boa qualidade e 10 ms para streaming de áudio de alta qualidade. Portanto, um teste baseado em VoIP com infraestrutura de boa qualidade sobre NFV foi utilizando para fornecer uma análise justa para esses casos. mMTC requer alta densidade de tráfego e boa interconexão e pode ser caracterizada por testar conexões de dispositivos da internet das coisas de forma temporizada em uma janela para identificar lacunas na transmissão e seu impacto no sistema geral, os testes deste trabalho para mMTC utilizaram conexões temporizadas com um gerador centralizado. Os testes integram três fatias diferentes nesses diferentes casos de uso. Para cada caso, cargas sintéticas foram geradas utilizando modelos bem estruturados de geração de tráfego para cada caso de uso.

Os testes e resultados apresentados neste trabalho mostraram inicialmente que a arquitetura de avaliação proposta tem a capacidade de monitorar e analisar com êxito o desempenho de um ambiente NFV utilizando a configuração padrão para implementações em 5G, com uma combinação do Open Source MANO e do Openstack. As avaliações focaram em três casos de uso e a quantidade de tráfego adicional criado para cada cenário.

Para o cenário de URLLC com VoIP, durante uma chamada média de 180 segundos, os dados reais corresponderam a cerca de 680,73 KB, enquanto a infraestrutura produziu 313.42 KB, o que equivale a 31,52% do total de informações transmitidas no ambiente. Essa situação mostra uma grande quantidade de dados adicionais produzidos para realizar uma chamada VoIP. Em cenários do mundo real, cerca de 30% do tráfego adicional gerado comprometeria seriamente qualquer utilidade de NFV. Durante as verificações de integridade, o tráfego de infraestrutura aumenta drasticamente. Uma análise de crescimento de tráfego mostrou que as verificações de integridade duram um período médio de 3 segundos mas que aumentam a quantidade de dados transferidos pela infraestrutura em mais de 600% ao se considerar a janela de 3 segundos anterior a ela. Ao separar a comunicação entre períodos normais e períodos de verificação de integridade, a infraestrutura corresponde a uma média de 14,90% do tráfego total no primeiro caso e 80,81% no posterior. Nesses períodos, a taxa de transferência do ambiente virtualizado para os dados reais mantém uma taxa estável; esse fato remove a possibilidade de uma correlação entre o aumento dos dados de infraestrutura e os dados do usuário. Para o cenário de teste analisado em URLLC, a transmissão de dados de verificação de integridade ocorre apenas 5% do tempo total e, mesmo com janelas tão pequenas, aumentou consideravelmente a quantidade total de informações transferidas. Para testes futuros e para qualquer implantação real com ambientes virtualizados, se a quantidade de dados adicionais interferir na comunicação de um cenário diferente, a janela de espera entre as verificações de integridade do Openstack

poderá ser aumentada para evitar esse alto tráfego adicional gerado pela infraestrutura. Neste caso em particular (URLLC) foram detectados alguns desvios na taxa de transferência de dados, os quais mostraram indícios de que o aumento no atraso em NFV pode ser considerativo.

Para o cenário de eMBB com vídeo sintético, durante um fluxo de vídeo médio de 150 segundos, os dados reais correspondiam a cerca de 74,46 MB, enquanto a infraestrutura produzia 1,77 MB, o que equivale a apenas 2,32% do total de informações transmitidas no ambiente. Para um caso específico utilizado como exemplo, os dados do fluxo de vídeo corresponderam a 75,63 MB e a infraestrutura produziu apenas 2,11 MB, ou seja, 2,72% do total de dados. Este valor representa uma redução significativa quando comparado com o teste anterior para um VoIP em URLLC. Durante as verificações de integridade, o tráfego da infraestrutura ainda aumenta drasticamente. Ao separar a comunicação entre períodos normais e períodos de verificação de integridade, a infraestrutura corresponde a uma média de 1,01% do tráfego total no primeiro caso e 17,96% no segundo. Nesses períodos, a taxa de transferência do ambiente virtualizado para os dados reais ainda se mantém estável; o que coloca indícios mais fortes na análise feita sobre uma correlação inexistente entre os dados crescentes da infraestrutura e os dados do usuário. Para o cenário de teste analisado, a transmissão de dados de verificação de integridade ocorre apenas 5% do tempo total. Para testes futuros e para qualquer implantação real com ambientes de fluxo de vídeo virtualizados, se a quantidade de dados adicionais interferir na comunicação de um cenário diferente, a de janela de espera entre as verificações de integridade do Openstack poderá ser aumentada para evitar esse alto tráfego adicional gerado pela infraestrutura, conforme sugerido para a análise anterior.

Na amostra específica para mMTC, a transmissão segue um ritmo constante com pequenas variações na taxa de transferência. Essa estabilidade ocorre devido às características temporizadas desses dispositivos. Para uma transferência de 135 segundos utilizada como exemplo, teve-se uma taxa de transferência média de 10,07 Mbps. A amostra gerou um total de 683899 pacotes nessa transmissão única, com apenas 6599 pacotes descartados ou 0,965%, gerando uma média de 5069,92 pacotes por segundo, que produziu 169985155 bytes de acordo com a detecção do ObF. Durante uma execução média de 135 segundos, os dados reais corresponderam a cerca de 169,98 MB, enquanto a infraestrutura produziu 2,69 MB, o que equivale a apenas 1,55% do total de informações transmitidas no ambiente. Quando comparado com testes anteriores, esse valor representa outra redução considerável na proporção de dados de infraestrutura para o total de dados.

É possível ver, com os resultados coletados, que o uso de funções distribuídas para coletar dados sobre o estado dos nós de computação e enviados para uma entidade centralizada é um método viável para evitar impor muita tensão, mesmo em infraestruturas

com restrição de recursos. As informações analisadas também mostraram que NFV é uma tecnologia viável para permitir um ambiente de rede móvel virtualizado. Para trabalhos futuros, os desvios na taxa de transferência de dados no caso de uso URLLC e o comportamento da quantidade de tráfego adicional em infraestruturas mais extensas são tópicos abertos interessantes para futuras pesquisas.

A ObF se liga aos comutadores virtuais criados pela NFV para realizar o monitoramento ativo e agregar informações a partir do nível de microsserviço até o nível de fatia. O foco neste trabalho foi a confirmação das capacidades de monitoramento do design pretendido. Como mostrado pelos resultados dos testes; essa abordagem resultou em um sistema de monitoramento de desempenho bem-sucedido na infraestrutura distribuída e gerou informações adequadas sobre o comportamento do ambiente virtualizado. Após toda a análise de teste, essa abordagem inovadora resultou em uma implementação de coleta de dados muito eficiente para NFV. Os resultados obtidos motivam o uso das técnicas propostas como parte integrante do arquitetura do 5G e o tornam uma plataforma atraente e promissora para arquitetura para infraestruturas orientadas a recursos. Até onde sabemos, este foi o primeiro trabalho a avaliar sistematicamente a introdução de tais funções de avaliação de desempenho no núcleo das redes móveis virtualizadas enquanto se concentra na análise do tráfego da rede de infraestrutura.

Em seu estado atual, as funções de avaliação podem funcionar como componentes ativos de arquitetura do 5G e são capazes de mostrar como ela se comporta em diferentes cenários. Para avaliar melhor o tráfego de rede adicional criado pela infraestrutura, as funções desenvolvidas não consideraram outros componentes no momento, como recursos de computação e memória. Com os resultados obtidos, um ambiente de virtualização de funções de rede (NFV) parece ser um candidato viável para implantação de rede móvel. A porcentagem de tráfego adicional criado pela infraestrutura tende a diminuir ao aumentar a quantidade de dados reais nos cenários de teste propostos. No momento, os testes consideraram apenas um pequeno número de nós de computação e máquinas virtuais, o que poderia oferecer um desempenho decente para cenários virtualizados. Para experiências futuras, o uso de uma quantidade mais significativa de nós de computação e máquinas virtuais pode mostrar uma visão melhor de como a infraestrutura cria tráfego adicional. Além disso, o uso do OpenStack Nova exige uma avaliação de seus componentes para verificar o tempo e os recursos de rede necessários em cada elemento.

A implementação dessa avaliação de desempenho considerou um ambiente NFV usando o software padrão de fato para virtualização de rede móvel, OpenStack e Open Source MANO, como a base da infraestrutura implantada. Essa configuração atende aos requisitos de outros novos projetos 5G e sua adoção permite consistência com trabalhos anteriores, permitindo a integração desse trabalho nos projetos atuais. Vale ressaltar que,

embora os resultados obtidos tenham dado uma uma ótima perspectiva de como lidar com as melhorias propostas pelo 5G, trabalhos futuros devem seguir estritamente as mesmas diretrizes para conseguir operar com serviços funcionais. E conforme mencionado, para expandir ainda mais a avaliação, a análise deve evoluir para considerar outros ambientes.

Palavras-chave: 5G, virtualização de funções de rede, arquitetura baseada em serviços

Abstract

Network Function Virtualisation (NFV) is a promising technology aiming to reduce costs, improve configurability and streamline network infrastructures. The technology facilitates the creation and management of networks for use cases of new generations of mobile networks. Because of this, 5G architecture uses this technology as one of its main enablers. However, the use of virtualisation brings to computer networks some problems related to that technology. The biggest challenges emerge from computational resources competition, causing problems related to resource rellocation, structure sharing, additional packet forwarding layers, and virtual machine degradation. These problems can considerably slow down network performance, and impact network compliance of service quality agreed with users. Therefore, the new generation of mobile networks will also have to deal with these issues. This work proposes the use of new network functions integrated within 5G architecture to evaluate the performance of resources used for virtualisation of network functions. It focuses on assessing the efficiency of additional network traffic created by the infrastructure. Specifically, it applies a distributed architecture of observation functions that monitor virtual switches and report the data obtained to a central evaluation function. Tests were performed using a virtualised infrastructure that follows the standards proposed by institutes and companies, aiming to ensure the viability of the proposed architecture under different use cases. The results show that it is possible to monitor the environment without interfering with transmissions between the monitored functions and that Network Function Virtualisation (NFV). These findings are also crucial to new efforts in the virtualisation of network components, which have been a much-discussed matter on mobile networks.

Keywords: 5G, network function virtualisation, service-based architecture

Contents

List of Figures

List of Tables

Acronyms

2G Second Generation of Mobile Communication.

3G Third Generation of Mobile Communication.

3GPP 3rd Generation Partnership Project.

4G Fourth Generation of Mobile Communication.

5G Fifth Generation of Mobile Communication.

5GC 5G Core Network.

5GPPP 5G Public Private Partnership.

AMF Access and Mobility Management function.

AN Access Node.

AP Access Point.

API Application Programming Interface.

AUSF Authentication Server Function.

C-Plane control plane.

CapEx Capital expenditure.

CN Core Network.

CSD Circuit Switched Data.

CUs Central Units.

DASH Dynamic Adaptive Streaming over HTTP.

DN Data Network.

DNS Domain Name System.

DUs Distributed Units.

E-UTRA Evolved UMTS Terrestrial Radio Access.

E-UTRAN E-UTRA Network.

EDGE Enhanced Data rates for GSM Evolution.

eMBB Enhanced Mobile Broadband.

eNodeB Evolved Node B.

EPC Evolved Packet Core.

EPS Evolved Packet System.

ETSI European Telecommunications Standards Institute.

FDMA Frequency Division Multiple Access.

GGSN Gateway GPRS Support Node.

gNB Next Generation NodeB.

GPRS General Packet Radio Service.

GSM Global System for Mobile Communications.

HSPA High Speed Packet Access.

HSS Home Subscriber Server.

HTTP Hypertext Transfer Protocol.

IaaS Infrastructure as a Service.

IMS IP Multimedia Subsystem.

InP Infrastructure Provider.

IoT Internet of Things.

IP Internet Protocol.

ISP Internet Service Provider.

LTE Long-Term Evolution.

MANO Management and Orchestration.

MME Mobility Management Entity.

mMTC Massive Machine Type Communication.

MNO Mobile Network Operator.

MTC Machine Type Communication.

NaaS Network as a Service.

NEF Network Exposure Function.

NF Network Function.

NFV Network Function Virtualisation.

NFVI NFV Infrastructure.

NFVO NFV Orchestrator.

NRF Network Repository Function.

NS Network Service.

NSSF Network Slice Selection Function.

ObF Observation Function.

OpEx Operational expenditure.

OSM Open Source MANO.

P-GW Packet Data Network (PDN) Gateway.

PCRF Policy and Charging Rules Function.

PDCP Packet Data Convergence Protocol.

PDN Packet Data Network.

PDU Protocol Data Unit.

PEvF Performance Evaluation Function.

QoS Quality of Service.

RAN Radio Access Network.

REST Representational State Transfer.

RLC Radio Link Control.

RPi Raspberry Pi.

RTP Real-time Transport Protocol.

S-GW Serving Gateway.

SBA Service-Based Architecture.

SBI Service-Based Interface.

SDN Software-Defined Networking.

SIP Session Initiation Protocol.

SLA Service Level Agreement.

SMF Session Management Function.

SOA Service-Oriented Architecture.

SP Service Provider.

TDMA Time Division Multiple Access.

U-Plane user plane.

UDM Unified Data Management.

UDR Unified Data Repository.

UE User Equipment.

UMTS Universal Mobile Telecommunication System.

UPF User Plane Function.

URLLC Ultra-Reliable and Low Latency Communications.

vCPU Virtual CPU.

VIM Virtualised Infrastructure Manager.

VM Virtual Machine.

VNF Virtualised Network Function.

VNFD VNF Descriptor.

VNFM VNF Manager.

VoIP Voice Over IP.

W-CDMA Wideband Code Division Multiple Access.

WAP Wireless Application Protocol.

Chapter 1

Introduction

The Fifth Generation of Mobile Communication (5G) integrates the use of telecommunication and computer systems. With the latest advancements in this area, the 5G Public Private Partnership (5GPPP) included the virtualisation of network components as one of the main enablers for the 5G [1], developing its core around the idea of virtualised functions providing services through an unified platform. Their adaptability and underlying hardware abstraction motivates the use of virtualised functions for network infrastructure deployment, but also because its use allows reducing costs for creation and maintenance of network environments [2]. However, achieving full virtualised environments is still a difficult task, which can be possible by using standards created by the network industry and research institutes.

5G uses virtualisation to deploy a group of new functionalities demanded by many industrial and social use cases. In order to attend to such vast requirements, it aims to create a very flexible environment to aggregate a plethora of technologies and to integrate different devices into a single mobile network infrastructure. The versatility of network virtualisation allows 5G to create such an environment but also brings to the implementation problems related to virtualisation, such as resource competition, machine placement, performance degradation [3] and other quandaries that require constant monitoring of available resources to avoid creating bottlenecks on these high data-intensive networks. Several research works using distributed monitoring systems for virtualised networks have discussed related problems [4, 5, 6, 7], but none tried to integrate a native evaluation evaluation model for 5G architecture and while also verifying the additional information transferred to enable this environment. Given this, while mobile networks are already trying to implement a full virtualisation of hardware resources, the 5G core itself lacks an integrated performance evaluation model for its deployed environments, especially on analysing function degradation.

The goal of this work is to study the viability of a distributed performance evaluation model integrated into the 5G virtualised environment standard, focusing on evaluating the efficiency of virtualised deployments by verifying how much additional information is produced to deal with different 5G use cases. Virtualised architectures use components known as compute nodes to deploy Virtual Machine (VM)s and execute computational work, while a centralised controller acts as a manager of the whole infrastructure, defining the placement of deployed functions. The approach is to analyse the viability of a distributed performance evaluation model with two monitoring modes. The first mode uses an external function to analyse network metrics of functions deployed on compute nodes, while the second integrates the proposed external function into the 5G model and maps structural implementation of the network environment to provide a better visualisation. Both functions use virtual switches to monitor traffic from the whole infrastructure and the second mode also supports the evaluations provided by the first. The study proposes two different modes because: (i) it allows a higher evaluation flexibility for the infrastructure provider and (ii) the first mode is viable even on completely unknown architectures, so it makes feasible an evaluation on sub located infrastructures. In this work, the choice of a distributed evaluation is motivated by the fact that compute nodes already suffer from resource shortage and competition, that is, a local evaluation function could impose high computation load and worsen the scenario for virtual resources.

1.1 Problem

Virtualised environments suffer from many problems such as degradation, resource competition, resource-constrained deployments and VM relocation. Mobile networks impose even more requirements for these arrangements. While high flexibility and low cost of deployment and maintenance are the primary motivations for using virtualisation on network functions, the performance factor and its evaluation are still research challenges. In this work, the focus is to propose a performance evaluation model for a virtualised environment in 5G networks, focusing on analysing the efficiency of generated traffic of the deployment.

1.2 Contributions

This work proposes a new distributed evaluation model composed by a centralised evaluation function and an assortment of distributed observation functions acting as monitors for the compute nodes in a virtualised infrastructure. The model, unlike previous attempts to evaluate virtualized networks, is designed to be integrated into 5G as a native

service. Through a comparative analysis of many monitoring scenarios in virtualised environments and by exploring virtual components, this work also studies the compromises of evaluating a virtualised infrastructure without impacting the infrastructure itself. Considering the ongoing specification and reserch on the 5G architecture, this work designs a new proposal of a performance evaluation model that can be integrated into the 5G architecture.

1.3 Text Organisation

This document has five chapters: chapter 2 starts with a storyline of mobile networks evolution; then it presents a literature review and the theoretical background about general aspects related to 5G networks, network slicing, softwarisation of the control plane, the use of virtualised resources for network function deployment and the problems related to it. Chapter 3 describes the proposal of the performance evaluation model based on two new components for the 5G architecture, one for monitoring and the other for evaluation. Then, the chapter presents the operation of both functions and how to integrate them into the new generation of mobile networks. Chapter 4 presents the implementation of a 5G virtualised environment following de-facto standard software, following by a definition of use cases required by 5G and the workload used to evaluate each scenario. Then it shows the gathered results and discusses the collected performance metrics. Finally, Chapter 5 debates about the outcomes and the differences from expected results, and then it presents proposed next steps and suggests future directions for this work.

Chapter 2

Literature Review and Theoretical Background

For a better understanding of the concepts and the research proposal to be discussed in subsequent chapters, this chapter presents the theoretical background and latest research advancements that support this work. First, the chapter presents a revision of mobile networks' historical background, describing major changes between each iteration. Then, a discussion of the main topics involving the 5G, such as paradigm shifts, architectural concepts, and notations, is provided. Next, topics related to 5G NFV Architecture and performance evaluation are introduced. Finally, related and recent research work are presented.

2.1 The Fifth Generation of Mobile Communication

This section gives a review of mobile communication history, quickly reporting details about significant changes between generations. Moreover, it offers a concise explanation of 5G architecture, describing enabling technologies and presenting challenges of proposed implementations, mainly based on virtualisation and softwarisation techniques.

2.1.1 Pre-5G Mobile Networks

There have been four iterations of mobile networks technologies so far. In the 1980s, the first generation emerged with a plethora of different telecommunications standards deploying voice conversation over analogue frequency modulation (FM). This first iteration of mobile communication had distinct technology implementations for each country, limiting adoption and hindering integration [8]. For this reason, the European Telecommunica-

tions Standards Institute (ETSI) focused on standardising a new mobile communication system for the next generation.

In the early 1990s, with the Second Generation of Mobile Communication (2G), the mobile conversation evolved to use a set of digital modulation patterns with the unified Global System for Mobile Communications (GSM) standard developed by ETSI. Early GSM used Circuit Switched Data (CSD), digitally encoding voice before forwarding and applying a combination of and Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA) to, respectively, subdivide available bandwidth between carriers and alternate the access time of carrier frequencies. It also conceived a standardised Core Network (CN) architecture, with fundamental components to provide interconnection, subscription management and other functions [9]. This generation subsequently included other standards. The General Packet Radio Service (GPRS) standard introduced packet-switched communication and added the Gateway GPRS Support Node (GGSN), that is a new CN component responsible for providing external mobile network access [10] using the Wireless Application Protocol (WAP) [11]. The last 2G standard, known as Enhanced Data rates for GSM Evolution (EDGE), focused on radio-access changes and further improved applied digital encoding techniques, increasing the mobile access rates up to threefold [12]. EDGE applies an improvement on modulation and coding scheme and can be built on top of GSM/GPRS.

Aiming to create a mobile Internet Protocol (IP) communication system, the 3rd Generation Partnership Project (3GPP) consortium developed a new mobile communication technology based on 2G GSM. With the Third Generation of Mobile Communication (3G), arrived the first high-speed internet access on mobile networks, the newly-introduced and GSM-based Universal Mobile Telecommunication System (UMTS) kept both digital modulation and packet-switching on the CN and improved the use of radio resources with a new air interface called Wideband Code Division Multiple Access (W-CDMA) [13]. UMTS inherited almost all aspects of 2G access network and only changed a few CN components from previous mobile standards. However, the second 3G standard, namely High Speed Packet Access (HSPA), changed some Radio Access Network (RAN) functions, although still keeping the same CN, and introduced IP multimedia service through the IP Multimedia Subsystem (IMS) [14, 15].

The Fourth Generation of Mobile Communication (4G) introduced Long-Term Evolution (LTE) as RAN. Architecture-wise, 4G provided more changes than previous generations. To allow independent improvements, LTE applies a functional split between the user plane (U-Plane) and the control plane (C-Plane), responsible for transferring user data and the data signalling, respectively. This generation employs a core network architecture called Evolved Packet Core (EPC), an evolution of the packet-switched ar-

chitecture used in GPRS/UMTS that thoroughly removes circuit-switched domains and employs IP as the foundation of all communication.

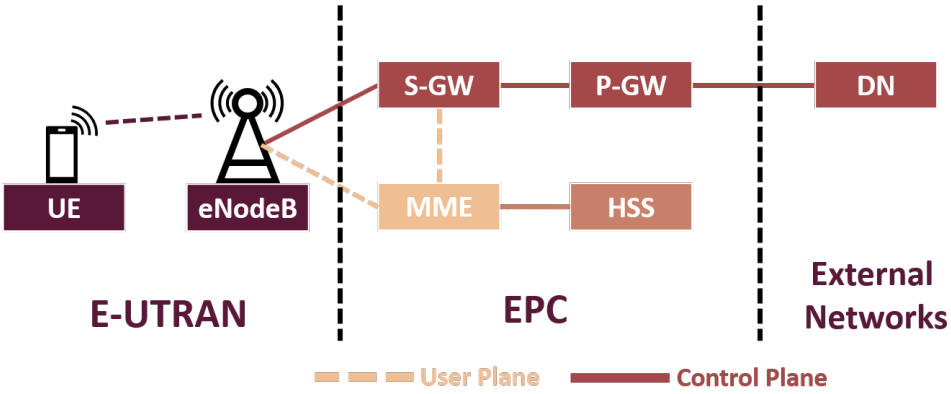

Figure 2.1: Basic EPS Architecture with E-UTRAN access.

