An efficient model for latency optimisation for 5G-enabled IoT applications in smart farming

An efficient model for latency optimisation for 5G-enabled IoT applications in smart farming

by

Ntshuxeko Makondo

(216081692)

Dissertation submitted in partial fulfilment of the requirements for the degree

MASTER OF COMPUTING: INFORMATION TECHNOLOGY

in the

Department of Information Technology

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

TSHWANE UNIVERSITY OF TECHNOLOGY

Supervisor: Prof. T.E. Mathonsi

Co-supervisors: Dr. H.I. Kobo and Dr. D.P. duPlessis

DECLARATION

I, **Ntshuxeko Makondo**, affirm that the dissertation titled "An efficient model for latency optimisation for 5G-enabled IoT monitoring applications in smart farming", submitted for the Master of Computing: Information Technology degree at Tshwane University of Technology, is solely my own work. It has not been previously submitted to any other institution of higher education. I confirm that all sources cited or quoted within the dissertation are appropriately acknowledged through comprehensive reference.

Signature……. Date…….

DEDICATION

This work is dedicated to my late aunt, Mthavini, my grandmothers, Mamayila (late) and Khubani. Lastly, this work is dedicated to my son, Andziso, and all my family members for their support and encouragement to continue my studies.

ACKNOWLEDGEMENTS

First, I would like to thank the Almighty God, who gave me the power and guidance to finish this research study. I also wish to express my heartfelt gratitude for the kindness of my supervisors, who contributed to the completion of this research study (Supervisor: Prof. T.E. Mathonsi; Co-supervisors: Dr. H.I. Kobo and Dr. D.P. duPlessis). These mentors continued to give me professional advice such as focusing on the research topic; and provided me with valuable feedback on my research study. Special thanks to Lusani Mamushiane and Thoriso Makhosa for their significant contributions to this research study. I also thank the CSIR and NGEI cluster for their financial support. Thanks to everyone at CNAS who has provided wonderful support and input.

Finally, I would like to express my gratitude to my entire family for their willing support and drive throughout my studies.

PUBLICATIONS

Parts of this dissertation have been submitted, presented, and published in the following conference proceedings.

Conference proceedings:

1. Ntshuxeko Makondo, Hlabishi I. Kobo, and Topside E. Mathonsi, “The Latest Developments in Software Defined Networking: Adoption rate and challenges”. In the proceedings of the 2023 IEEE AFRICON, Nairobi, Kenya, 20 – 22 September 2023, ISBN:979-8-3503-3621-4, pages 1 - 6. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10293567
2. Ntshuxeko Makondo, Hlabishi I. Kobo, Lusani Mamushiane, and Topside E. Mathonsi, “Edge Computing in 5G-enabled IoT for Agricultural applications: Opportunities and challenges”. In the proceedings of the 3rd International Conference on Electrical, Computer and Energy Technologies (ICECET 2023), Cape Town, South Africa, 16-17 November 2023, ISBN: 979-8-3503-2781-6, pages 1-6. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10389349
3. Ntshuxeko Makondo, Hlabishi Kobo, Topside E. Mathonsi, Deon Du Plessis, Lusani Mamushiane, and Thoriso M. Makhosa, “An efficient model for latency optimisation in 5G using Edge Computing for uRLLC use cases”. Accepted for publication to the 2024 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD2024), 1-2 August 2024, Mauritius.

Journals:

1. Ntshuxeko Makondo, Hlabishi Kobo, Topside E. Mathonsi, Deon Du Plessis, " Implementing an Efficient Model for Latency Optimisation in Smart Farming" IEEE Access (Submitted for review).

ABSTRACT

In various industries, including agriculture, the application of the fifth generation of wireless technology (5G) has led to significant progress. One of the most intriguing aspects of 5G technology is the potential to reduce latency for internet of things (IoT) applications, especially those essential for latency-sensitive smart farming applications. The data generated by IoT devices such as sensors, cameras, and actuators in intelligent farming applications are growing exponentially. Traditional methods of processing and storing IoT data usually include cloud data centres, which are often far from data sources, resulting in multiple network hops that increase latency. To this end, the existing network infrastructures struggle to cope with increasing traffic and to meet the stringent low-latency requirements of various IoT applications. To solve this problem, edge computing has emerged as a solution. Edge technology allows the deployment of 5G core (5GC) network functions close to IoT sensors. This method allows data processing to be performed near the sensor, thereby reducing latency. As a result, this study has proposed an efficient model to minimise latency in 5G networks by moving the user plane function (UPF) node to the edge of the network closer to users by means of the control and user-plane-separation (CUPS) strategy. Furthermore, this study proposed software-defined networking (SDN)-based backhaul. This backhaul was configured to use the open network operating system (ONOS) controller, which has been customised with a distributed core to improve throughput, latency, and scalability. Using SDN in the 5G backhaul network allows operators to create dynamic, scalable, and efficient networks capable of serving a diverse variety of services and applications with varying performance needs. The results of the experiment conducted on the third-generation partnership project (3GPP)-compliant 5G testbed demonstrate that the proposed model reduced the average round-time trip (RTT) by 60.7%, thereby improving the throughput by approximately 40.48%.