Figure 2.1 shows the EPC on a basic LTE EPS reference point. On the left is the User Equipment (UE) connected on a base station, also called Evolved Node B (eNodeB) or eNB, through the LTE air interface, the Evolved UMTS Terrestrial Radio Access (E-UTRA), thus composing the E-UTRA Network (E-UTRAN). In the middle is the EPC and it has four main components: the Serving Gateway (S-GW), the Packet Data Network (PDN) Gateway (P-GW), the Mobility Management Entity (MME) and the Home Subscriber Server (HSS). The S-GW is responsible for routing and forwarding user data, for U-Plane communication between eNodeBs and for mobility management between legacy 2G/3G and LTE. The P-GW provides connectivity to the UE. The MME performs C-Plane access for the LTE network. The HSS is a database with user subscription data.

2.1.2 5G Core Network and Architectural Improvements

The Fifth Generation of Mobile Communication (5G) aims to improve significantly on previous generations, and because of that, it contemplates even more architectural changes than LTE. 3GPP created the 5GPPP to consolidate an overall architectural vision and respond to a wide range of use cases, categorised under Enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low Latency Communications (URLLC) and mMTC [16]. eMBB use cases require high download rates, even greater connectivity density compared to LTE, improvements on UE mobility and increased coverage. URLLC demands a combination of low latency and high availability to achieve roughly 99.9% of reliability and improve edge computing scenarios. mMTC calls for high traffic density and interconnection of a plethora of different technologies. Figure 2.2 shows how these scenarios correlate.

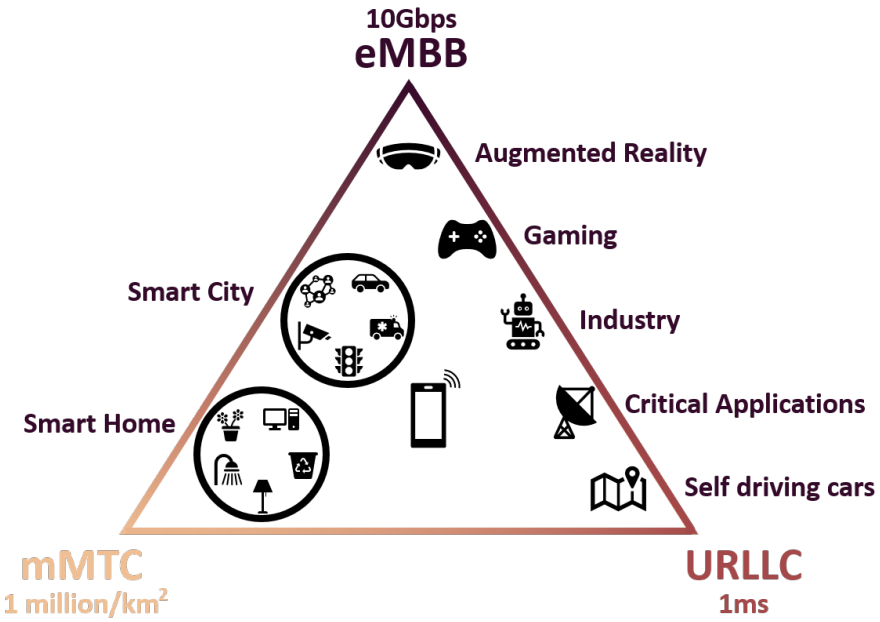

Figure 2.2: 5G Use Cases.

These cases require a versatile and adaptable system, so 5GPPP further improved LTE EPC to be highly versatile, introducing Software-Defined Networking (SDN) [17], NFV and Network Slicing to increase the network flexibility using a Service-Based Architecture (SBA). Softwarisation and virtualisation allow network programmability in 5G and also in many of its components by re-designing the network configuration based on the service requested. SDN and NFV are complementary technologies that help 5G to reach its goals, and Network Slicing uses both to deploy the desired 5G flexibility.

5G Service-Based Architecture

5G employs a core network component called 5G Core Network (5GC), with significant architectural changes from LTE EPC. The new core allows the use of traditional reference point applied by the EPC, but introduces a new Service-Based Architecture (SBA) as an alternative communication system between core Network Function (NF)s on 5G. With a SBA, functions expose their outputs as microservices or using a Representational State Transfer (REST) Application Programming Interface (API), and other functions can consume these services with a request-respond or a subscribe-notify scheme depending on the deployed model [18]. 5GC takes advantage of many EPC concepts but with several modifications on NFs, partitioning some functions between different components and further specialising them for C-Plane and U-Plane control.

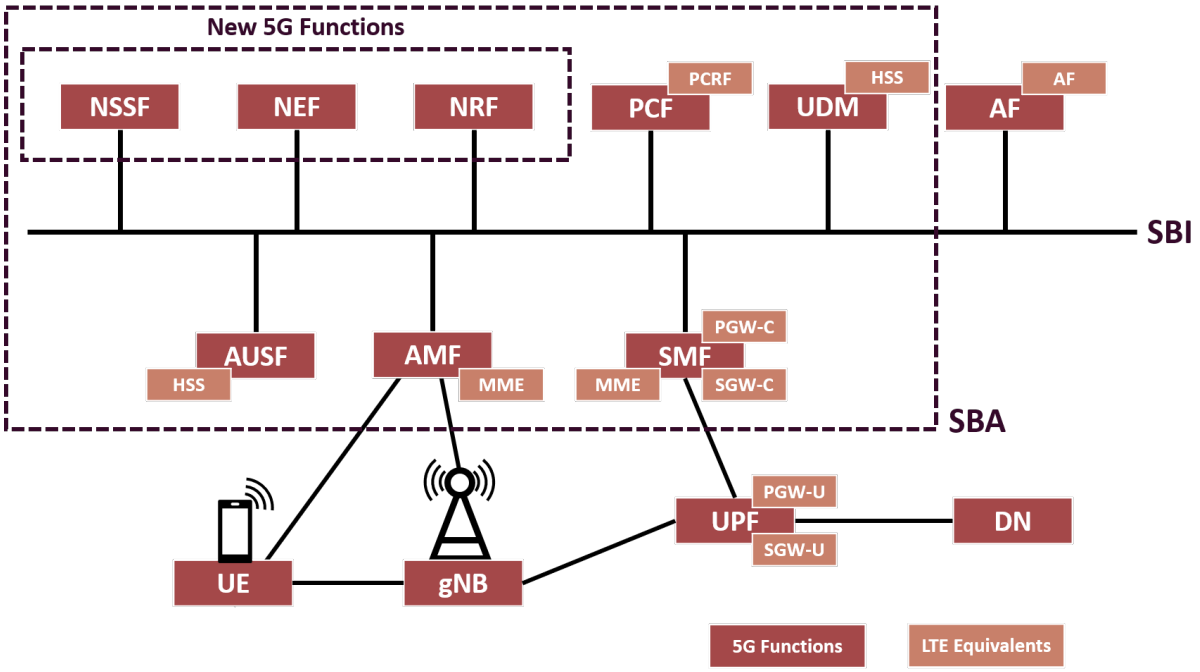

Figure 2.3: 5GC SBA Network Function and LTE equivalents.

5GC is a group of modularised NFs providing and consuming services between themselves. Figure 2.3 depicts NFs of 5GC SBA and their LTE counterparts, also showing the Service-Based Interface (SBI) used by them to communicate. Access and Mobility Management function (AMF) is responsible for conducting numerous operations and dealing with paramount tasks, including the management of all mobility assignments and supervision of access control. AMF interacts with many other NFs to perform cooperative chores, such as authentication (AUSF), subscription control UDM, slicing orchestration (NSSF) and PDU handling. (SMF). Session Management Function (SMF) take over LTE C-Plane duties from S-GW and P-GW, but also manages some MME functions. This NF deals mostly with session governance (management, establishment, modification and release). Other important feature include linking UPF with AN and general policy application with PCRF. User Plane Function (UPF) performs S-GW and P-GW U-Plane functions. It gathers U-Plane Quality of Service (QoS) and policies like gating, redirection and traffic steering, from PCRF and enforces them. It also deals with packet inspection, routing and forwarding and is the external Protocol Data Unit (PDU) Session point to any Data Network (DN). AMF, SMF and UPF do not require implementation of all of their functionalities, being possible to deploy only some functions in an instance of a U-Plane Network Slice. UE represents any equipment connected to the 5G network, while Next Generation NodeB (gNB) is the communication node, also called base station, which is responsible for providing the air interface connectivity.

Policy and Charging Rules Function (PCRF) major functionalities include supporting application execution priority, managing mobility-related guidelines, maintaining a policy framework used to govern all network behaviour and providing rules for QoS Control and Charging on routed traffic. It stores and retrieves all policy information using a central database called Unified Data Repository (UDR). The PCRF also defines session context by interoperating with AMF and SMF, respectively. Unified Data Management (UDM) inherits HSS functions of subscription data management. In case of a service migration scenario from LTE to 5G, it is possible to deploy a shared infrastructure using LTE HSS as UDM to provide a common base for both EPC and 5GC [19]. The UDM also stores its data using the UDR. Authentication Server Function (AUSF) is responsible for supporting authentication of both 3GPP access and untrusted non-3GPP access and informing the UDM of all subscriber authentication attempts [20].

There is also a group of 5GC-exclusive NFs, created to support the SBA scheme and the Network Slice environment. Network Exposure Function (NEF) acts on internal-external communication translation, storing and retrieving information on capabilities and events about other NFs to external agents but also translating information from non-3GPP agents to internal NFs. The Network Repository Function (NRF) contains information about all available services and provides this data as its service output, acting as a NF broker. Like the PCRF and UDM, the NRF uses the UDR for structured data storing. The Network Slice Selection Function (NSSF) supports Network Slice instance selection to provide services for UEs

Network Slicing

According to 5GPPP [1], Network Slicing is a promising future-proof framework and a fundamental concept for 5G that also satisfy demands from vertical sectors of lifecycle automation of services on mobile networks. Network Slicing is the allocation of available physical, virtual and emulated resources to deploy multiple simultaneous and isolated logical networks adjusted to particular cases [21]. Each of these end-to-end logical networks created is called a Network Slice Instance and has mainly, but not only, independent computing, storage and networking assets. A Network Slice is a composition of an infrastructure subset and interconnected Network Services Network Service (NS), which are groups of NFs acting together to deliver a common goal. It is possible to create, change, combine and terminate any slice in a flexible manner when required. Network Slicing aims to deliver high levels of network customisation and adaptable QoS by abstracting underlying hardware, softwarising functions and their management, and isolating any deployed environments. The process of selection and definition of which available resources, both physical and virtual, must integrate a slice is called orchestration.

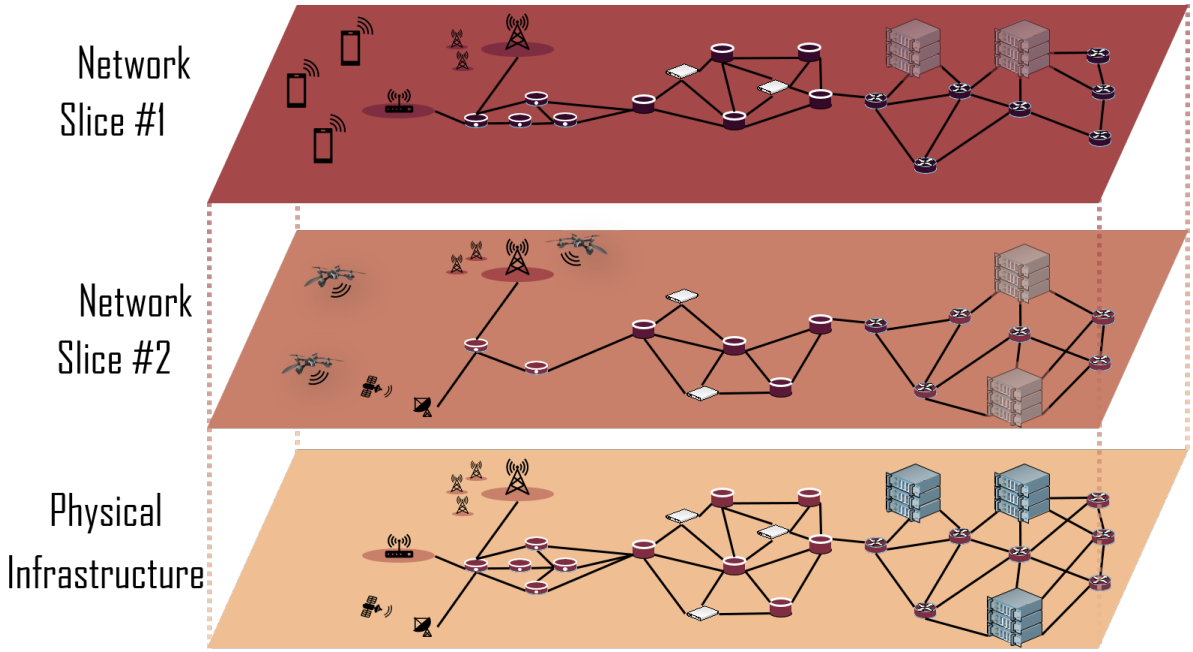

Figure 2.4: Network Slicing an Infrastructure for Two Different Slices.

Figure 2.4 shows an example of Network Slicing, with two different use cases deployed using portions of a shared infrastructure. Network Slicing is a feasible way to deploy a plethora of services from multiple actors of vertical industries, such as energy, automotive and health [22]. With the abstraction between resources and deployed services, it is also possible for agents to act only as Infrastructure Provider (InP) to multiple clients while, at the same time, clients (tenants) can combine resources from several InPs to perform a single horizontal deployment, this feature is called Multi-Tenancy and is also a fundamental concept for 5G [1, 23]. On 5G, Network Slicing allows the replication of 5GC SBA on a Multi-Tenancy scenario, with Internet Service Provider (ISP)s sharing or leasing multiple infrastructures to deploy various services. Two main enablers of network slicing are SDN and NFV. The first allows the softwarisation of the control plane, while the second provide virtualised infrastructure-independent resources.

Software-Defined Networking

SDN is not the main focus of this work, but it often couples with NFV [4], so it is necessary to make a clear division between these two technologies. Forwarding and routing are the two most crucial functions of the most complex layer in the protocol stack, known as the network layer. This layer is responsible for (1) encapsulating segments received from the transport layer into a datagram and sending it to another router and (2) reversing it on the destination by extracting the content of the received datagram and dispatching to the

transport layer. On traditional networks, this layer couples datagram forwarding and its control within the routers themselves. However, just like described for LTE, SDN splits these functions as the C-Plane and the U-Plane, making the network switches responsible only for data forwarding (the U-Plane) and allowing the management of network topology (the C-Plane) by a logically centralised controller. The SDN controller defines forwarding table values for all switches and sends these values to them using a well-defined programmed API, with OpenFlow [24] as the most prominent implementation. SDN achieves its softwarisation goals by applying a well-defined separation between policy definition, its enforcement throughout the switches and the data traffic forwarding, which allows programmability of the network services. For 5G, SDN can be directly applied on S-GW and P-GW functionalities to achieve Packet Data Network (PDN) softwarisation [25].

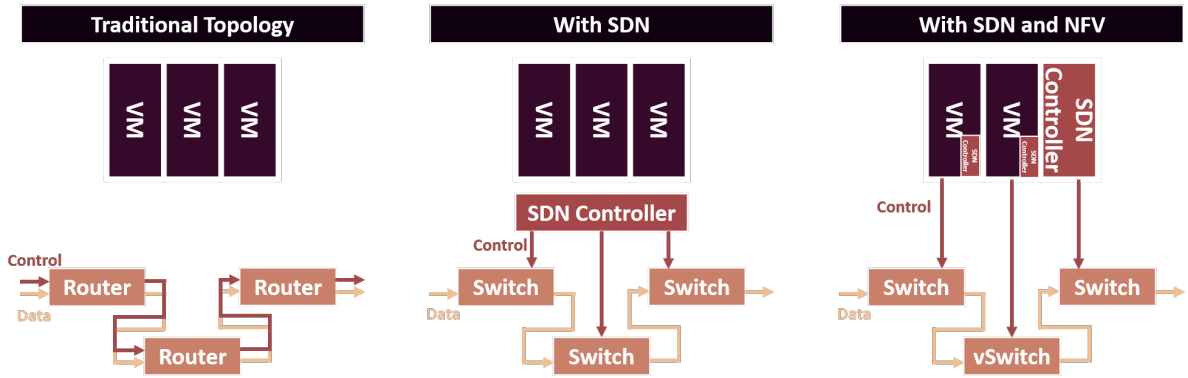

Figure 2.5: Coupling NFV and SDN.

Another emerging trend, NFV, seeks to transform networks using virtualised functions of physical components, thus abstracting available resources for application deployment. SDN and NFV approaches operate well together, as they are complementary solutions aiming to improve networking. Figure 2.5 shows an example of how SDN and NFV can be combined to achieve network softwarisation. The next section deeply analyses NFV architecture, implementation alternatives and the problems with current implementations.

2.2 Network Function Virtualisation

This section introduces basic concepts of NFV, describing the ETSI NFV architecture and presenting a component called Virtualised Network Function (VNF), which is a central element in NFV and whose QoS evaluation parameters, especially the service degradation and Service Level Agreement (SLA), are the focus of this work.

2.2.1 ETSI NFV Architecture

To deploy a network infrastructure, carriers use multiple hardware pieces typically defined as middleboxes or network appliances, such as gateways, firewall, transcoders and proxies. These specialised physical equipment have complex requirements, high maintenance cost, and each new network generation entails updates on several of these devices to offer new services. To satisfy SLAs, this recurring demand for infrastructure modernisation increases both the Capital expenditure (CapEx) and Operational expenditure (OpEx) for carriers on each technology iteration, and hardware failures demand surplus devices to achieve a fault-tolerant system. Furthermore, the number of middleboxes is roughly the same as the number of routers in a network [26].

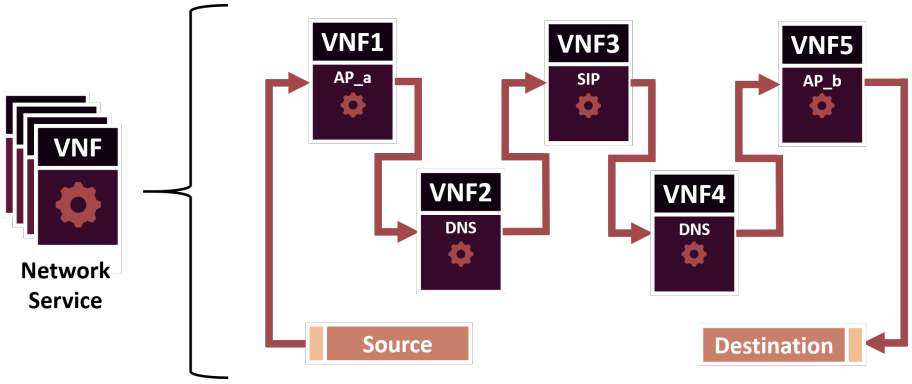

Figure 2.6: VNF composition of an NS.

To solve the issues described above and reduce the ever-increasing expenditures, the ETSI and a group of global leading Mobile Network Operator (MNO)s (or also Service Provider (SP)s) devised the NFV standard, relocating middlebox functions and their management to modularised software applications [27]. The standard implements a NF of a middlebox through a virtualised entity named VNF using SP own private cloud computing infrastructure. Each VNF implements a distinct function in an NFV Infrastructure (NFVI) and can be reproduced or migrated among servers on different instances. A group of VNFs compose a NS, Figure 2.6 shows an example of VNF chaining for NS composition. An appealing advantage of the architecture is that the same infrastructure used by an NFV can be used to support cloud computing applications simultaneously in a multi-tenancy scheme. The NFVI provides compute and storage following Infrastructure as a Service (IaaS) ideas of a Service-Oriented Architecture (SOA), but also provides network capabilities in a similar fashion of a Network as a Service (NaaS). Other advantages of NFV include enhanced flexibility of hardware usage; outstanding adaptation toward novel technologies; power-consumption reduction by deactivating idle equipment; and interface standardisation in favour of better interoperability.

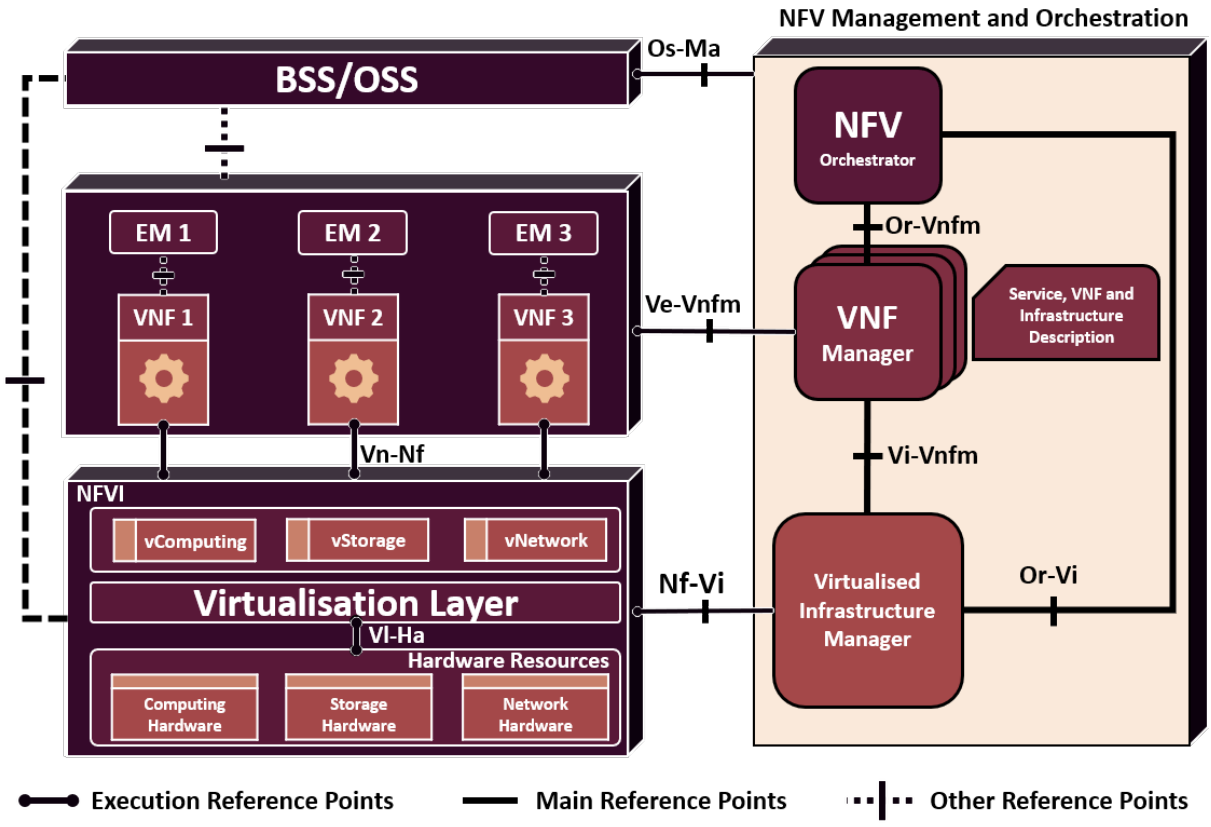

Figure 2.7: ETSI NFV Standard Architecture.

Figure 2.7 shows ETSI NFV Architecture [28] with the NFVI on the bottom left and the NFV Management and Orchestration (MANO) on the right. The proposed MANO domain encompasses the orchestration and the lifecycle management and of all physical and software resources, but also handles VNFs lifespan control, having three main constituents. The first comprises the NFV Orchestrator (NFVO) and is responsible for the management of software resources and service realisation on the NFVI. VNF lifecycle management resides on the second MANO component, the VNF Manager (VNFM). It is possible to use a multiplex of VNFMs, coupling them on a single NFV environment. The last building block, the Virtualised Infrastructure Manager (VIM), controls all interactions between the upper software stratum and the bottom hardware resources of computing, storage and network. It also conducts a plethora of operations, including resource visibility ruling, infrastructure performance analysis and other monitoring proceedings. Additionally, the MANO features a data collection with each VNF Descriptor (VNFD) (a VNF information template), a catalogue of NFVI resources, registers about VNF instances and the arrangement of all deployed NSs.

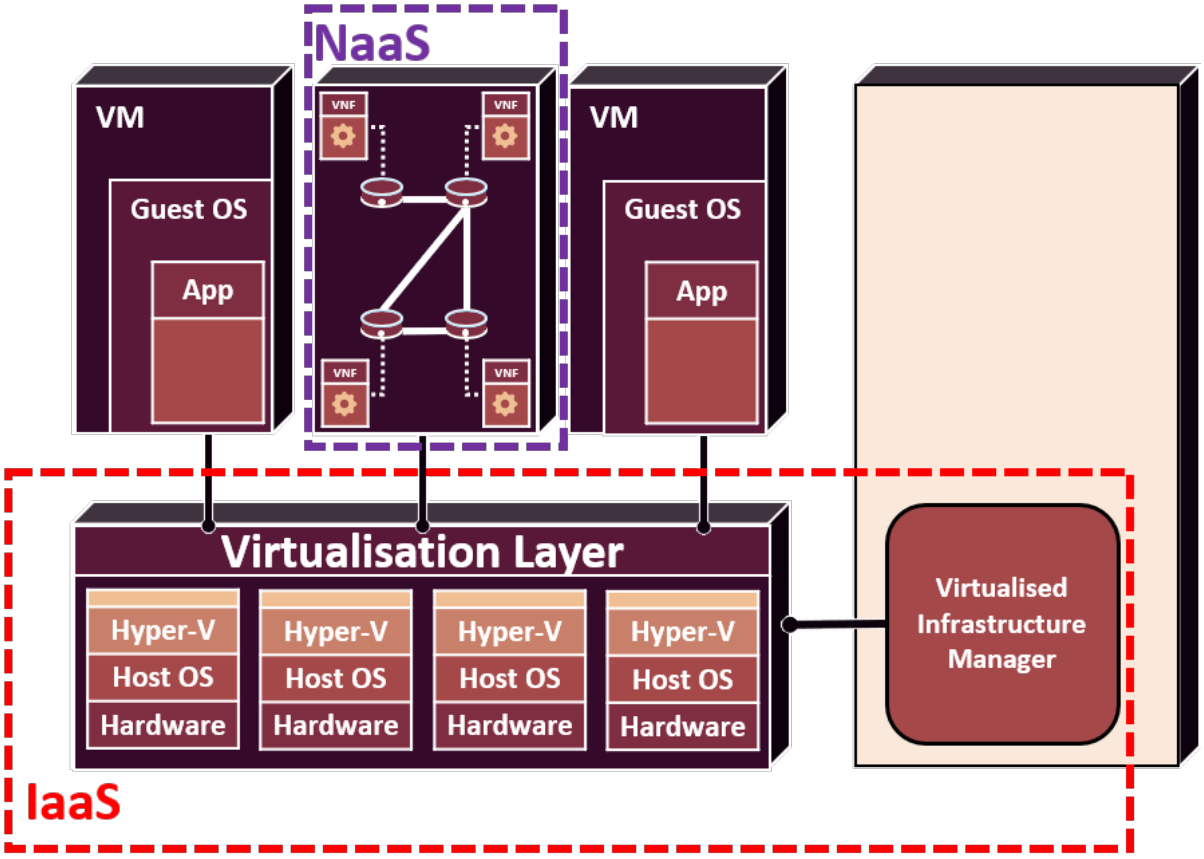

Figure 2.8: Virtualisation of Network Functions.

The bottom left domain encompasses available infrastructure as the NFVI and, on top of it, a virtualised stratum containing all VNFs and their management components. The NFVI comprises all physical resources of computing, storage and network. The infrastructure can be local or distributed, with the Virtualisation Layer providing required interoperability for each virtualised entity and making the entire environment a single unit. Figure 2.8 delineates how VIM handles the NFVI to provide resources with IaaS and NaaS. The Virtualisation Layer acts as a middle agent and encapsulates hardware-specific treatments. Typical implementations use a Hypervisor and create a VM to decouple a VNF from the NFVI. This approach distinguishes from traditional bare metal implementations of applications, where NFs directly access and control physical resources but still interoperates with the later. Another possible implementation is to use Linux namespaces to create containerised applications that abstract the underlying hardware from NFs [29]; it is a lightweight alternative, but also a more vulnerable one [30]. Figure 2.9 shows the difference between these three implementations.

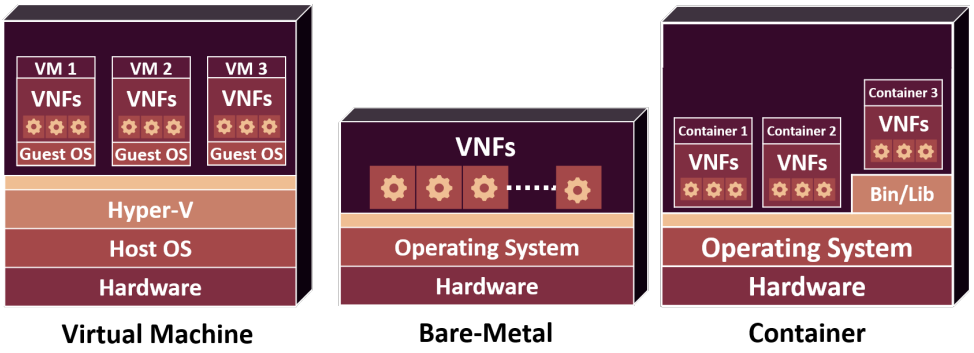

Figure 2.9: Comparison of Resource Virtualisation with Alternative Implementations.

2.2.2 5G Network Function Virtualisation Performance

Cloud Radio Access Network (C-RAN) is a common technique applied on cellular networks to transfer portions of computing load from Base Stations to Cloud Virtual Machines, allotting parts of Distributed Units (DUs) computation to Central Units (CUs). 5G dynamic environments, such as transportation and Internet of Things (IoT) applications, require, among other things, the high mobility and flexibility provided by such separation. This split divides the computing load among many agents and creates challenges about how to address the division. On 5G, the 3GPP defined the optimal split separating the processing of protocol stack, depicted on Figure 2.10, into two groups: Radio Link Control (RLC) and lower layers at DUs; and Packet Data Convergence Protocol (PDCP) and upper layers at CUs [1]. There are numerous benefits on this split, particularly on the reduction of end-to-end service delay by keeping the computing of some components at the edge. However, centralising at least signalling functions on the cloud can reduce the deployment cost without significantly increasing data overhead [2].

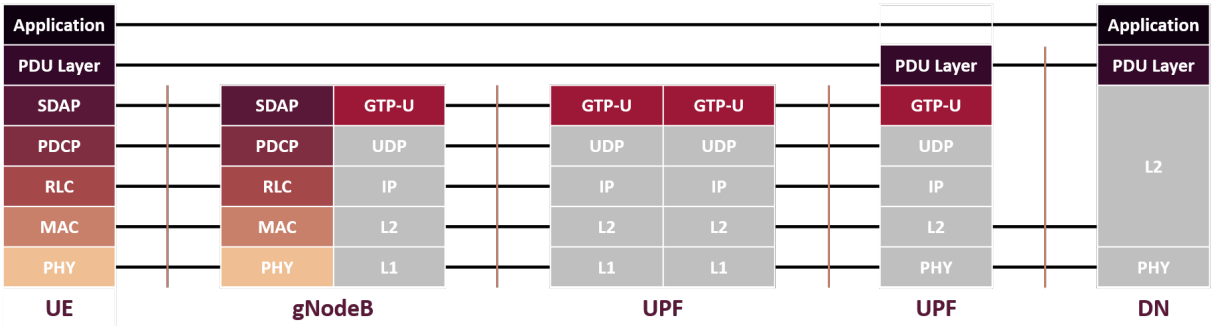

Figure 2.10: 5G Stack Layers on User Plane.

NFV further specialise the computing division by allowing the migration of NFs across the network. Traditional environments use static physical network components and apply a centralised architectural composition highly tightened to the underlying hardware. 5G NFV abstracts the resource infrastructure to allow the placement of logical network elements across the implementation but also to deploy network functions on edge devices to reduce delay on critical applications. The use of this close-to-the-edge computing scheme also creates problems related to NF placement by the orchestrator and imposes challenges on how to equitably distribute functions to avoid the reduction of QoS perceived by UEs without heavily impacting the resource usage [31]. This situation begets a challenge because the centralised infrastructure can afford massive computational load, while distributed components cannot, making the wise and fair use of provided resources a mandatory matter. The work on [32] introduced a formal definition of the NFV scheduling problem, which represents the difficulty of choosing between available resources for the instantiation of a VNF, and showed that it is an equivalent of the NP-hard Resource-Constrained Project Scheduling Problem (RCPSP). Moreover, VNF ever-changing properties facilitate an unstable environment and, because of mobility aspects of 5G use cases, function migration demands special attention on service composition, component placement and resource rearrangement.

The use of VMs as the foundation of VNFs is required to allow flexible and reconfigurable conditions, but it creates yet another problem. Virtualised environments performance degrade over time [33], and such environments must apply a composition of virtualised and physical components to reduce the generated impact and achieve intended QoS levels. Latency and reliability are essential requirements that could be heavily impacted by VNF degradation on virtualised conditions and, because ISPs are legally obliged to fulfil SLAs for consumers on mobile networks, this could severely impact the overall performance of delivered services. One approach to circumvent this problem is the creation of a cost model to evaluate the virtualised environment performance over time and predict when a VNF degrade. This model must consider and evaluate critical points of provided features on mobile core and cloud nodes [34] and their computational differences, but must also take into account the system evolution to avoid placing VNFs on compute nodes that are probably going to degrade soon, this must follow metrics such as average degradation period, expected short-time user influx and resource availability while also considering bottlenecks of 5G SBA instantiation and service overload.

2.3 Related Work

According to [4], most recent NFV performance analysis approaches solely focus on investigating system designs for vertical integration and network function placement [2, 31]. Some other efforts try to reduce placement impact by predicting VNF requirements and changing its rules. The study on [5] divides the works in this area by general functions placement and specific functions placement, the first deals with placement strategies, like replication, forwarding graph and chaining policy, while the second focus only on specific services, commonly the P-GW and the S-GW, but also caching and virtual packet inspection.

Adopting a method focused on the optimal distribution of available resources, the authors of [6] developed a performance characterisation framework for VNF private cloud deployments. The introduced framework centres on VNF capacity estimation to Network Services composition and analysis of clusterised VNF for bottleneck detection, which allows the selection of ideal resources for each workload. An essential factor of the research above is the use of Clearwater, an IMS with a typical VNF deployment, for testing. The intended scenario is the closest to our proposed use case because it tries to analyse a virtualised mobile network component using a Session Initiation Protocol (SIP) environment. It also allows user inputs to specify details about analysed deployments, as a response to the plethora of VNF application cases, which could gravely prejudice the analysis. The framework shows a considerable success on analysing CPU utilisation and component scaling even on different usage patterns.

The study on [35] introduces a framework for VNF performance profile construction, which continuously monitors the available NFVI and creates profiles of single VNFs to evaluate infrastructure deployments. This service tries to analyse the impact of each VNF at the VNF Forwarding Graph (VNF-FG) and creates a benchmarking audition history for these VNFs. This work is later improved by [7], with a new framework called GYM that defines standardised interfaces for VNF analysis and allow user-defined test inputs. The proposed framework tries to link the results of some traffic analysis tools and aims to create a reusable methodology that could be integrated into NFV Orchestrators to ease the VNF Orchestration process and avoid capacity overflows. The modularisation of its components allows higher programmability than the previous iteration. The proposed testbed also uses the Clearwater project implementation and applies SIPp Prober to generate traffic for analysis.

The work on [36] evaluates the performance of multiple VNFs along a network service chain to detect irregular operations. The tool autonomously creates profiles with performance issues encountered on different VNFs by analysing the relation between the hypervisor and the functions on each layer of the deployed NFV environment. The tool

also identifies bottlenecks and tries to achieve broader compatibility by working on numerous chaining workflows. The authours test the tool against Xen and KVM hypervisors, providing good resource evaluation results.

From the discussion above and at the best of our knowledge, there is not any work that focuses on implementing a complete virtualisation analysis integrated into 5G SBA itself. This work aims to contribute by introducing improvements at the 5GC and proposing a new function collection that aims to natively monitor the virtualised environment in real-time. The proposed performance evaluation model tries to improve on the original architecture by adding this service as a native component and by integrating its capabilities with other network services. And as a starting point, it evaluates the amount of additional traffic created by the virtualised environment itself.

2.4 Chapter Review

This chapter focused on presenting the theoretical foundations that support the proposal to be developed in this work. The chapter started by describing the history and the main concepts of mobile networks and the technological evolution between each iteration, followed by a complete description of the newly created 5G SBA and its novel implementation using network slicing. The chapter also explains and distinguishes network slicing two main enablers: SDN and NFV. Finally, the chapter discusses related work to NFV performance on 5G mobile networks. The next chapter follows by presenting the proposal of the performance evaluation model for NFV on 5G mobile networks.

Chapter 3

5G NFV Performance Evaluation Model

This chapter presents the architecture developed during this work. Firstly, in the introduction, the context of the proposal is presented, then a general description is given, explaining the architecture development. After the general description, follows a detailed architectural explanation of 5G SBA and its components, showing the new proposed components, their micro services and how they will be implemented. Finally, this work presents a discussion covering used metrics for the performance evaluation.

3.1 Introduction

This work follows a new evaluation approach, by creating two new autonomous functions and integrating their functionalities into 5G SBA, working as monitoring and evaluation functions. The proposal introduces new mechanisms and protocols necessary to enable direct communication between the SBA components and the evaluation mechanisms. For this architecture to be feasible, this work integrates overall evaluation functions as native implementations of the SBA. This integration allows deep inspection of network function state and health. However, external tools can still be used to complement the overall performance evaluation. All implemented functions follow 3GPP and ETSI guidelines to allow a direct integration but also to enable an easy convergence with the original architecture. Besides that, all other 5G projects can use the described model within their own implementations. To better visualise the benefits of this approach, to chapter will show a complete description of all proposed procedures and their expected results.

3.2 5G SBA Components

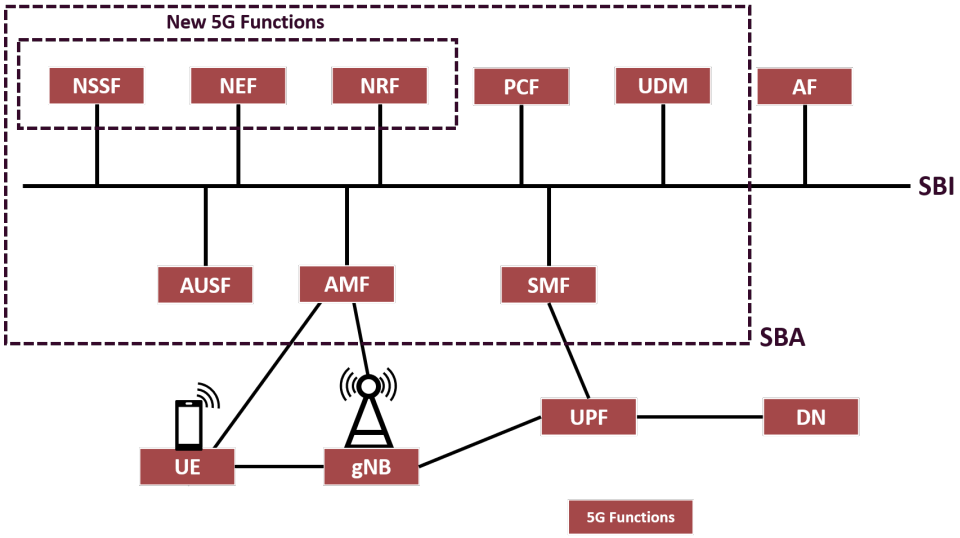

Figure 3.1: 5GC SBA Network Functions System Architecture on a Non-Roaming Configuration.

Following 3GPP TS 23.501 [19], 5G SBA services interact by standardised micro-service interfaces on a modularised Provider-Consumer architecture. Each micro-service has a well-defined procedure which allows direct communication between different NFs using a unified interface called SBI. The whole architecture follows stateless resource implementations, decoupling compute resources from storage resources. Figure 3.1 depicts the 5G SBA System architecture with a non-roaming configuration and only one Data Network (DN). Any two NFs interact using a common framework. Figure 3.2 shows the interaction between a Consumer NF and a Provider NF requesting a resource. The communication follows a Subscribe-Notify and Request-Response interaction over simple Hypertext Transfer Protocol (HTTP) methods.

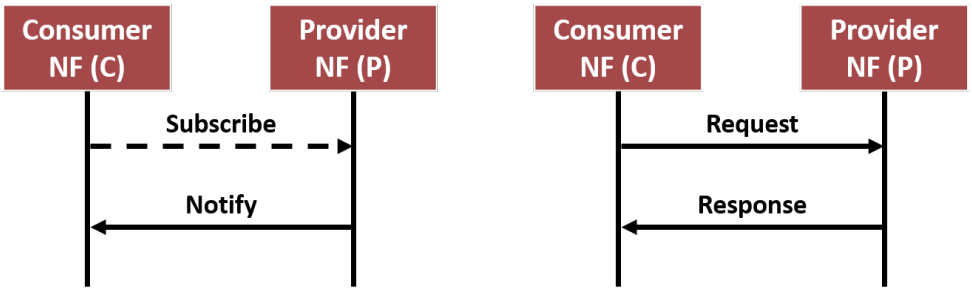

Figure 3.2: Interactions between a Service Consumer and a Service Provider NF.

3.2.1 5G SBA Virtualised Implementation

5G SBA can be a composition of virtual and physical resources. Figure 3.3 shows an example of how a 5G SBA virtualised environment would deploy a NFVI following ETSI NFV implementation standards. On the left, the environment has a group of virtualised and physical functions. On the right, it shows how an actual NFVI could implement such architecture using two compute nodes, four virtual machines and a group of virtualised resources distributed among these resources.

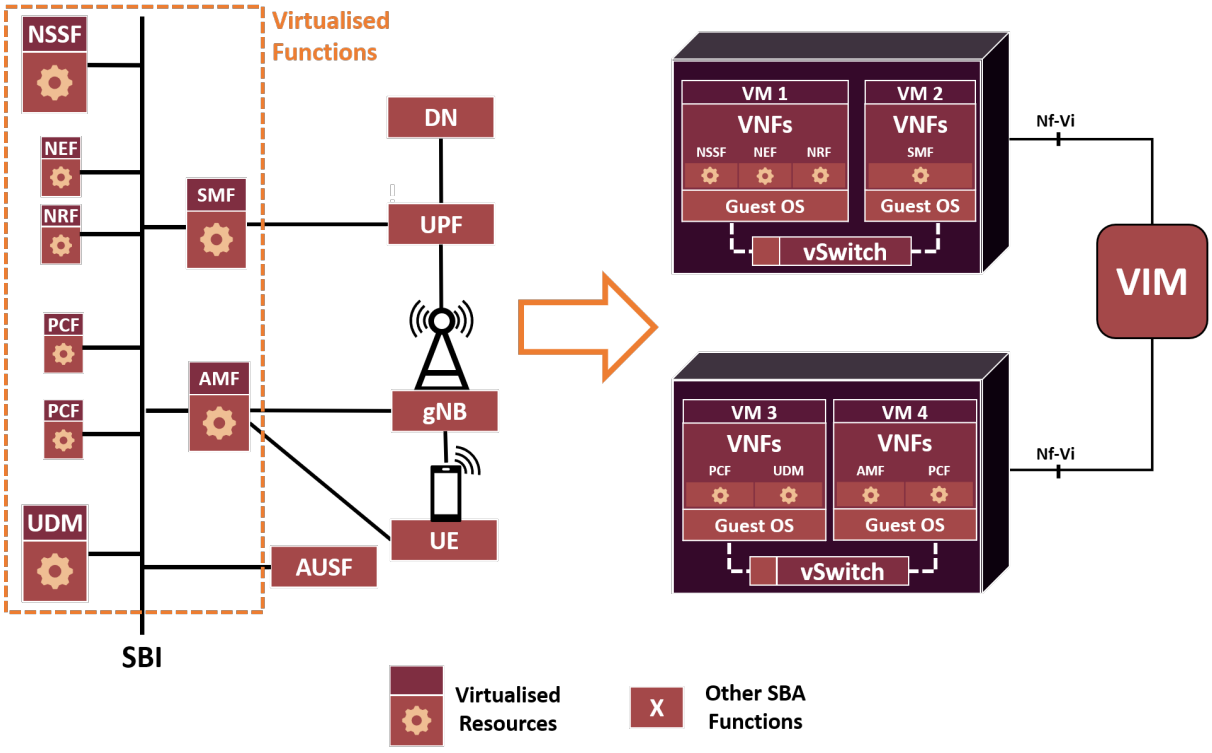

Figure 3.3: Example of a Virtualised 5G SBA Implementation.

The performance evaluation model aims to analyse the degradation and the usage of these virtual resources. The solution needs to sniff all packets carried between virtual components within the common framework in order to realise a complete verification of virtualisation status. On a standard virtualised architecture, Virtual Switches (vSwitch) created by the VIM generate a virtual network to address all packets transferred between virtual resources, Figure 3.4 shows this configuration. In hardware-based implementations, servers physically connect using local switches. In virtual implementations, all servers have more than one vSwitch responsible for connecting all virtual machines and delivering a logical connection. These switches provide inter-VM connectivity but also manage the connection to outside networks.

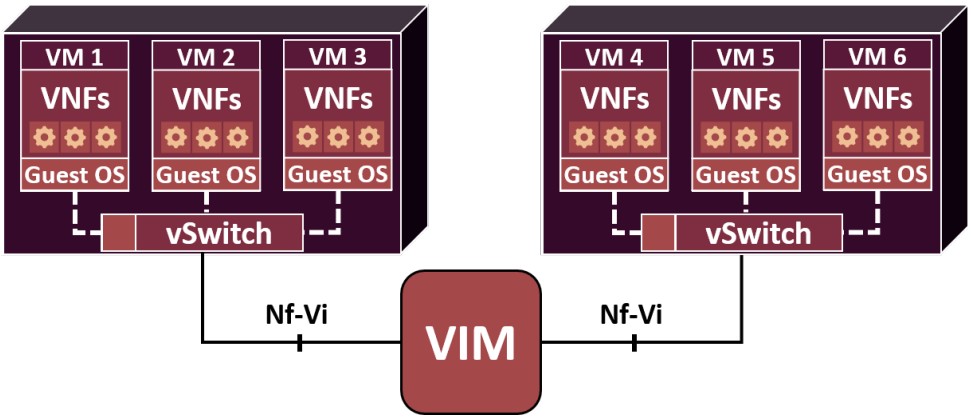

Figure 3.4: The use of vSwitches by the VIM to provide Network Control over the NFVI.

3.3 Virtualised Environment Observation Functions

This work offers a NFV performance evaluation architecture integrated into 5G SBA. The used monitoring mechanisms intend to evaluate virtualised functions without heavily impacting the environment. This project deeply analyses proposed processes, their efficiency and further review applied techniques to ensure a cohesive system. To allow a constant analysis of virtualised functions, the solution monitors the SBI at each compute node using a new Observation Function (ObF). Each ObF forwards gathered data to a centralised Performance Evaluation Function (PEvF) responsible for monitoring all virtual resources and generating consolised analysis based on the collected information. Figure 3.5 shows how these two new functions act at 5G SBA by using the simple SBI communication as an access point.

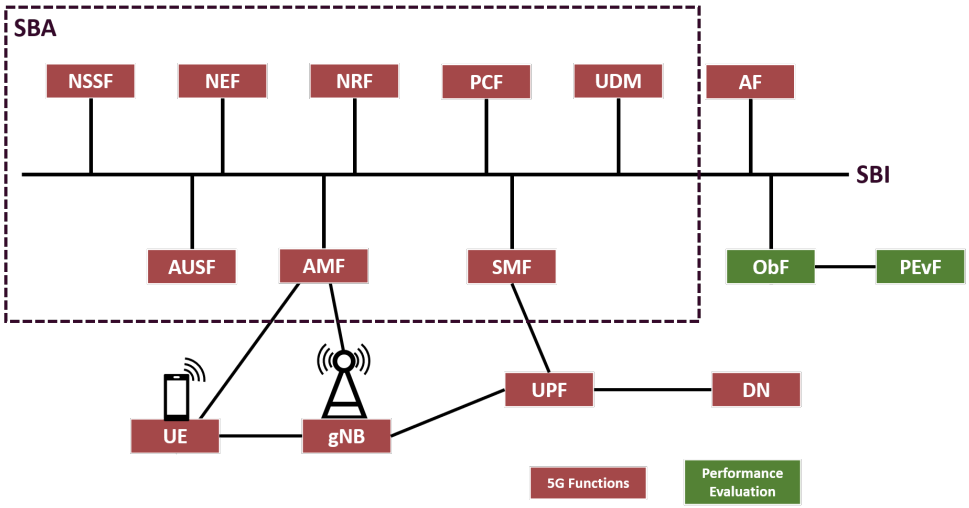

Figure 3.5: 5G SBA with the two Newly-Introduced Functions.

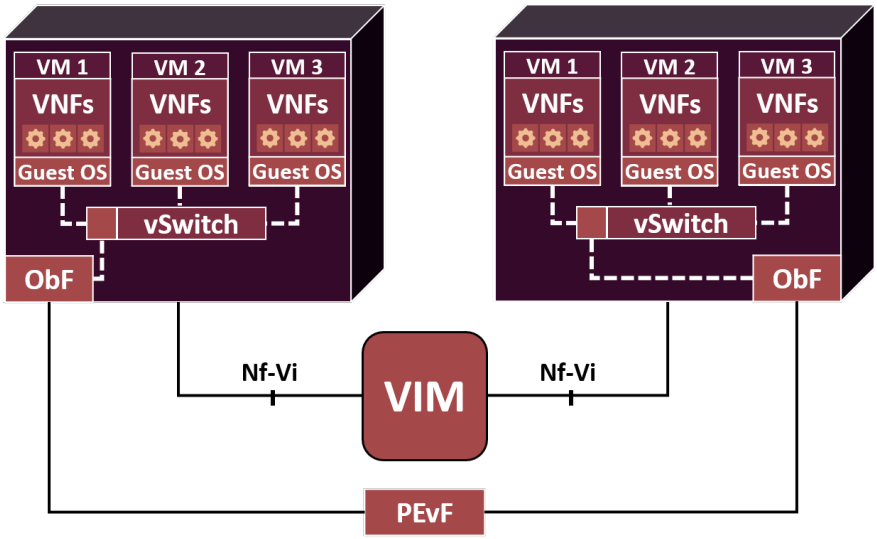

Figure 3.6: Enabling new NFVI Components for Evaluation.

To perform an easy analysis service, a local ObF will listen to the main vSwitch at all compute nodes, monitoring transferred data and performing packet inspection to filter which packets are 5G SBA common interactions. This will allow the delivery of precise information and will provide the PEvF with necessary data to generate a full performance evaluation. Figure 3.6 shows how to enable the addition of these components at an NFVI using a simple configuration. In order to avoid more resource competition with other 5G SBA components, the ObF will be a bare-metal implementation. This choice also simplifies the development and creates a solid component with little impact on the overall architecture. Table 3.1 contains all only micro-services provided by ObF and the semantics of their interactions. All procedures follow 5G SBA pattern for micro-service interactions.

Table 3.1: ObF Services

Service	**Operation**	**Semantic**
VNFMonitor	VNFSubscribe	Subscribe/Notify
	VNFUnSubscribe	
	VNFStatus	Request/Response
SliceMonitor	SliceSubscribe	Subscribe/Notify
	SliceUnSubscribe	
	SliceStatus	Request/Response
ServiceMonitor	NRFStatus	Request/Response
	ServiceSubscribe	Subscribe/Notify
	ServiceUnSubscribe	
	ServiceStatus	Request/Response
ObservationMode	StopObservation	Request/Response

The PEvF, on the other hand, will be centralised at the controller node to avoid imposing high processing loads at the compute nodes as the CUs has more load capacity than DUs. The proposed PEvF analyses and evaluates how packets flow between network functions to identify bottlenecks and delays that could cause an MNO not to fulfil any agreed SLAs. The function focuses on monitoring micro-services packet delivery to detect delay between any two functions and to identify performance degradation on virtual machines. Using collected information, the function tries to foresee bottlenecks impacts at NS composition and how many time it would take to impact the SLA compliance. PEvF will have two approaches, firstly, a packet monitoring mode where all vSwitch sniffing will be analysed only by its delay and will provide enough performance information about the overall architecture and specific VNF functions, and a native SBA mode, where it will analyse micro-services, filtering the groups by NS and also by Network Slices to provide in-depth analyses about specific network subdivisions and to identify the response time of each NF and each of its micro-services. PEvF does not provide micro-services itself and only consumes from ObF.

3.3.1 Overall Architectural Advantages

The performance evaluation model has a composition of distributed and centralised functions, working in a passive mode to avoid interfering on the other components. The use of a centralised PEvF creates a unified interface for performance monitoring, but it also avoids imposing high computation loads on each compute node, this decision follows a similar approach of ETSI optimal split between CUss and DUs [1]. On the other hand, the use of a bare-metal ObF implementation prevents the function to generate resource competition with the VNFs but also allows a constant monitoring system, while the sniffing mode also reduces any possible computation load that the function could apply to the VNFs.

As described, the model has two operational modes. The first is focused on giving a passive monitoring about packets of unknown operation, while the second mode improves on the first by integrating its functionalities within the 5G SBA .The use of two performance evaluation modes allows a hybrid solution. The first mode serves on situations where the details of 5G SBA VNF implementation are not well known, while also reducing the impact of performing any in-depth packet inspections on resource-constrained environments. The second mode fully integrates both ObF and PEvF functions as native 5G network functions that can deliver micro-services using 5G guidelines, while allowing the segregation of the analysis by Slices and services for a better overall visualisation.

3.4 Packet Monitoring Mode

The packet monitoring mode does not assume any SBA implementation or any previously instantiated NF. The PEvF verifies with the centralised VIM the current virtualised and physical infrastructure dependancy by requesting a complete list of current compute nodes, the VMs running at each node, and VNFs instantiated at each VM. This first step gives the PEvF enough information to create a map of instantiated VNFs, vSwitches, VMs and the underlying compute nodes. Figure 3.7 shows a virtualised infrastructure map example, and the separation of it in three different layers to ease how PEvF deals with degradation analysis. The current architecture assumes a northbound interface available at the VIM with enough information about the virtualised infrastructure as required by ETSI NFV definitions [37].

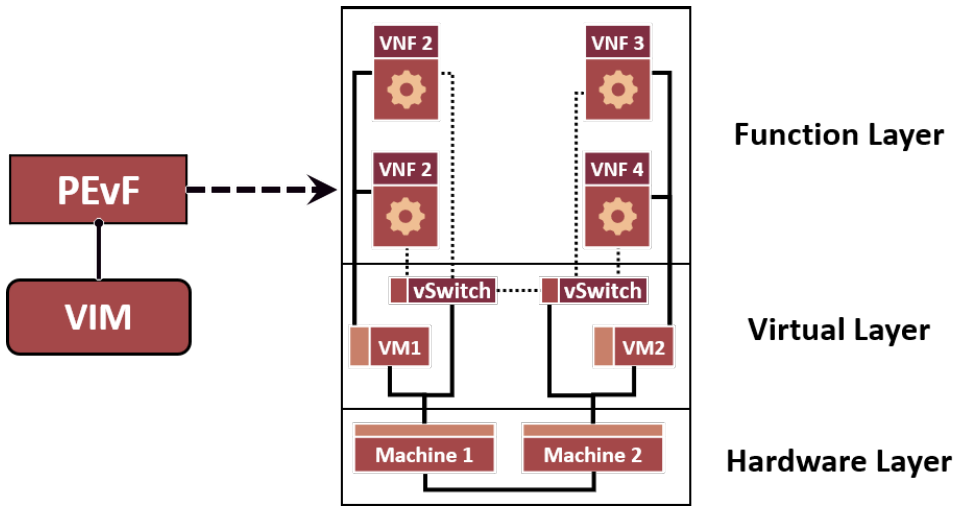

Figure 3.7: Virtualised Infrastructure Map and each of its layers.

After receiving enough data to produce the virtualised infrastructure map, PEvF stores collected information using a structured JSON file (as shown in Appendix A) and then it starts the sniffing mode for each distributed ObF at all active compute node discovered. The initial connection between PEvF and ObF follows two distinct procedure possibilities, as shown in Figure 3.8. On the left communication pattern, PEvF send a *VNFSubscribe* operation to ObF *VNFMonitor* service, listing the targeted vSwitch and all wanted VNFs related to it. Then, ObF starts to operate at sniff mode, attaching itself to the vSwitch, monitoring all of its traffic. After that, every time any monitored VNF send or receive any data, ObF will sniff and forward the information about the packet and the followed path to PEvF using a *VNFNotify* command. This pattern repeats until PEvF send a *VNFUnSubscribe* operation to ObF *VNFMonitor* service, listing the targeted vSwitch and any VNFs it doesn't want to monitor anymore.

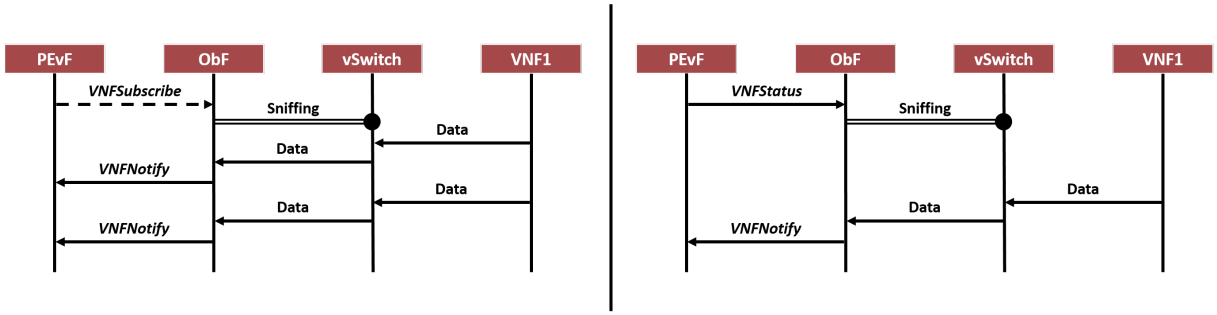

Figure 3.8: Representation of *VNFSubscribe* and *VNFStatus* Procedures.

The other procedure possibility follows a simpler sequence with only only on request sent and one response received by the PEvF. As shown on the right part of Figure 3.8, PEvF sends a simple *VNFStatus* operation to ObF *VNFMonitor* service, informing the targeted vSwitch and a single VNF related to it. Then, if not already operating at sniff mode, ObF starts to operate at it and attaches itself to the chosen vSwitch, monitoring all of its traffic until something related to the chosen VNF is forwarded by it. Only the first time the monitored VNF send any data, ObF will sniff and forward the information about the packet to PEvF using a *VNFNotify* command. The response from ObF to PEvF only happens this one time on this procedure. In general, ObF only activates its sniffing mode after receiving either a *VNFSubscribe* operation or a *VNFStatus* operation and it continuously keeps sniffing until PEvF sends a *VNFUnSubscribe* textit at its *VNFMonitor* service or a *StopObservation* operation at its *ObservationMode* service.

3.5 Service-Based Mode

The Service-Based Mode also produces the virtualised infrastructure map following the same pattern defined for the packet monitoring mode. The PEvF also creates a second map called slicing map. To do this, it verifies with both the VIM and the VNFM the current slicing separation by requesting a complete list of VNF intances related to each slice and the services provided by them. It doesn't need to create a correlation between compute nodes, VMs and VNFs for each slice because the virtualised infrastructure map already provides this information. Figure 3.9 shows a slicing map example, and the separation of it in three different layers to ease how PEvF deals with micro service analysis. The current architecture also assumes a northbound interface available at both the VIM and the VNFM with enough information about the virtualised infrastructure and micro-services provided by VNFs as required by ETSI NFV definitions [37].

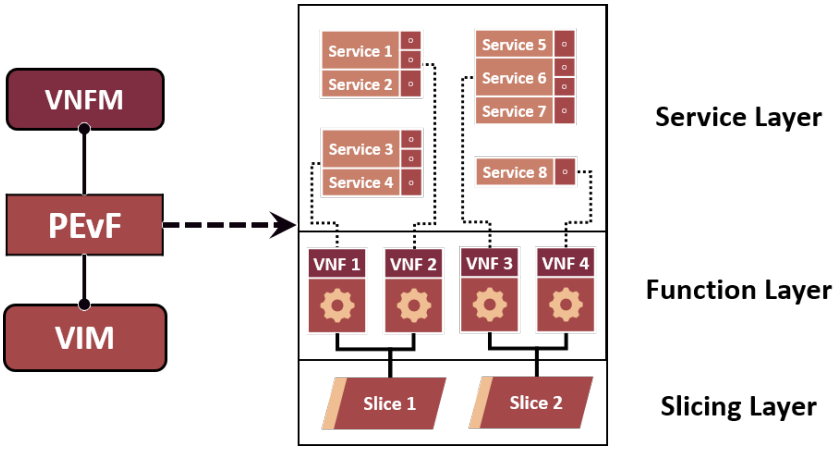

Figure 3.9: Slicing Map and each of its layers.

In Service-Based Mode, also ObF analyses common 5G SBA operations at monitored packets, allowing to filter them by Network Slice or by offered micro-services. To do this, ObF not only monitors common packets but also verifies the composition of Services and Operations transmitted between a Consumer NF and a Provider NF that are using a Subscribe-Notify or a Request-Response interaction over HTTP as previously shown by Figure 3.2. Bot the VNF discovery and selection by ObF also follow 5G SBA rules using the three new 5G core functions NEF, NRF and NSSF as service providers in sniffing mode to identify active slices and micro-services and constantly monitor their operations. With the virtualised infrastructure map, the slicing map and the gathered information from these three specific functions, PEvF can offer a proper separation between slices and also between services for monitoring.

3.5.1 NF Discovery and Selection

All NFs must discover each other to start providing and consuming information. The NRF acts as a central repository of available NF services, it allows the initial inter-NF interaction by providing a NF service catalogue and all operations related to them. The function provides this feature with two micro-services. The *NFManagement* service deals with registration, update, status and service subscription-related operations, while *NFDiscovery* only operates a service request operation, which enables any NF to discover all NF instances providing a particular service. This second micro-service is particularly important because more than one instance can provide the same service. Therefore an NF can choose between available providers and decide which one better fit its requirements. Table 3.2 shows all services provided by NRF and the semantics of their interactions as specified in 3GPP TS 23.502, clause 5.2.7 [38].

Table 3.2: NRF Services

Service	**Operation**	**Semantic**
NFManagement	NFRegister	Request/Response
	NFUpdate	Request/Response
	NFDeregister	Request/Response
	NFStatusSubscribe	Subscribe/Notify
	NFStatusNotify	
	NFStatusUnSubscribe	
NFDiscovery	Request	Request/Response
AccessToken	Get	Request/Response

The details of the operations won't be described because ObF only needs to identify the service related to the sniffed packets, which can be done without further requesting more information than the already analysed from requests and responses at sniffed packets. The initial connection between PEvF and ObF also follows three distinct procedure possibilities. Figure 3.10 shows the details of the first possible procedure conformed by PEvF and ObF for communication about micro-services. In this procedure, PEvF send a *ServiceSubscribe* operation to ObF *ServiceMonitor* service, listing the targeted vSwitch and all wanted VNFs micro-services related to it. Then, ObF starts to operate at sniff mode, attaching itself to the vSwitch and monitoring all of its traffic. After that, every time any monitored VNF send or receive data about these specific services or also when NRF sends responses related to this VNF (*NFManagement* and *NFDiscovery*), ObF will sniff and forward the information about the packet and the followed path taken to PEvF using a *ServiceNotify* or a *NRFNotify* command, respectively. This pattern repeats until PEvF send a *ServiceUnSubscribe* operation to ObF *ServiceMonitor* service, listing the targeted vSwitch and any services realted to a secific VNF that it doesn't want to monitor anymore. PEvF can also be halt the service monitoring by sending a *StopObservation* operation for ObF *ObservationMode* service. NRF doesn't monitor its own instance, so it is not possible to monitor NRF operations about itself. But the second procedure allowed by ObF covers the NRF monitoring.

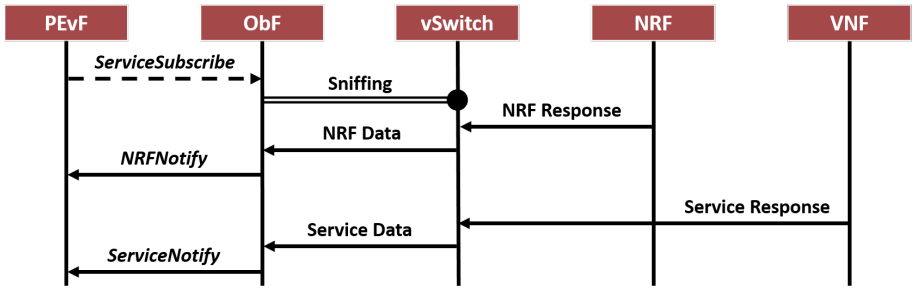

Figure 3.10: Representation of *ServiceSubscribe* Procedures.

The second possible procedure follows a simpler sequence using the same approach of VNF-related status monitoring. As shown on the left part of Figure 3.11, PEvF send a *ServiceStatus* operation to ObF *VNFMonitor* service and listing the targeted vSwitch and a single VNF service related to it, on a similar fashion of VNF-related procedures. Then, if not already operating at sniff mode, ObF starts to operate at it and attaches itself to the chosen vSwitch, monitoring all of its traffic. After that, only the first time the monitored VNF send any data about the requested service, ObF will sniff and forward the information about the packet to PEvF using a *ServiceNotify* command. The response from ObF to PEvF also happens only one time in this case.

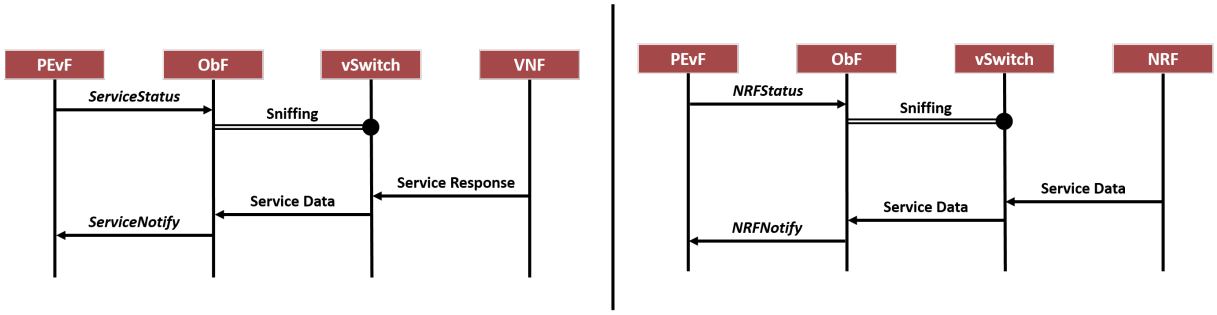

Figure 3.11: Representation of *ServiceStatus* and *NRFStatus* Procedures.

On the third possible procedure, to check the status of VNF-related NRF responses, the PEvF can also send an *NRFStatus* informing the targeted VNF, its vSwitch and the wanted service. As shown on the right part of Figure 3.11, after that, following the same approach of the previous procedure, only the first time the NRF send data about the monitored VNF service, ObF will sniff and forward the information about the packet to PEvF using a *NRFNotify* command. The separation between *ServiceNotify* and *NRFNotify* for service status allows the PEvF to use *ServiceNotify* operation to watch even the NRF by decoupling the VNF service monitoring from NRF SBA monitoring.

Table 3.3: NSSF Services

Service	**Operation**	**Semantic**
NSSelection	Get	Request/Response
Availability	Update	Request/Response
	Subscribe	Subscribe/Notify
	UnSubscribe	
	Notify	
	Delete	Request/Response

3.5.2 NF Slice Identification

For Network Slice selection and monitoring, all NFs must use NSSF services. NSSF centralises all information about Slicing, and each Network Slice has a unique identification called S-NSSAI as defined in clause 5.2.16 of TS 23.502 [38]. This component is continuously communicating with AMF to provide information about slice selection on user authentication procedures, and this constant communication is useful for data gathering by the ObF. The function provides its features with two micro-services. The *NSSelection* service deals with slice selection in all network scenarios, including home networking or roaming serving. The *Availability* service enables the update of information about provided Network Slices by the AMF. The use of slice monitoring with these services can facilitate the environment isolation of tenant supervision. Table 3.3 shows all NSSF services provided and the semantics of their interactions as specified in 3GPP TS 23.502, clause 5.2.7 [38].

3.6 Performance Evaluation Model

The goal of the case study is to evaluate the performance of 5G Network Function Virtualisation using an implementation of 5G SBA under a complete virtualised infrastructure. The key component under study is the virtual switch. A virtual switch is responsible for forwarding all traffic between network functions under a virtualised architecture. The system consists of a group of virtualised network functions deployed at an NFVI. The requests are sent via virtual links from one VNF to another using the aforementioned virtual switches. The model only evaluates NFV applications over a 5G environment, therefore only the subsets of 5G SBA VNFs are considered to be part of the system. The study will be conducted so that the effect of components outside the system is minimised. The focus is to evaluate the amount of additional traffic generated by a virtualised environment.

3.6.1 Services and Expected Outcomes

On Monitoring Mode, the services offered by the VNFs are not a great concern because the ObF evaluates the system performance only by measuring packet delivery ratio and direct virtual resource degradation. On this first scenario, monitoring virtual switches provide enough information to define packet delivery delays, which later consolidates as NFVI mapping information. On SBA Mode, NRF and NSSF services are vital connection points as they offer information about available services over the infrastructure and Slice mapping over the 5G SBA, respectively. Other SBA components provide their services using the same Request-Response and Subscriber-Notify scheme as described before. The

performance evaluation made by PEvF will consider each service separately using the Slicing map, allowing the compilation of degradation data.

Considering that the systems performs all services correctly, the evaluation model will measure its efficiency by a time-rate-resource scheme [39], using a comparison between total forwarded information and control data to evaluate the performance costs of such network infrastructure. This evaluation shows how much useful data actually flows at the system and give a view of NFV efficiency. First, it will consider the total time for the completion of a service, then it will evaluate the service performing rate, and lastly, it will verify the total resource consumed while performing the service. There are two possible scenarios when the system does not perform correctly, the first is when it gives a result, but with an incorrect output. And the seconds happens when it does not give any result. To give meaning to the analysis, PEvF will consider relevant metrics. The next section will discuss the selected metrics to perform this evaluation.

3.6.2 Metrics

PEvF continuously monitors VIM and VNFM northbound interfaces and also aggregates multiple metrics gathered by the distributed ObFs. From VIM it receives information to create both the virtualised infrastructure map and the slicing map, also using the information received from VNFM to create the slicing map. From ObF, it receives information collected from many compute nodes and then use it on two different modes. PEvF consolidates all gathered information and then use the generated maps to create events and notifications about virtualisation health. This information is stored on logs but can also be requested online.

Table 3.4: NFV Performance Metrics

Resource	**Code**	**Metric**	**Units**	**Source**
Compute	C.1	Processor Usage	Seconds (s)	VIM
	C.2	Processor Utilisation	%	VIM
Network	N.1	Packet Count	Number (#)	ObF
	N.2	Octet Count	Number (#)	ObF
	N.3	Dropped Packet Count	Number(#)	ObF
	N.4	Errored Packet Count	Number (#)	ObF
Memory	M.1	Buffered Memory	MB	VIM
	M.2	Cached Memory	MB	VIM
	M.3	Free Memory	MB	VIM
	M.4	Memory Slab	MB	VIM
	M.5	Total Memory	MB	VIM
	M.6	Memory Used	MB	VIM

It is necessary to define metrics and default evaluation methods to store and display these logs. ETSI GS NFV-TST 008 V2.4.1 [40] describes and analyses a group of metrics to measure the QoS perceived by the consumer of any NFV regarding both physical resources and their virtual equivalents. Table 3.4 show these metrics classified by resource impacted. Each metric has a unique code to facilitate its identification on further analysis and the information about the source used by PEvF to gather such data. For compute resources, Processor Usage (C.1) represents the total time of instruction execution at the compute node compared with the whole monitoring interval, while Processor Utilisation (C.2) represents the ratio between C.1 and the monitoring interval. For network resources, Packet Count (N.1) represents the number of packets successfully transferred over an interface, Octet Count (N.2) represents the total amount of bytes transferred over an interface considering the sum of all successfully delivered packets, Dropped Packet Count (N.3) represents the number of unsuccessful packet transmissions which occurred because of lack of resource, and Errored Packet Count (N.4) accounts for two groups of corrupted packets in relation to a unique interface. The first group is related to corrupted packets received with wrong sizes or integrity problems; while the second covers all failed attempts to send any packet. For memory resources, Buffered Memory (M.1) represents the total space used for raw disk block storage at a given time; Cached Memory (M.2) is the sum of all memory used as cache; Free Memory (M.3) accounts for the unused memory; Memory Slab (M.4) is the total amount memory used as cache by the kernel for data structure and object storage; and Total Memory (M.5) is the sum of usable memory; and Used Memory (M.6) is a generated metric that can be calculated using previous memory-related metrics [40]. The performance evaluation of this work aims to verify the efficiency of user data forwarding by analysing how much of useful data flows within the network, thus leaving Compute and Memory information as redundant or not valuable data to collect. Future works can evaluate such parameters on a more complete study.

All described metrics provide unique pieces of information which add paramount value to the evaluation. Therefore, the performance evaluation model proposed by this work can use all presented metrics without generating data redundancy [39]. Because virtualisation imposes resource competition and new levels of transmission between packets, it also creates more chances of unavailability issues. Lastly, all this information will be compared with the Amount of User Data (N.5) and Amount of Control Data (N.6) to show the actual performance of a virtualised environment. Table 3.5 shows the chosen metrics used by PEvF for performance evaluation. These metrics can give a good overview about performance evaluation. Next section will further improve this view by defining the evaluation use cases and characterising the test workloads for each case.

Table 3.5: Final NFV Performance Metrics

Resource	Code	Metric	Units	Source
Network	N.1	Packet Count	Number (#)	ObF
	N.2	Octet Count	Number (#)	ObF
	N.3	Dropped Packet Count	Number(#)	ObF
	N.4	Errored Packet Count	Number (#)	ObF
	N.5	Amount of User Data	Ratio (%)	ObF
	N.6	Amount of Control Data	Ratio (%)	ObF

3.6.3 Parameters and Important Factors

To give meaning to the performance evaluation, the process will consider a list of crucial parameters with high impact over the system performance. To suffice such need, speed of remote physical and virtualised CPUs, speed of the network, virtualisation OS overhead and virtualisation network overhead will be considered. The performance evaluation will also consider two important factors, the number of calls and the number of active functions on the overall architecture on each test.

3.7 Chapter Review

This chapter presented the proposed model for NFV evaluation on 5G mobile networks. It started by detailing 5G SBA communication mechanisms and its actual implementation using NFV. Then it describes the two new proposed functions ObF and PEvF and how they deliver a feasible evaluation model for NFV on 5G, outlining their calls and how they interact. The chapter finishes by defining the metrics used by PEvF to evaluate an NFV environment. The next chapter will discuss the implementation of the proposed infrastructure, the use cases for performance tests and the methods used to generate valid synthetic workloads for each use case, followed by the results collected for each test case.

Chapter 4

Implementation and Experimental Results

This chapter presents the experimental implementation of the ETSI NFV architecture that was used to evaluate the 5G NFV standard use cases under different circumstances and configurations. Due to equipment limitation, a set of power-constrained devices were used. The first evaluations include an analysis of the main functionalities given by NFV deployment for NFs, as well as the effectiveness of the implemented virtualisation enablers. It is evaluated how the selected software implementation cover all the requirements for a complete NFV scenario, the possibility of using this infrastructure as an actual 5G SBA and finally, a set of tests were performed in the deployed infrastructure that covered the URLLC, mMTC and eMBB use cases.

The implemented infrastructure deployed for experimental evaluation is based on a resource-limited scenario. Nevertheless this was considered enough to enable resource competition, data gathering and traffic analysis. The deployment replicates a complete NFV architecture followin 5GPPP standards and allows to design test scenarios close to actual use cases, which made possible to test the VNFs and its evaluation.

This chapter is organised as follows. First, a general description of the aimed architecture is given, explaining the development and all used technologies for deployment, with the explanation about such choices and how they follow defined standards. After an initial description, follows a detailed explanation of used components, showing their capabilities, connections and explaining the infrastructure composition with its specifications. Then, all the test scenarios are described with their deployed virtualised functions and connections, followed by the monitoring perspectives and their advantages. Finally, this work presents the results, some observations about them and a complete discussion with the future steps of this research.

4.1 Architecture Implementation

Figure 4.1 depicts the implementation of the virtualised architecture using ETSI NFV guidelines. Open Source MANO (OSM) Release FIVE [41] acts as NFVO and VNFM, while OpenStack (Ocata) [42] provides VIM functionalities. These choices follow the de-facto standard software used by many 5G-PPP projects [**?**, 43, **?**]. For VNFM, OSM provides native VNF instantiation, service initialisation and runtime management of virtualised services, achieving complete lifecycle management. For NFVO, it provides software resource management [44]. Openstack allows the implementation of compute, storage (both through Nova component) and network (through Neutron component) resources over an NFVI for the whole architecture. Both OSM [45] and Openstack [46] provide northbound interfaces as required by ETSI NFV guidelines.

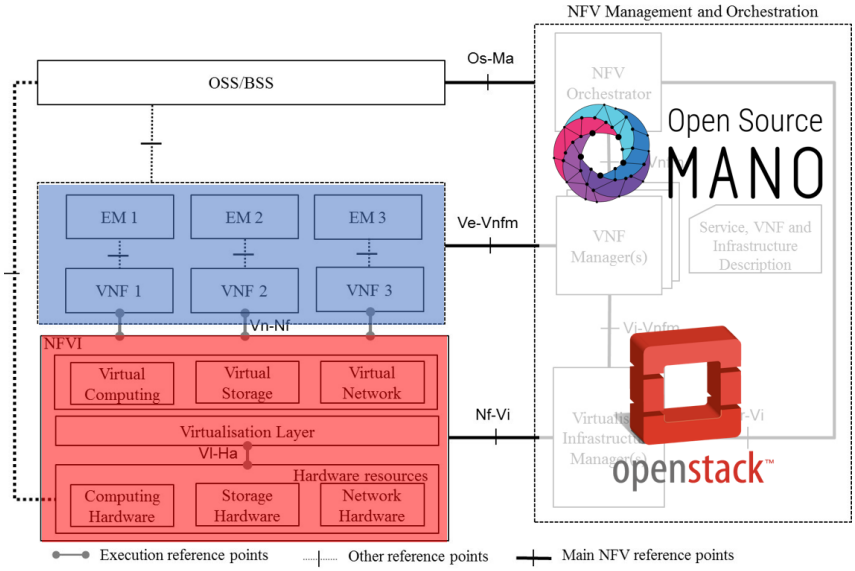

Figure 4.1: Implementation of ETSI NFV Architecture.

This configuration enables a flexible, automated, and cost-effective implementation of network services, while also following ETSI guidelines for virtualised network environments. Figure 4.1 shows that the provided architecture also fully implements all requirements presented in Section 2.2 for an NFV environment. For advanced functionalities, Openstack natively supports Service Function Chaining and Network Slicing, and OSM started to support Network Slicing since version 5. Figure 4.2 [47] presents the integration between OSM and Openstack components and also their interfaces. It is worth noting that the lack of supporting material about a novel technology such as OSM hindered the configuration of testing environments.

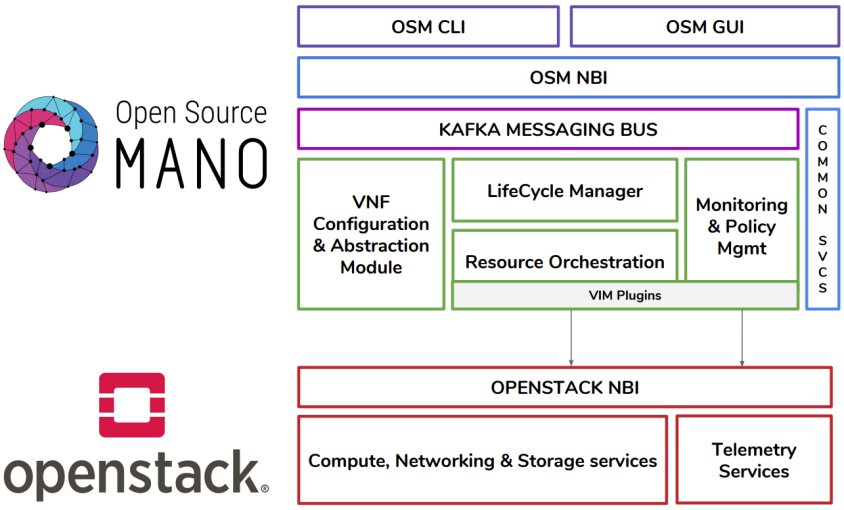

Figure 4.2: OSM and Openstack integration points.

4.1.1 Infrastructure Composition

In order to deploy the required architecture and allow reliable performance evaluation model tests, it is necessary to replicate an actual 5G SBA scenario with decentralised VNF communication over resource-constrained infrastructure. Figure 4.3 shows the current implementation of a decentralised NFV developed until now, having two main components. The first component is a server acting as a Centralised Unit and performing both Management and Orchestration for available resources, serving as an NFV MANO. The second component is the NFVI, composed by an assortment of compute nodes acting as Distributed Units and making virtual resources available for VNF deployment.

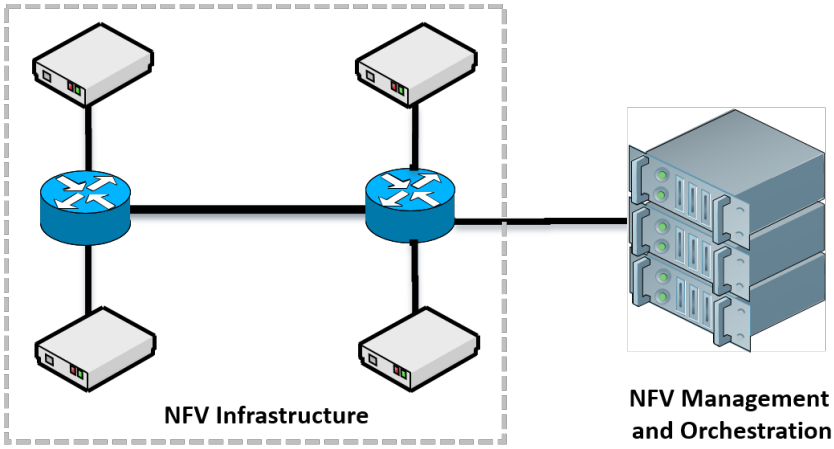

Figure 4.3: Infrastructure Implementation and relevant Components.

Server Specification

The server is a Mini-ITX i5 3Ghz with 8GB and 1TB SDD, operating with Ubuntu Server 18.04 for Hyper-V support. Its runs a VM which encompasses an OpenStack Ocata controller along with OSM Release FIVE. This VM runs as a KVM guest and has three network interfaces. The first one delivers VIM communication between the OpenStack controller and the NFVI. The second interface is for VNF lifecycle management by the OSM. The last interface gives connectivity and allows remote configuration of the server.

Compute Nodes Specification

The compute nodes are a group of different Raspberry Pi (RPi) versions (3B, 3B+, 4B and Zero W) to allow a low-cost and power-constrained test scenario. All RPis operate with Raspbian Release from 06-2019. It runs Openstack Nova and Neutron components on bare-metal configuration for a reduced environment. All RPis also has three interfaces. The first one delivers VIM communication between the local Nova and Neutron components and the centralised OpenStack controller for physical resource management. The second interface is exclusively for VNF lifecycle management by the OSM. The last interface acts as an access point for NS users, providing connectivity to deployed VNFs through virtualised interfaces.

4.2 Use Cases and Scenario Configuration

The use cases under evaluation are categorised under three groups, eMBB, URLLC and mMTC. eMBB requires moderate to high download rates on stable connections for end-users. A simple Web-based access test with medium throughput can fulfil this requirement by analysing if increasing user data throughput will also increase NFV infrastructure data. URLLC demands low latency and high availability, focusing on edge computing scenarios. Voice conversation requires a maximum delay of 150 ms to perform an acceptable call [48, 49], 50 ms to deliver it with good quality [50] and 10 ms for high-quality audio streaming [48]. Therefore a VoIP-based test with good quality over NFV infrastructure can deliver a fair analysis for these cases. mMTC requires high traffic density and good interconnection and can be characterised by testing temporised IoT connections [51] over a window to identify gaps on transmission and their impact of the overall system. The tests integrate three different slices within these different use cases. The next sections discuss the methodology used to generate the synthetic workloads used to evaluate the performance on each use case while also introducing expected results.

4.2.1 URLLC - VoIP Scenario

Potential URLLC use cases on communication services include vertical industries such as Smart Energy, Augmented Reality and Smart Traffic control [52]. Common available traffic sets and use case of URLLC still do not offer direct traffic replication or synthetic workload generation methods to evaluate this scenario [53, 54]. Because of that, in this work, to deploy an URLLC use case under an NFV testbed and make it close to an actual scenario, the presented test implements a Voice Over IP (VoIP) NS call and its components using VNFs. Figure 4.4 depicts the implementation of the physical infrastructure and virtualised resources used by ETSI NFV. The centralised server locally runs a SIP Core using a Kamailio server VNF with native Real-time Transport Protocol (RTP) support. The first compute node acts as a router between the others by deploying two VNFs: a Router and a Domain Name System (DNS). The second and the third compute nodes run one Access Point (AP) VNF each, connecting with the first compute node for service integration.

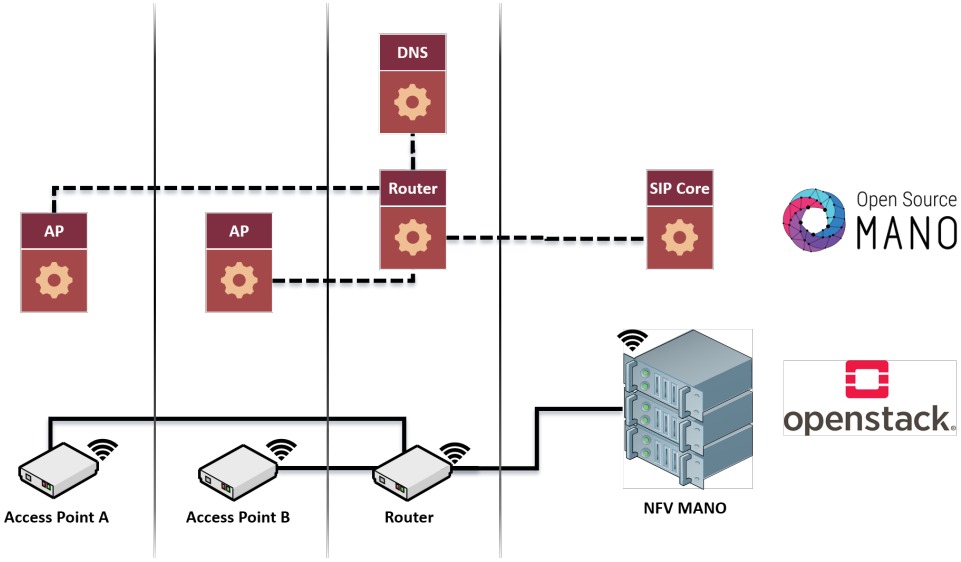

Figure 4.4: VoIP Infrastructure Implementation with Virtualised Resources above it.

Both APs were RPi 3B+, and the Router were a RPi 3B. All VNFs ran using VMs with one Virtual CPU (vCPU), 128MB of memory and 4GB of storage. Figure 4.5 shows all virtual links and VNFs created at OSM and connection points between them. It is important to note that the physical connection topology differs from the actual virtualised connections deployment. The NS only needs to know the configuration of the virtualised network, and this happens because the NFV environment abstracts the details about the physical topology.

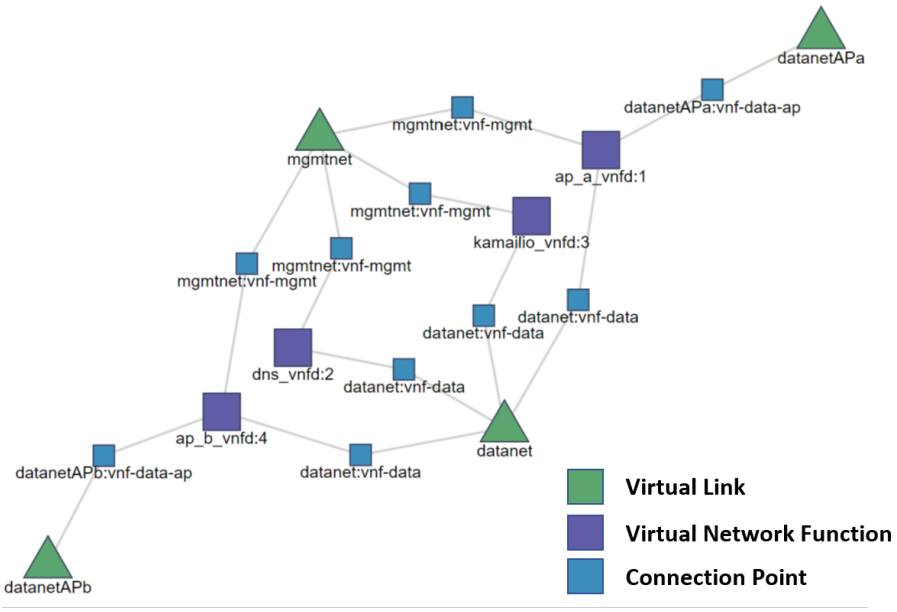

Figure 4.5: Virtual Components for the URLLC Use Case and their Virtual Links.

Workload Generation

Artificial voice payload was generated for the tests. It works by sending a packet with a voice payload of 20 Bytes and a sample period of 20 ms. The time between packets is 0.02 seconds, modelled using a normal distribution and a standard deviation of 0.0038 seconds. With such configuration, this workload follows the specifications of Codec G.729, with a sample size of 10 bytes (80 bits) and operating on sample intervals of 10 ms, allowing a bit rate of 8 Kbps. G.729 is among the most used Codecs for VoIP calls and it imposes one of the lowest delays on VoIP communication, a crucial detail for URLLC.

Test Sample

User calling activity strongly varies depending on the time and day of the week [55], and the rise of social networks made it even harder to analyse calling behaviour because of the decreased use of this call service [56]. Most recent studies point that when G.729 is the Codec employed by several service providers, calls have an average duration of 181 seconds with a standard deviation of 2.3 seconds [57]. Also, each base station serves an average of 750 devices, or 250 per sector taking into account LTE's 3-sector coverage base stations [58]. Because of infrastucture limitations which made not possible the execution of concurrent calls, the analysis uses a sample of 250 calls with only one call at a given time. The test results in more than 12 hours of gathered data to be analysed.

4.2.2 eMBB - Web Scenario

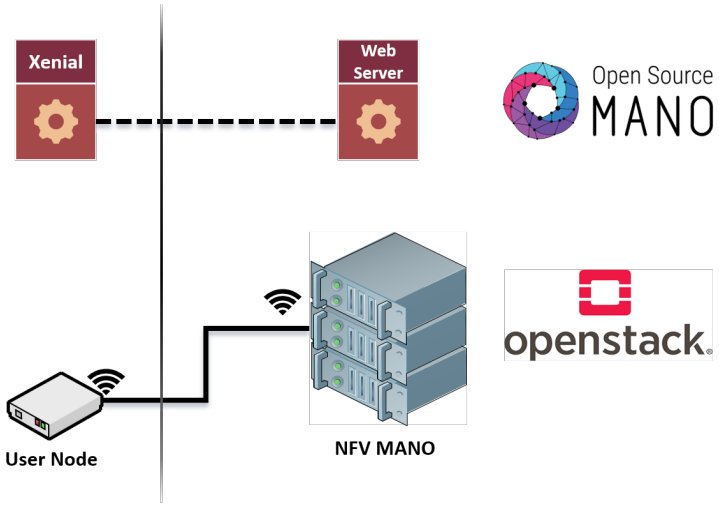

Figure 4.6: Video Server Infrastructure with Virtualised Resources above it.

For an eMBB testbed close to a real web server scenario, the proposed test implements a HTTP synthetic video streaming traffic and its components using VNFs. Figure 4.6 shows the correlation between physical infrastructure and virtualised resources. The centralised server locally runs a HTTP Server which generates synthetic video traffic. While the User Node runs an Ubuntu Xenial container responsible for requesting such data. To avoid impacting the connection performance, it implements an Apache Server with an HAProxy for load balancing and autoscaling [59]. Using a video-based test allows the evaluation of NFV efficiency on high-demanding traffic as required by an eMBB slice.

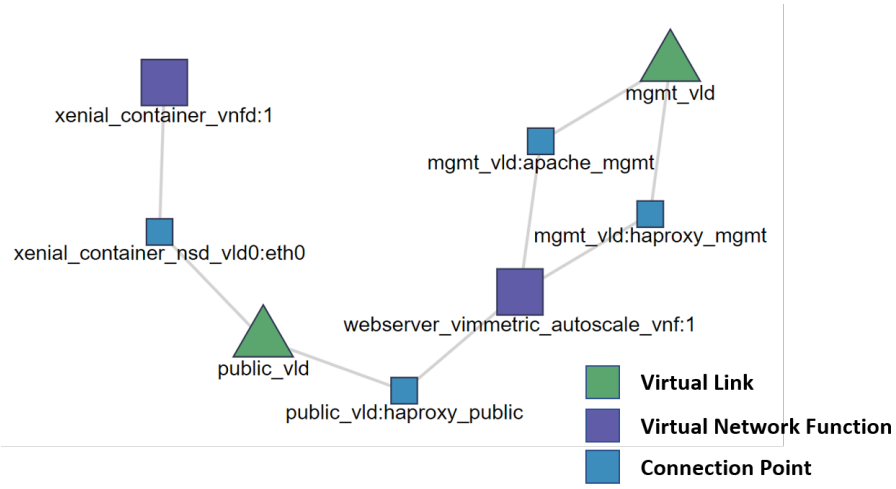

Figure 4.7: Virtual Components for the eMBB Use Case and their Virtual Links.

The Ubuntu Xenial Node is a RPi 3B+. Its VNF runs using a VM with one vCPU, 64MB of memory and 4GB of storage. The HTTP service VNF runs at the centralised server using a VM with one vCPU, 1.8GB of memory and 32GB of storage. Figure 4.7 shows all virtual links and VNFs created at OSM and connection points between them. The web server has two additional virtual links to allow the apache management and the autoscaling by the HAProxy. In a similar manner of the previous experiment, the physical connection topology does not follow the same pattern of the virtualised deployment because the NFV environment abstracts any details about the physical infrastructure.

Workload Generation

Web tests for eMBB also used synthetic workload. According to [60], video streaming corresponded for 57.69% of internet total downstream in 2018, and some researches estimate that it could reach around 82% of the Internet's total consumed traffic by 2022 [61]. Because of that, the chosen workload generates synthetic adaptive streaming data following Dynamic Adaptive Streaming over HTTP (DASH) from a centralised web server, which delivers the generated traffic through a public interface. The adaptive HTTP-based video streaming traffic follows the stochastic model proposed by [62] because of its usefulness on performance evaluation.

The test works by sending requests defined in terms of the duration of each video. The number of segments is a relation between video length and a defined segment time of 2 seconds. For each synthetic stream, the download rate of each request follows a random value defined by a Burr XII distribution with a scale of 1.469 and shape parameters of d=1.915 and c=3.014. The time between requests follows a random value of a t-distribution with location 1.932, 2.086 degrees of freedom and scale of 0.245, which corresponds to dash.js streaming client inter-request times [62].

Test Sample

The tests generate random video stream samples with a normal (Gaussian) distribution, using lengths (means) of 10 minutes (600 seconds) for 25 samples, 5 minutes (300 seconds) for 50 samples and 2.5 minutes (150 seconds) for 100 samples and a standard deviation of 5 seconds for all cases. This configuration produces groups containing between 298 and 303 segments, 148 and 153 segments and 73 and 78 segments per video stream for the first, second and third group, respectively, considering the defined segment time of 2 seconds. Such tests result in 12.5 hours of gathered data to analyse.

4.2.3 mMTC - IoT Scenario

To create an mmMTC testbed scenario close to a real use case, tests replicate IoT data transmission assuming a group of self-paced homogeneous sources. On IoT environments with a large number of devices sending data to an IoT cloud, infrastructures implement medium points called load balancers to deal with the massic traffic, with HAProxy as an appropriate implementation [63]. For this reason, the mMTC tests use the same virtual implementation for the video stream as described previously by Figure 4.7, with the NFV MANO acting as the load balancer and sending the data to the Ubuntu Xenial Node as depicted in Figure 4.8.

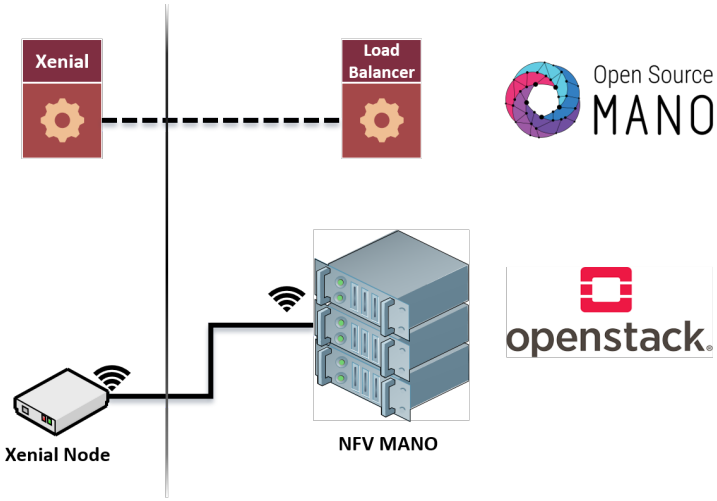

Figure 4.8: Physical Infrastructure for mMTC scenario with Virtualised Resources above.

Workload Generation

IoT traffic happens as temporised small communication messages [64] from many devices, but it also implies occasional event-based transmissions [65]. 3GPP [51] states that Poisson process can replicate a multi-user communication pattern. The tests presented here use a combination of many temporised Machine Type Communication (MTC) to mimic a Poisson process as described by [66] for large scale deployments. The workload generation works by following a periodic distribution of packets, defining the start of each theoretical device by a random value within a uniform distribution between 0 and a value T, which allows modelling an asynchronous packet transmission for the IoT devices. The same value T defines the waiting time between transmission for all devices if considering an ecosystem with temporised transmissions. For the tests, T=1 because it allows a straightforward representation of the number of packets per second for the synthetic workload.

Test Sample

Following 3GPP Study on RAN Improvements for MTC [67], a central urban scenario has a density of 4968 IoT nodes per cell. Combined with the minimum requirements for mMTC traffic [68] to provide a viable downlink quality for MTC, an average packet has 251.61 bytes. According to [69], taking into account the average active time of 8 seconds and sleep time of 241 seconds for devices with an approximate packet size of 234 bytes, each device produces an average of 145 packets during its most active period, which lasts for 10 hours. Considering that the tests have a defined waiting time of 1 second between each transmission, the necessary transmission window for the same amount of traffic is 135 seconds. The tests will run 27 samples of 135 seconds to gather approximately 1 hour of data for the aggregated information of these 4968 theoretical nodes.

4.3 vSwitch Traffic Sniffing

To sniff data using the deployed scenario, all Neutron vSwitches operate at promiscuous mode, allowing the attachment of one ObF on each compute node as previously described in Figure 3.6. This configuration enables the monitoring of VNF communication over the infrastructure. It operates on two modes depending on the monitoring execution as shown before: A Monitor Mode with simple packet sniffing and the advanced SBA Mode with recognition and perception of function calls. On the centralised server, PEvF runs as an autonomous performance evaluation function, providing data analysis and using the distributed ObFs as monitors. The centralised PEvF also runs a local ObF modified to observe the infrastructure communication. To confirm the viability of SBA Mode, an additional test with the first use case will verify each function separately. The other tests will run with all ObFs on Monitoring Mode to confirm the viability of monitoring a virtualised environment using the proposed distributed model, while a local ObF focus on evaluating the efficiency of the overall environment.

The source codes for all performance evaluation methods performed by PEvF were developed using Python 3.7 to gather the defined metrics. The ObF implementation mainly uses Pcapy, which allows an interface with libpcap to enable capturing packets over the network. Over the tests, the distributed ObFs gathered information directly from the vSwitches to avoid reading data not related to the virtualised environment. With this configuration, any influence of external communication mediums such as Wi-Fi did not interfere on gathering the data. Another ObF running alongside PEvF monitored the infrastructure traffic. This scenario allowed a direct monitoring of the whole environment.

4.4 Results

This section presents a performance evaluation of 5G's NFV using the proposed functions ObF and PEvF on URLLC, eMBB and mMTC scenarios. The testbed uses the de-facto standard software for 5G networks as described by previous sections, while the tests try to show the viability of the proposed evaluation model and also an analysis of the impact of using a virtualised infrastructure on the total network traffic — specifically, the correlation between the traffic on the virtualised layer and the effects on physical communication.

4.4.1 SBA Mode

On this experiment, the ObFs operated on SBA Mode and monitored three perspectives about implemented scenarios of the first use case as shown by Figure 4.4. This approach covers the whole system and gives a better visualisation of how a physical infrastructure reacts to virtualised functions running common tasks. The first monitoring perspective gathered both SIP Core and DNS VNFs management data sent to the NFV MANO aiming the Kamailio SIP server, comparing them to discover their influence over the infrastructure. The second perspective monitored the actual data transferred between the APs by sniffing traffic sent and received by it through RTP, this allows to better visualise how data plane communication happens, and can also show if the power-constrained environment can handle a voice call service without many hurdles. The third perspective monitored all data sent and received between the Router RPi and the server; this last test helps to evaluate if an actual service running at the virtualised layer could impact the infrastructure, only needing a comparison with data collected by the second perspective to develop this observation. The ObFs monitored all different perspectives for 4 minutes, gathering data over the same circumstances to avoid the influence of unknown conditions.

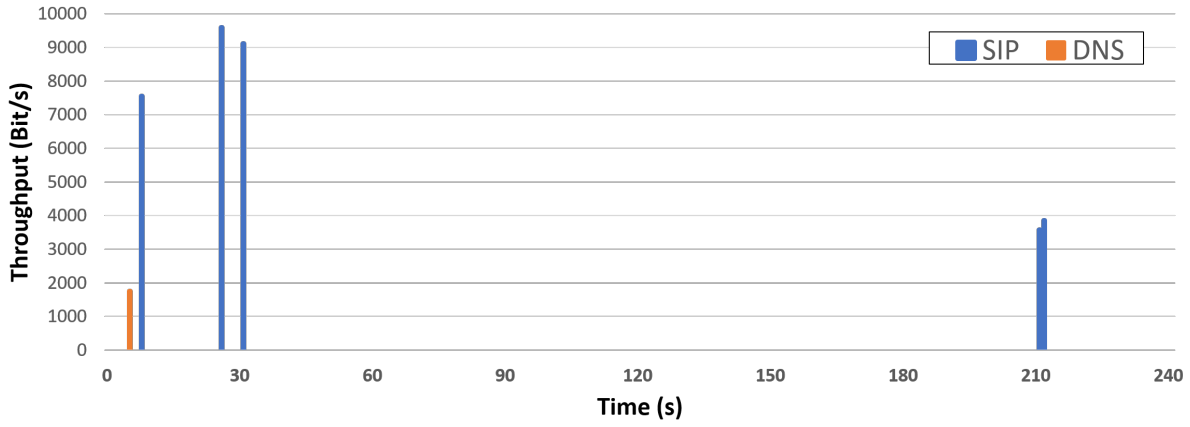

Figure 4.9: SIP and DNS throughput over the infrastructure.

Figure 4.9 shows the results of the first perspective. The DNS server only produced a single packet of just 1758 bits (around 220 bytes) at 5 seconds, which only served SIP Core request for name resolution. SIP produced only a few packets along the course of the connection, peaking at 9587 bits (around 1.20 kilobytes) at 26 seconds and sending only five messages through the whole monitoring process. The result of the first perspective alone only shows that both DNS and SIP are performing appropriately, and such small transmission packets cannot confirm any details about the performance of virtualised functions over provided infrastructure, but further analysis and comparisons with the two other perspectives can give a better view of the implementation results.

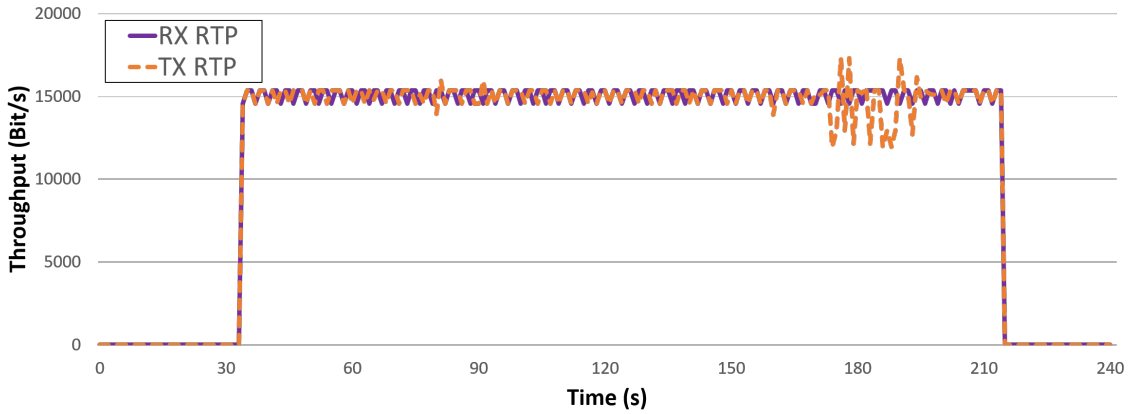

Figure 4.10: User Data Fluctuations during the Tests.

Figure 4.10 shows the results of the second perspective, with details about how user data fluctuated over the connection. TX RTP represents the data sent from one of the APs, while RTP RX represents data received by it. Between the start of the call at 34 seconds and around 171 seconds, both RX and TX kept a stable throughput between 14558 bits (1.82 kilobytes) and 15358 bits (1.92 kilobytes), with RX keeping this pace until the end of the call at 215 seconds. TX, on the other hand, heavily fluctuated its throughput between 174 seconds and 195 seconds, but it also ended the data flow around the 215 seconds mark with stable throughput. Workload generation techniques show that these variations happen when using high Standard Deviation in VoIP simulations [70], degrading VoIP QoS and allowing an acceptable quality at best [71]. Comparing the result the Figure 4.9 shows that the connections followed complementary paths, but it also shows that neither SIP nor DNS had any influence in the throughput fluctuation.

Figure 4.11 shows the results for the third perspective; which monitors NFVI control data flow between the RPi and the server. Openstack uses an AMQP message queue to communicate with all compute nodes available, implemented with RabbitMQ on the infrastructure [46]. Analysed data from test results show a little increase in the total

throughput of both channels every 60 seconds, which follows the expected temporisation defined at OpenStack for health check of virtualised infrastructure.

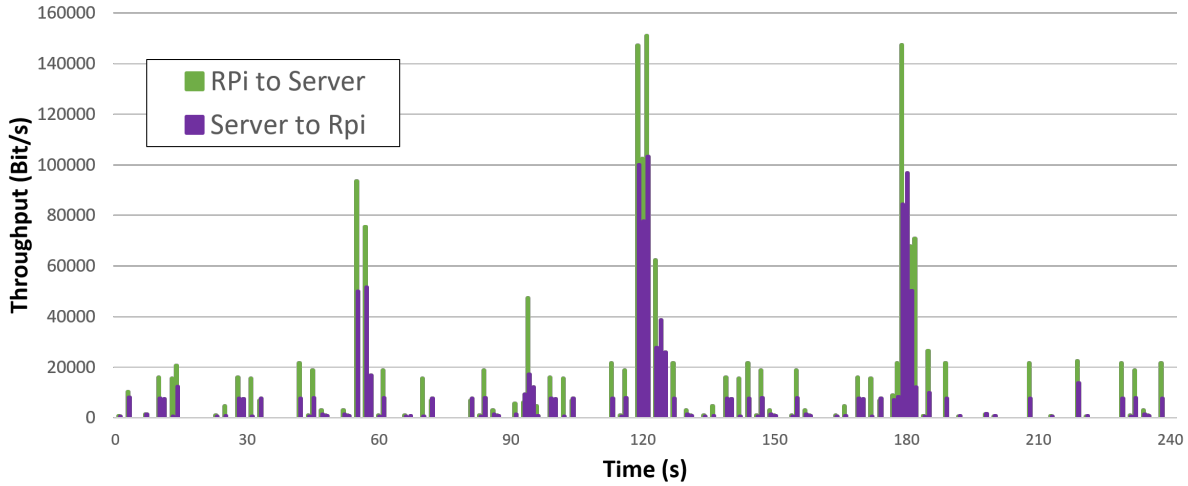

Figure 4.11: NFVI Control Data Flow during the Tests.

Data sent by the RPi had an average of 7269.46 bits/s and peaked with 151021 bits at the 121 seconds mark, while data sent by the server reached an average of 457382 bits and peaked with 103220 bits at the same 121 seconds mark. The first health check, near the 57 seconds mark, happened a little after SIP established the call and was the smallest for both channels. The second health check happened around the 119 seconds mark and had a substantial increase on both channels if with the first health check period. The last health check happened around the 208 seconds mark, right in the middle of the high throughput fluctuation perceived by TX; comparing with previous health check period, this one had a small decrease on data sent by the RPi while maintaining almost the same data rate sent by the server. During almost the whole connection, RPi kept a higher throughput than the server. Comparing the results from Figure 4.9 and Figure 4.10, it is clear that NFVI health check messages start to increase when compute nodes are performing virtualisation functions, but that it also shows a reasonably stable ratio even during high fluctuations on data throughput at the virtualised layer.

The same test was repeated a few times with similar circumstances, obtaining equivalent results and with some scenarios even generating similar fluctuations as the one observed in Figure 4.9 on TX or RX. From all collected results, data almost always followed the behaviour of a real scenario, showing that ObF implementation could allow a distributed NFV monitoring for a centralised performance evaluation service. This also shows that SBA Mode can be fully integrated into an NFV environment to monitor specific perspectives and scenarios.

4.4.2 URLLC - VoIP Scenario

Figure 4.12 shows an example of the results for a URLLC test using the same VoIP sample used on the previous test, complying with the average VoIP call defined for the tests, but only considering the 180 seconds of the VoIP call. It shows details about how user data fluctuated over the connection and the data sent and received within the infrastructure. User data represents the total actual VoIP data sent and received from one of the APs, being RTP TX and RTP RX, respectively. This data is the information sent and received by the VNFs; these VNFs perform the functional tasks of the virtualised environment and produce the traffic monitored by the distributed ObFs. Another ObF is deployed at the central server to monitor both the locally deployed VNFs and infrastructure communication. As described before, OpenStack runs a health check of virtualised infrastructure every 60 seconds; this communication tries to verify the compute nodes operation. This traffic variation is shown clearly by the picture and can heavily influence the analysis.

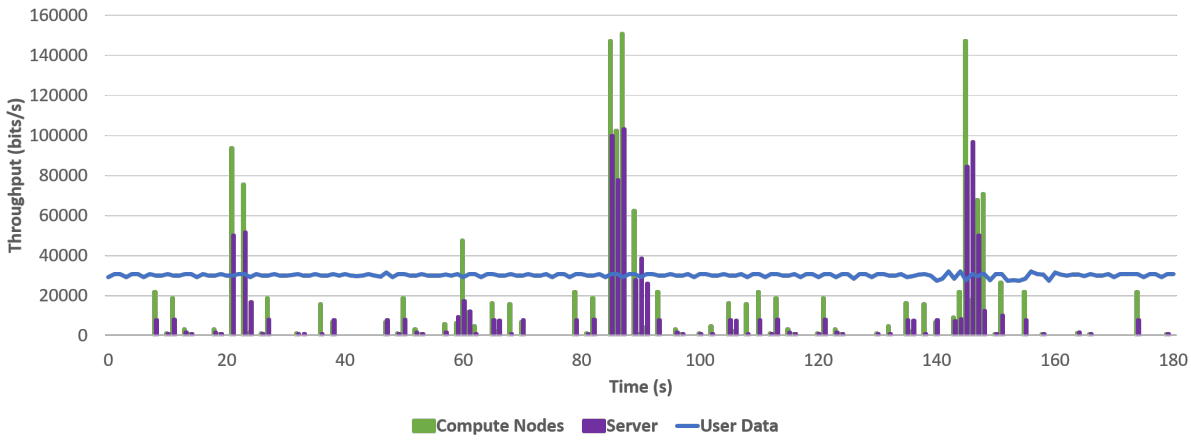

Figure 4.12: Throughput Comparison During a VoIP Call.

By monitoring the synthetic VoIP workload generated and the infrastructure behaviour, the deployed ObFs gathered 12 hours of traffic. The average throughput of actual data was 30.08 kbps, with a peak of 32.85 kbps. This throughput complies with expected bandwidth for Codec G.729 when using RTP if considering the two active channels (TX and RX). Figure 4.13 shows an example of the number of packets transferred at each moment per call. As shown by its tendency line, an average call delivered around 50.33 packets per second, with a total of 9060 packets (N.1) for a 180 seconds long call. At the same situation, it produces an average of just 72 dropped packets (N.3), accounting for only 0.795% of the total number of packets. As this work tries to verify the viability of virtualised environments, these observations do not take into account the failures generated by the Wi-Fi connection.

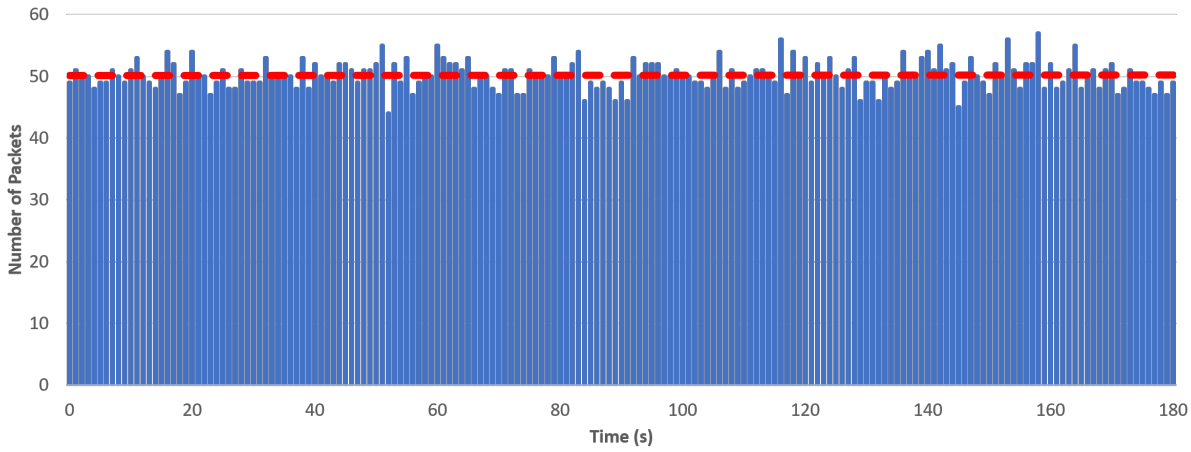

Figure 4.13: Number of Packets Transferred During a VoIP Call.

During an average call of 180 seconds, the actual data corresponded for around 680.73 KB (N.2), while the infrastructure produced 313.42 KB, which is equivalent to 31.52% (N.6) of total information transmitted within the environment. This situation shows a large amount of additional data produced to perform a VoIP call. On real-world scenarios, around 30% of additional traffic generated would seriously compromise any usefulness of NFV. To further investigate this case, it is relevant to consider OpenStack health checks because they significantly increase the amount of information transferred within the infrastructure. As shown before by Figure 4.12, during health checks, the infrastructure throughput dramatically increases. A health check lasts for an average period of 3 seconds and increases the amount of data transferred by the infrastructure by more than 600% if considering the 3 seconds window before it. When separating the communication between normal periods and health check periods, the infrastructure corresponds for an average of 14.90% of the total traffic on the first case and 80.81% on the later as shown by Figure 4.14. In such periods, the throughput of the virtualised environment for the actual data keeps a stable rate; this fact removes the possibility of a correlation between the increasing infrastructure data and user data. The AMQP message queue used by the infrastructure has a defined time between each health check [46]. For the test scenario analysed, health check data transmission happens only 5% of the total time, and even with such small windows, it considerably increased the total amount of transferred information. For future tests and to any actual deployment with virtualised environments, if the amount of additional data interferes on the communication of a different scenario, the waiting window between Openstack health checks could be increased to avoid this high additional traffic generated by the infrastructure.

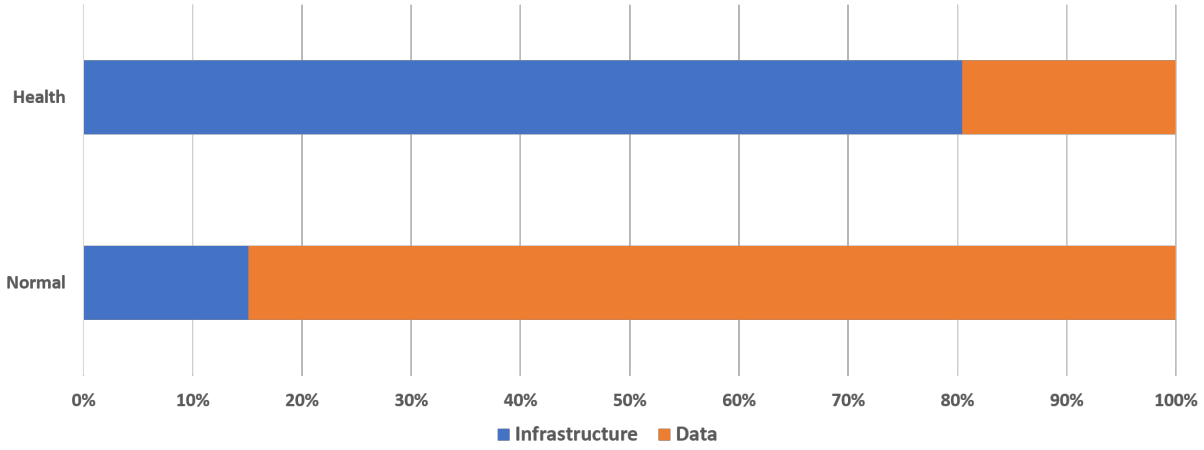

Figure 4.14: Difference Between the Amount of Data Transferred on Normal and Health Check Periods.

As shown by the results, VoIP communication reached throughputs, similar to the expected value. The number of delivered packets maintained a fair amount, and the results show a viable scenario. Table 4.1 presents shows the final results collected by the ObFs and analysed by PEvF. Packet Count (N.1) and Octet Count (N.3) are easily verified by looking at gathered data. Dropped Packet Count (N.3) required a deeper verification to ignore packets dropped between two VNFs because the used medium could impact this metric. The use of ObF attached to all virtual switches allowed to circumvent this problem by detecting the packets regardless of used transmission medium because of its direct connection if VNFs. Over this experiment, Errored Packet Count (N.4) presented a big challenge for the evaluation because Wi-Fi's redundancy checks did not allow a direct count of errored packets in comparison with NFV, which hindered the possibility of a precise count. Because of that, the analysis did not succeed in analysing this metric. PEvF was responsible for calculating both the Amount of User Data (N.5) and the Amount of Control Data (N.6) by consolidating all received information from the distributed ObFs.

Table 4.1: NFV Performance Metrics for a VoIP Call of 180 Seconds

Resource	**Code**	**Metric**	**Units**
Network	N.1	Packet Count	9060 (#)
	N.2	Octet Count	680730 (#)
	N.3	Dropped Packet Count	72(#)
	N.4	Errored Packet Count	-
	N.5	Amount of User Data	68.48 (%)
	N.6	Amount of Control Data	31.52 (%)

4.4.3 eMBB - Web Scenario

Figure 4.15 depicts an example of a video stream for an eMBB scenario using a video sample of 2.5 minutes (150 seconds), which took 95 seconds to transfer, with an average throughput of 6.32 Mbps. In the same fashion of the VoIP scenario, it shows details user data fluctuation in comparison with infrastructure communication. User data is the equivalent of video stream transferred from the central server to a dash user node. A distributed ObF monitors the information sent from the centralised server to the user node, while a centralised ObF monitors the data generated by the infrastructure. As in the previous test, OpenStack runs a health check of virtualised infrastructure every 60 seconds to verify the operation of the compute node. To facilitate the visualisation, the infrastructure throughput depicted by Figure 4.15 accounts for all the traffic generated by the messages sent by the compute node and also by the centralised server.

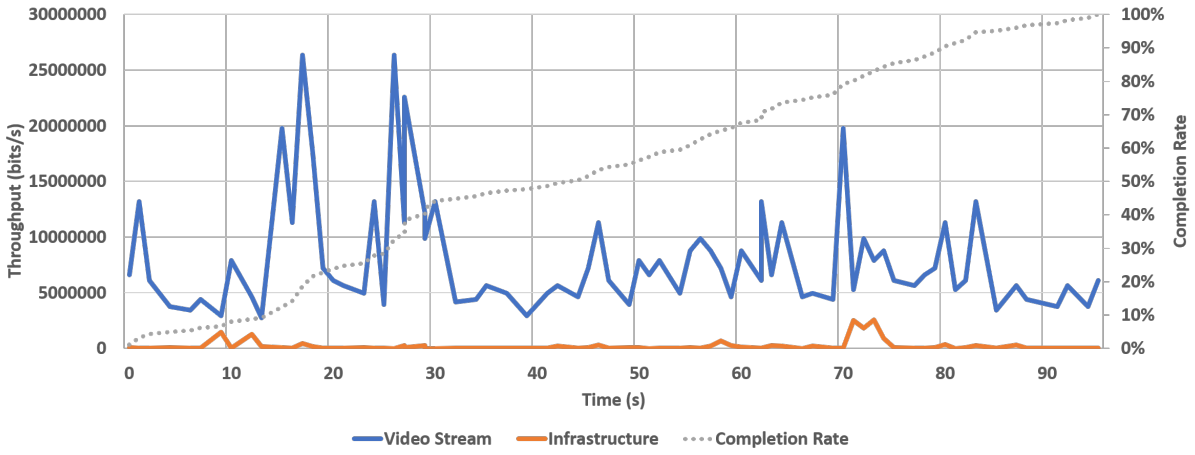

Figure 4.15: Throughput Comparison During a Video Stream.

Dash clients often download multiple segments using a single connection, but also delay a segment to avoid causing a buffer overflow [62]. Because of that, it is common to see variations on segment download times, which, with available bandwidth, also causes variations at download rates. The tests did not use a buffer to store transferred data, which allowed the transmission to happen without interruptions. Figure 4.16 shows an example of the number of segments downloaded at each moment considering the same video stream of 150 seconds, which generated 75 segments. With 45000 seconds of total analysed population, the ObF monitored 60744 packets (N.1) on this single transmission, with just 481 dropped packets (N.3) or 0.792%. With 809.91 packets per second in this sample, each packet had 1249.147 bytes on average and a standard deviation of 72.741. With 95% confidence, the mean packet size for the total population is between 1248.57 and 1249.73 bytes, which follows the expected outcome for an HTTP DASH stream [72].

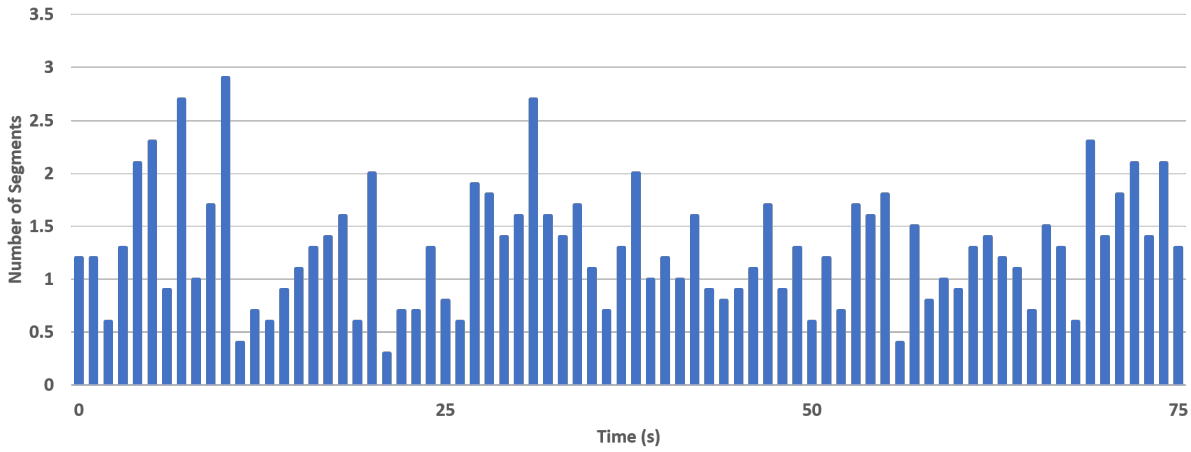

Figure 4.16: Transmission Time of Each Video Segment.

With the synthetic video workload produced and the infrastructure control messages, the ObFs gathered 12.5 hours of generated traffic. Following PEvF analysis, the average throughput of the total video stream was 8.51 Mbps, with a peak of 28.74 Mbps, which happens because DASH adapts video stream quality to download segments on different paces according to available bandwidth. Throughout the tests, it produced an average of 479.99 dropped packets per 150 long video streams, with a standard deviation of 130.37. If taking into account the total transmission of 45000 seconds, there is a total population of 300 tests, and with 95% confidence, the dropped packets mean for the total population is between 465.19 and 494.79 per transmission. This number accounts for between 0.765% and 0.814% of total packets. In the same manner of the previous test, these observations do not take into account the failures generated by the Wi-Fi connection.

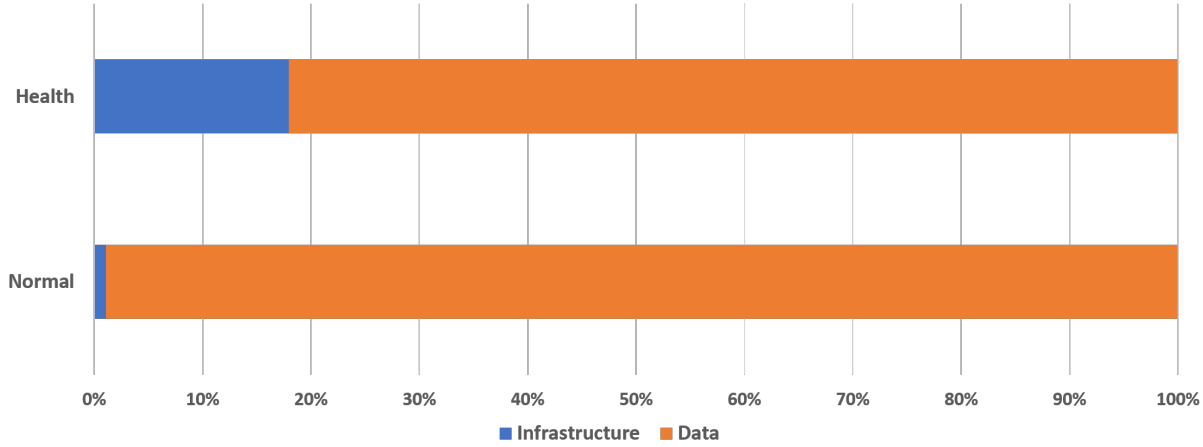

Figure 4.17: Difference Between the Amount of Data Transferred on Normal and Health Check Periods for a Video Stream.

During an average video stream of 150 seconds, the actual data corresponded for around 74.46 MB (N.2) while the infrastructure produced 1.77 MB, which is equivalent to only 2.32% (N.6) of total information transmitted within the environment. For the specific case of Figure 4.15, video stream data corresponded for 75.63 MB and the infrastructure produced only 2.11 MB, which is 2.72% of the total data. This amount represents a significant reduction when compared with the previous test for a VoIP. As shown by Figure 4.17, during health checks, the infrastructure throughput still dramatically increases. When separating the communication between normal periods and health check periods, the infrastructure corresponds for an average of 1.01% of the total traffic on the first case and 17.96% on the later. In such periods, the throughput of the virtualised environment for the actual data still keeps a stable rate; maintaining the analysis about a non-existing correlation between the increasing infrastructure data and user data. For the test scenario analysed, health check data transmission happens only 5% of the total time. For future tests and to any actual deployment with virtualised video stream environments, if the amount of additional data interferes on the communication of a different scenario, the waiting window between Openstack health checks could be increased to avoid this high additional traffic generated by the infrastructure as suggested for the previous analysis. Table 4.2 shows a compilation of the metrics collected for this use case.

Table 4.2: NFV Performance Metrics for a Video Stream of 150 Seconds

Resource	**Code**	**Metric**	**Units**
Network	N.1	Packet Count	60744 (#)
	N.2	Octet Count	75639343 (#)
	N.3	Dropped Packet Count	482 (#)
	N.4	Errored Packet Count	-
	N.5	Amount of User Data	97.68 (%)
	N.6	Amount of Control Data	2.32 (%)

4.4.4 mMTC - IoT Scenario

Figure 4.18 shows an example of a monitoring window of 135 seconds for an mMTC scenario with the workload using specifications described in Section 4.2.3. It presents the data fluctuation over the connection for both the infrastructure and the MTC communication within the virtualised environment. The MTC traffic is the equivalent of the data sent by the Load Balancer VNF at the NFV MANO to the Xenial Node VNF running at the remote compute node. A distributed ObF monitors this data and reports all information to the centralised PEvF. As in previous tests, OpenStack health check happens every 60 seconds, and as in the eMBB test, infrastructure data accounts for all the traffic generated from messages sent by the compute node and also by the centralised server.

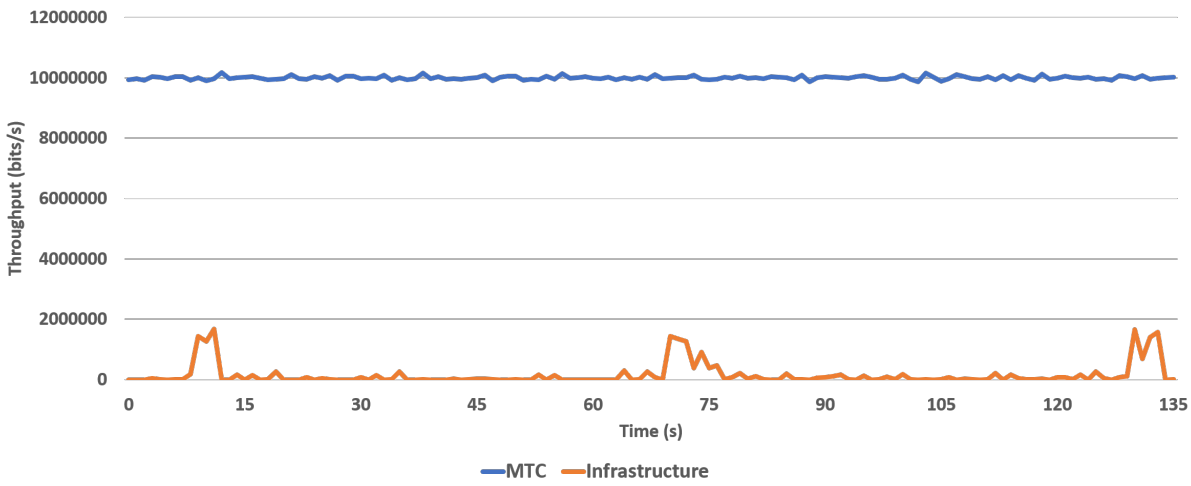

Figure 4.18: Throughput Comparison During an mMTC Workload Transmission.

On the particular sample showed by Picture 4.18, the MTC follows a steady pace with small throughput variations. The stable transmission happens because of the temporised traits of these devices [51]. With an average throughput of 10.07 Mbps, the sample generated a total of 683899 packets (N.1) on this single transmission, with just 6599 dropped packets (N.3) or 0.965%, giving an average of 5069.92 packets per second, which produced 169985155 bytes (N.2) according to ObF detection. Figure 4.19 shows the number of packets transferred at any given moment during this particular test sample. The number of packets often resembles the throughput variations because MTC usually uses standard sizes for messages exchanged between entities [72]. Given the Poisson distribution of mMTC traffic for temporised IoT communication, the number of packets often approximate the number of devices sending information at any given moment.

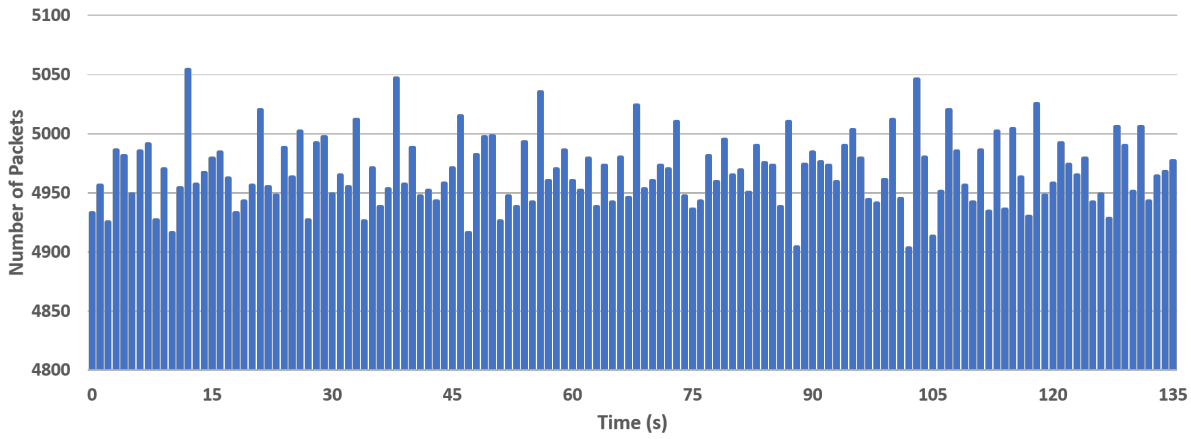

Figure 4.19: Number of Packets Transferred for an mMTC Sample of 135 seonds.

The ObFs gathered approximately 1 hour of synthetic mMTC workload and infrastructure control messages. During the complete test routines, PEvF detected an average throughput of 9.98 Mbps. With a total population of 27 tests of 135 seconds and a standard deviation of 2.9767, with 95% confidence, the mean is between 9.88 Mbps and 10.08 Mbps. During an average mMTC run of 135 seconds, the actual data corresponded for around 169.98 MB while the infrastructure produced 2.69 MB, which is equivalent to only 1.55% (N.6) of total information transmitted within the environment. When compared with previous tests, this amount represents another considerable reduction in the ratio of infrastructure data to the total data.

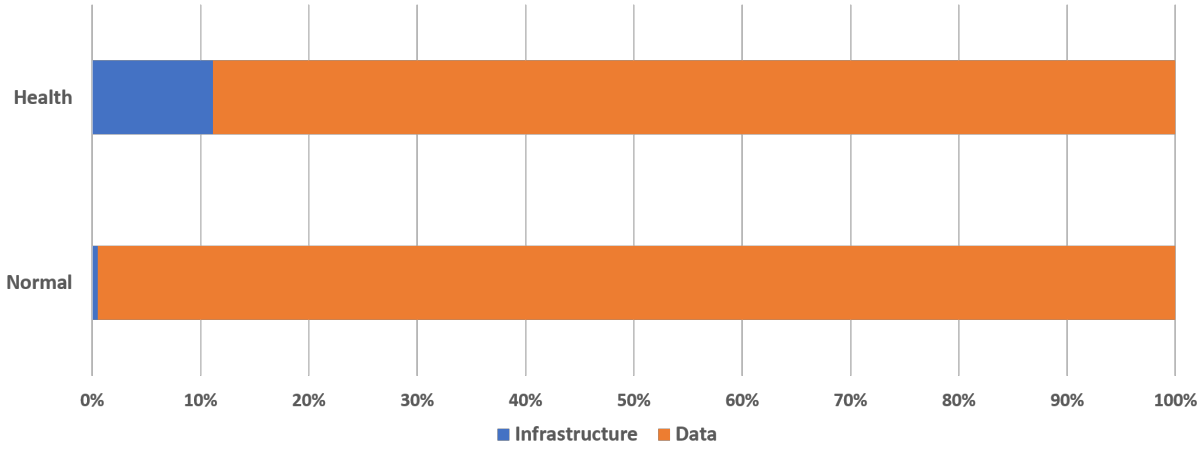

Figure 4.20: Difference Between the Amount of Data Transferred on Normal and Health Check Periods for an mMTC scenario.

As shown by Figure 4.20, during health checks, the infrastructure throughput still dramatically increases. When separating the communication between normal periods and health check periods, the infrastructure corresponds for an average of 0.52% of the total traffic on the first case and 11.16% on the later. Even in health check periods, mMTC keeps a stable rate, which compared with previous tests, strongly suggests that NFV operation does not imply heavy traffic on the overall deployment. Table 4.3 presents the final metrics collected for this use case.

Table 4.3: NFV Performance Metrics for an mMTC Transmission of 135 Seconds

Resource	**Code**	**Metric**	**Units**
Network	N.1	Packet Count	683899 (#)
	N.2	Octet Count	169985155 (#)
	N.3	Dropped Packet Count	6599 (#)
	N.4	Errored Packet Count	-
	N.5	Amount of User Data	98.45 (%)
	N.6	Amount of Control Data	1.55 (%)

4.5 Chapter Review

The tests and results presented in this chapter showed that the proposed evaluation architecture could successfully monitor and analyse the performance of an NFV environment using the standard configuration for 5G implementations. The evaluations focused on three 5G use cases and the amount of additional traffic created for each scenario. It is possible to see with collected results, that using distributed functions to collect data about the state of compute nodes and sent to a centralised entity is a feasible method to avoid imposing much stress even on resource-constrained infrastructures. Analysed information also showed that NFV is a viable technology to enable a virtualised mobile network environment. For future works, the deviations on data throughputs on URLLC use case and the behaviour of the amount of additional traffic on more extensive infrastructures are interesting open topics for further research.

Chapter 5

Conclusions and Future Work

In this work, we designed, implemented and evaluated a virtualised 5G mobile network infrastructure under different use cases by developing a performance evaluation model that address NFV environments, aiming to create a complete integration into 5G SBA. The deployment of necessary infrastructure was executed using industry and academic standards for the architecture. With the creation of such an environment, we validated both the viability of the whole 5G infrastructure on a power-constrained scenario and the use of observation functions attached to virtual switches at distributed compute nodes.

Due to the high installation and maintenance costs of mobile network infrastructures, ISPs needed to find a cost-effective and very flexible technology to allow more straightforward implementation and evolution of their networks. NFV enabled this scenario by virtualising physical components and abstracting underlying hardware for more natural resource distribution and redistribution. At the same time, it also brought to mobile networks some common problems of virtualised environments. Due to its virtualised properties, NFV suffers from many hurdles of this technological application, such as resource competition, function placement and resource degradation. Also, mobile networks imply high mobility of users between the provider's equipment, requiring that virtualised networks also deal with component replacement and migration by a reasonable speed. In addition to that, ISPs are legally obliged to fulfil user's contracted SLAs. Therefore, any infrastructural delay caused by performance degradation of virtualised environments could profoundly impact the mobile provider operation.

This work proposes a performance evaluation model integrated into 5G SBA to monitor and analyse the behaviour of this virtualised environment, aiming to anticipate and circumvent bottlenecks. As a way to demonstrate its viability on large environments, the tests focused on the efficiency of NFV environments. The evaluation model adds two new functions into the SBA, a distributed Observation Function (ObF) and a centralised Performance Evaluation Function (PEvF). ObF collects data from compute nodes, mon-

itoring their virtualised environments and reporting them to the other function. PEvF receives the data collected by all ObFs and compiles it for a complete evaluation of virtualised resources, slices and VNFs.

While some works tried to create a performance monitoring system for NFV, none has implemented a native 5G SBA integration to evaluate the performance of virtualised mobile networks focusing on the additional traffic created over a virtualised infrastructure. ObF attaches itself to virtual switches created by the NFV to accomplish active monitoring and aggregate information starting from the micro-service level up to the slicing level. In this work, the focus was the confirmation of monitoring capabilities of the aimed design. As shown by the test results; this approach resulted in a successful performance monitoring system over the distributed infrastructure and generated adequate information of the virtualised environment behaviour. After all the test analysis, this innovative approach resulted in a very efficient data gathering idea for NFV environments. The results obtained motivate the use of proposed techniques as an integrated part of 5G SBA and make it an attractive and promising platform for architecture for resource-oriented infrastructures. To the best of our knowledge, this is the first work to systematically evaluate the introduction of such performance evaluation functions into the core of virtualised mobile networks while focusing on the infrastructure network traffic analysis.

On its current state, the evaluation functions can work as active components of 5G SBA and are capable of showing how it performs under different scenarios. To better evaluate the additional network traffic created by the infrastructure, the developed functions did not consider other components for now, such as compute and memory resources. With obtained results, an NFV seems to be a viable candidate for mobile network deployment. The percentage of additional traffic created by the infrastructure tends to decrease when increasing the amount of actual data under proposed test scenarios. As it is now, the tests only considered a small number of compute nodes and VMs, which could give a decent performance for virtualised scenarios. For future experiments, the use of a more significant amount of compute nodes and VMs can show a better view of how the infrastructure creates additional traffic. Also, the use of OpenStack Nova requires an evaluation of its components to verify their timing and the necessary network resources on each element.

The implementation of this performance evaluation considered an NFV environment using the de-facto standard software for mobile network virtualisation, OpenStack and Open Source MANO, as the base of deployed infrastructure. This configuration meets the requirements of other novel 5G projects, and its adoption allows consistency with earlier works, enabling the integration of this work into current projects. It is worth noting that, although obtained results give a great perspective on how to deal with proposed improvements of 5G, future work must strictly follow 3GPP and ETSI guidelines for

function services. And to expand the evaluation even further, the analysis must evolve to consider other environments.

References

[1] 5GPPP: *View on 5G Architecture (Version 2 . 0).* (July), 2017. 1, 9, 10, 15, 24

[2] Basta, Arsany, Wolfgang Kellerer, Marco Hoffmann, Klaus Hoffmann, and Ernst Dieter Schmidt: *A virtual SDN-enabled LTE EPC architecture: A case study for S-/P-gateways functions.* Technical report, 2013, ISBN 9781479927814. 1, 15, 17

[3] Herrera, Juliver Gil and Juan Felipe Botero: *Resource Allocation in NFV: A Comprehensive Survey.* IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, 13(3), 2016. `http://www.ieee.org/publications_standards/publications/rights/index.html`. 1

[4] Condoluci, Massimo and Toktam Mahmoodi: *Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges.* Computer Networks, 146:65–84, December 2018. 1, 10, 17

[5] Laghrissi, Abdelquoddouss and Tarik Taleb: *A Survey on the Placement of Virtual Resources and Virtual Network Functions.* IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 21(2), 2019. 1, 17

[6] Cao, Lianjie, Puneet Sharma, Sonia Fahmy, and Vinay Saxena: *NFV-VITAL: A framework for characterizing the performance of virtual network functions.* 2015 IEEE Conference on Network Function Virtualization and Software Defined Network, NFV-SDN 2015, pages 93–99, 2016. 1, 17

[7] Rosa, Raphael Vicente, Claudio Bertoldo, and Christian Esteve Rothenbe: *Take Your VNF to the Gym: A Testing Framework for Automated NFV Performance Benchmarking.* 2017. 1, 17

[8] Scourias, John: *Overview of the Global System for Mobile Communications, Technical report, University of Waterloo.* Technical report, 1995. 4

[9] Mouly, M and M B Pautet: *The GSM system for mobile communications.* 1992, ISBN 2950719007. 5

[10] Cai, Jian and David J Goodman: *General Packet Radio Service in GSM.* Technical report. 5

[11] Bettstetter, Christian, Hans Jorg Vogel, and Jorg Eberspacher: *GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface.* Technical report. 5

[12] Furuskar, A., S. Mazur, F. Muller, and H. Olofsson: *EDGE: enhanced data rates for GSM and TDMA/136 evolution.* IEEE Personal Communications, 6(3):56–66, June 1999. 5

[13] Dahlman, Erik, Jens Knutsson, Fredrik Ovesjö, Magnus Persson, and Christiaan Roobol: *WCDMA-The Radio Interface for Future Mobile Multimedia Communications.* Technical Report 4, 1998. 5

[14] 3GPP: *3GPP TS 21.101 - V5.14.0 - Technical Specifications and Technical Reports for a UTRAN-based 3GPP system (Release 5).* TS 21.101 - V5.14.0, 2009. 5

[15] 3GPP: *3GPP TS 21.101 - V6.10.0 - Technical Specifications and Technical Reports for a UTRAN-based 3GPP system (Release 6).* TS 21.101, 2009. 5

[16] Popovski, Petar, Kasper Fløe Trillingsgaard, Osvaldo Simeone, and Giuseppe Durisi: *5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View.* IEEE Access 6, 2018. 6

[17] Kreutz, Diego, Fernando M.V. Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig: *Software-defined networking: A comprehensive survey.* Technical Report 1, 2015. 7

[18] Kekki, Sami, Walter Featherstone, Yonggang Fang, Pekka Kuure, Alice Li, Anurag Ranjan, Debashish Purkayastha, Feng Jiangping, Danny Frydman, Gianluca Verin, Kuo Wei Wen, Kwihoon Kim, Rohit Arora, Andy Odgers, Luis M Contreras, and Salvatore Scarpina: *ETSI White Paper: MEC in 5G networks.* ETSI White Paper No. 28, (28):1–28, 2018. 7

[19] 3GPP: *ETSI TS 123 501 - V15.3.0 - 5G; System Architecture for the 5G System.* ETSI TS 23.501 V15.3.0 Rel. 15, 0, 2018. 9, 20

[20] 3GPP: *ETSI TS 133 501 - V15.1.0 - 5G; Security architecture and procedures for 5G System.* ETSI TS 33.501 V15.1.0 Rel. 15, 2018. 9

[21] NGMN Alliance: *Description of Network Slicing Concept.* Ngmn, 1(1):7, 2016. 9

[22] Maternia, Michał, Salah Eddine El Ayoubi, Mikael Fallgren, Panagiotis Spapis, Yinan Qi, David Martín-Sacristán, Óscar Carrasco, Maria Fresia, Miquel Payaró, Martin Schubert, Jean Sébastien Bedo, and Vivek Kulkarni: *5G PPP Use Cases and Performance Evaluation Models.* 5GPPP, 2016. 10

[23] Samdanis, Konstantinos, Xavier Costa-Perez, and Vincenzo Sciancalepore: *From Network Sharing to Multi-Tenancy: The 5G Network Slice Broker.* IEEE Communications Magazine - Communications Standards Supplement, 2016. 10

[24] Mckeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner: *OpenFlow: Enabling Innovation in Campus Networks.* 2008. 11

[25] Sama, Malla Reddy, Luis M. Contreras, John Kaippallimalil, Ippei Akiyoshi, Haiyang Qian, and Hui Ni: *Software-Defined Control of the Virtualized Mobile Packet Core*. IEEE Communications Magazine, 2015. 11

[26] Sherry, Justine and Sylvia Ratnasamy: *A Survey of Enterprise Middlebox Deployments A Survey of Enterprise Middlebox Deployments Background*. Technical report, 2012. 12

[27] ETSI: *GS NFV 001 V1.1.1 - NFV: Use Cases Group Specification*. GS NFV 001 - V1.1.1, 2013. 12

[28] ETSI: *GS NFV 002 V1.2.1 - NFV: Architectural Framework*. GS NFV 002 - V1.2.1, 1:1–21, 2014. 13

[29] Rosen, Rami: *Resource Management: Linux kernel Namespaces and cgroups*. Technical report, 2013. 14

[30] Lin, Xin, Lingguang Lei, Yuewu Wang, Jiwu Jing, Kun Sun, and Quan Zhou: *A measurement study on linux container security: Attacks and countermeasures*. In *ACM International Conference Proceeding Series*, pages 418–429. Association for Computing Machinery, December 2018, ISBN 9781450365697. 14

[31] Taleb, Tarik, Miloud Bagaa, and Adlen Ksentini: *User mobility-aware Virtual Network Function placement for Virtual 5G Network Infrastructure*, volume 2015-Septe. 2015, ISBN 9781467364324. 16, 17

[32] Riera, Jordi Ferrer, Xavier Hesselbach, Eduard Escalona, Joan A. Garcia-Espin, and Eduard Grasa: *On the complex scheduling formulation of virtual network functions over optical networks*. 2014, ISBN 9781479956005. 16

[33] Rahman, M. M., Charles Despins, and Sofiene Affes: *Configuration cost vs. QoS trade-off analysis and optimization of SDR access virtualization schemes*. 2015, ISBN 9781479978991. 16

[34] Iovanna, Paola and Fabio Ubaldi: *SDN solutions for 5G transport networks*. 2015, ISBN 9781479988211. 16

[35] Rosa, Raphael Vicente, Christian Esteve Rothenberg, and Robert Szabo: *VBaaS: VNF Benchmark-as-a-Service*. 2015 Fourth European Workshop on Software Defined Networks, 2015. 17

[36] Nam, Jaehyun, Junsik Seo, and Seungwon Shin: *Probius: Automated Approach for VNF and Service Chain Analysis in Software-Defined NFV*. page 13, 2018. 17

[37] ETSI: *ETSI GR NFV-IFA 028 - V3.1.1 - Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on architecture options to support multiple administrative domains*. GR NFV-IFA 028 V3.1.1, 2018. 25, 26

[38] 3GPP: *ETSI TS 123 502 - V15.2.0 - 5G; Procedures for the 5G System*. ETSI TS 23.502 V15.2.0 Rel. 15, 2018. 27, 30

[39] Jain, Raj K: *The Art of Computer System Performance Analysis*. 1991. 31, 32

[40] ETSI: *ETSI GS NFV-TST 008 V2.4.1 - Network Functions Virtualisation (NFV) Release 2; Testing; NFVI Compute and Network Metrics Specification*. ETSI GS NFV-TST 008 V2.4.1, 2018. 32

[41] ETSI: *OSM Release FIVE - Technical Overview*. OSM White Paper, 2019. 35

[42] Sefraoui, Omar, Mohammed Aissaoui, and Mohsine Eleuldj: *OpenStack: Toward an Open-Source Solution for Cloud Computing*. Technical Report 03, 2012. 35

[43] Rizou, Stamatia, Takis Athanasoulis, Francesco Iadanza, Daniele Pavia, David Breitgand, Avi Weit, George Agapiou, David Griffin, Khoa Phan, Gino Carrozzo, Francesca Moscatelli, Ugur Acar, and David Lopez Meco: *5G-MEDIA: Programmable edge-to-cloud virtualization fabric for the 5G Media industry; D3.1 - Initial Design of the 5G-MEDIA Operations and Configuration Platform*. Technical report, WP3 - Operations and Configurations Framework, 2018. 35

[44] ETSI: *OSM White Paper - OSM VNF Onboarding Guidelines*. OSM White Paper, 2019. 35

[45] ETSI: *OSM White Paper - OSM Scope, Functionality, Operation and Integration Guidelines*. ETSI White Paper, 2019. 35

[46] Kavanagh, Alan: *OpenStack as the API framework for NFV: the benefits, and the extensions needed*. Technical report, Ericsson AB - Ericsson Review, 2015. 35, 45, 48

[47] Lavado, Gianpietro and José Miguel Guzmán: *Achieving end-to-end NFV with OpenStack and Open Source MANO*. Openstack Summit, 2018. 35

[48] ITU Recommendation G.1010: *End-user multimedia QoS categories*. 2001. 37

[49] Rahrer, Tim, Riccardo Fiandra, Steven Wright, David Allan, and David Thorne: *Triple-play Services Quality of Experience (QoE) Requirements*. Technical report, 2006. 37

[50] Phemius, Kévin and Mathieu Bouet: *Monitoring latency with OpenFlow*. In *2013 9th International Conference on Network and Service Management, CNSM 2013 and its three collocated Workshops - ICQT 2013, SVM 2013 and SETM 2013*, pages 122–125. IEEE Computer Society, 2013, ISBN 9783901882531. 37

[51] 3GPP: *3GPP TR 43.868 V12.1.0 - GERAN improvements for Machine-Type Communications (MTC)*. 3GPP TR 43.868 V12.1.0 Rel. 12, 2014. 37, 42, 53

[52] Hou, Xinli, Liang Xia, Guangyu Li, Qiuxiang Li, Lei Sun, Wang Rui, Javan Erfanian, Billy Liu, Anthony Chan, Bruno Tossou, Ana Galindo Serrano, Berna Sayrac, Georg Wannemacher, Arndt Kadelka, Andreas Frisch, Deutsche Telekom, Joachim Sachs, Dhruvin Patel, and Roberto Sabella: *Verticals URLLC Use Cases and Requirements by NGMN Alliance Contributors*. Technical report, 2019. 38

[53] Deghel, Matha, Salah Eddine Elayoubi, Patrick Brown, and Ana Galindo-Serrano: *Uplink contention-based transmission schemes for URLLC services.* In *ACM International Conference Proceeding Series*, pages 87–94. Association for Computing Machinery, March 2019, ISBN 9781450365963. 38

[54] Ma, Shengcheng, Xin Chen, Zhuo Li, and Ying Chen: *Performance Evaluation of URLLC in 5G Based on Stochastic Network Calculus.* Mobile Networks and Applications, 2019, ISSN 15728153. 38

[55] Candia, Julián, Marta C. González, Pu Wang, Timothy Schoenharl, Greg Madey, and Albert László Barabási: *Uncovering individual and collective human dynamics from mobile phone records.* Journal of Physics A: Mathematical and Theoretical, 41(22), 2008, ISSN 17518113. 39

[56] Wortham, Jenna: *Cellphones now used more for data than for calls.* New York Times, Online, Pu:2010, 2010. 39

[57] Holub, Jan, Michael Wallbaum, Noah Smith, and Hakob Avetisyan: *Analysis of the Dependency of Call Duration on the Quality of VoIP Calls.* IEEE WIRELESS COMMUNICATIONS LETTERS, 2018. 39

[58] Sauter, Martin: *From GSM to LTE-Advanced Pro and 5G: An Introduction to Mobile Networks and Mobile Broadband.* 2017. 39

[59] Diaz, Benjamin: *OSM Hackfest-Session 7 Performance & Fault Management.* Technical report, 2019. 40

[60] Sandvine: *The Global Internet Phenomena Report 2018.* Technical report, Sandvine, 2018. 41

[61] Cisco: *Cisco Visual Networking Index: Forecast and Trends, 2017–2022.* Technical report, Cisco, 2017. 41

[62] Waldmann, Silvio, Konstantin Miller, and Adam Wolisz: *Traffic model for HTTP-based adaptive streaming.* In *2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2017*, pages 683–688. Institute of Electrical and Electronics Engineers Inc., November 2017, ISBN 9781538627846. 41, 50

[63] Hou, Lu, Shaohang Zhao, Xiong Xiong, Kan Zheng, Periklis Chatzimisios, M. Shamim Hossain, and Wei Chen: *Internet of Things Cloud: Architecture and Implementation.* IEEE Communications Magazine, 54(11):32–39, December 2016, ISSN 01636804. 42

[64] Laner, Markus, Navid Nikaein, Dejan Drajić, Philipp Svoboda, Milica Popović, and Srdjan Krčo: *M2M traffic and models.* In *Machine-to-Machine Communications: Architectures, Technology, Standards, and Applications*, pages 57–86. CRC Press, January 2014, ISBN 9781466561243. 42

[65] Nikaein, Navid, Markus Laner, Kaijie Zhou, Philipp Svoboda, Dejan Drajic, Milica Popovic, and Srdjan Krco: *Simple Traffic Modeling Framework for Machine Type Communication.* Technical report. 42

[66] Hossfeld Tobias, Florian Metzger, and Poul E. Heegaard: *Traffic Modeling for Aggregated Periodic IoT Data*. 2018. 42

[67] 3GPP: *3GPP TR 37.868 V11.0.0 - Study on RAN Improvements for Machine-type Communications*. 3GPP TR 37.868 V11.0.0, 2011. 43

[68] Itu-r: *Minimum requirements related to technical performance for IMT-2020 radio interface(s) M Series Mobile, radiodetermination, amateur and related satellite services*. Technical report, 2017. 43

[69] Sivanathan, Arunan, Daniel Sherratt, Hassan Habibi Gharakheili, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman: *Characterizing and classifying IoT traffic in smart cities and campuses*. In *2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2017*, pages 559–564. Institute of Electrical and Electronics Engineers Inc., November 2017, ISBN 9781538627846. 43

[70] Mattos, Carlos Ignacio, Eduardo Parente Ribeiro, and Evelio Martín García Fern: *An unified VoIP model for workload generation*. Multimedia Tools and Applications, 2012. 45

[71] Clark, Alan: *Internet Telephony Fall 2005 VoIP Performance Management*. 2005. 45

[72] Ito, Maria Silvia, Rafael Antonello, Djamel Sadok, and Stênio Fernandes: *Network level characterization of adaptive streaming over HTTP applications*. In *Proceedings - International Symposium on Computers and Communications*. Institute of Electrical and Electronics Engineers Inc., September 2014, ISBN 9781479942787. 50, 53

Appendix A

JSON Example Structure for Compute Node Mapping

```
[
  {
    "machine": "machine 1",
    "cpu": 99,
    "memory": 99,
    "drives": [
      {
        "drive": "d1",
        "size": "99"
      }
    ],
    "vms": [
      {
        "vm": "vm1",
        "cpu": 11,
        "memory": 11,
        "drives": [
            {
              "vdrive": "vd1",
              "size": 22
            }
        ],
        "vnfs": [
            {
              "vnf": "vnf 1"
            },
```

```
            {
              "vnf": "vnf 2"
            }
        ]
      }
    ],
    "vswitches": [
      {
        "vswitch": "v1",
        "vnfs": [
            "vnf 1",
            "vnf 2"
        ]
      }
    ]
  }
]
```