Keywords- Throughput, 5G, Latency, Cloud Computing, Edge UPF, Cloud UPF, Smart Farming, SDN, CUPS, IoT.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

1G	First Generation Wireless Technology
2G	Second Generation Wireless Technology
3G	Third Generation Wireless Technology
3GPP	3rd Generation Partnership Project
4G	Fourth Generation Wireless Technology
5G	Fifth Generation Wireless Technology
5GC	5G Core
5GNR	5G New Radio
ACT	Algorithm Classifier Tool
AF	Application Function
AMF	Access and Mobility Management Function
AR	Augmented Reality
ARC	Agricultural Research Council
AUPF	Anchor UPF
AUSF	Authentication Server Function
AWS	Amazon Web Services
BBO	Biogeography Based Optimization
BBU	Baseband Processing Unit
BPF/XDP	Berkley Packet Filter/eXpress Data Path
BSS	Business Support System

CC	Cloud Computing
CI	Cloud Infrastructure
CP	Control Plane
CPU	Central Processing Unit
CSIR	Council for Scientific and Industrial Research
CSP	Cloud Service Provider
CUPS	Control and User Plane Separation
DN	Data Network
DPDK	Data Plane Development Kit
DT	Digital Twin
DU	Distributed Unit
EC	Edge Computing
EGT	Evolutionary Game Theory
eMBB	Enhanced Mobile Broadband
EN	Edge Nodes
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FAO	Food and Agriculture Organization
FTTH	Fibre-to-the-Home
GA	Genetic Algorithm
GML	Geography Markup Language

GPRS	General Packet Radio Service
GPS	Global Positioning System
GTP	GPRS Tunnelling Protocol
GWO	Grey Wolf Optimization
HD	High Definition
HLPA	Hybrid Latency and Power-aware Approach
ICT	Information and Communications Technology
ILP	Integer Linear Programming
IMT-2020	International Mobile Telecommunications-2020
IoT	Internet of Things
IP	Internet Protocol
ITU	International Telecommunication Union
LAN	Local Area Network
LPWAN	Low-Power Wide-Area Network
LTE	Long Term Evolution
MEC	Multi-Access Edge Computing
MiMRA	Mixed-numerology Mini-slot-based Resource Allocation
mMTC	Massive Machine Type Communication
MTN	Mobile Telephone Network
MTU	Maximum Transmission Unit
NB	Narrowband

NF	Network Function
NFV	Network Function Virtualization
NI	Network Infrastructure
NIC	Network Interface Card
NSA	NonStandalone
OLT	Optical Line Terminal
ONOS	Open Network Operating System
OSS	Operation Support System
OST	Stopping Theory
PCF	Policy Control Function
PCIe	Peripheral Component Interconnect Express
PDU	Packet Data Unit
PFCP	Packet Forwarding Control Protocol
PGW	Packet Gateway
PtMP	Point-to-Multipoint
PtP	Point-to-Point
QoS	Quality of Service
RAN	Radio Access Network
RF	Radio Frequency
RRU	Remote Radio Unit
RTT	Round Time Trip

SA	Standalone
SANReN	South African Research Network
SBA	Service Based Architecture
SDN	Software Defined Networking
SDR	Software Defined Radio
SF	Smart Farming
SGW	Serving Gateway
SISO	Single-Input Single-Output
SmartNIC	Smart Network Interface Card
SMF	Session Management Function
SPS	Semi-Persistent Scheduling
TIA	Technology Innovation Agency
UDM	Unified Data Management
UE	User Equipment
UP	User Plane
UPCR	User Plane Function Placement and Chaining Reconfiguration
UPF	User Plane Function
uRLLC	ultra Reliable and Low Latency Communications
USRP	Universal Software Radio
vEPC	Virtual Evolved Packet Core
VLAN	Virtual Local Area Network

VM	Virtual Machine
VNF	Virtual Network Functions
VR	Virtual Reality
WAN	Wide Area Network

CHAPTER 1 INTRODUCTION

1.1 Background

The agricultural sector has recently entered the fourth industrial revolution (4IR) in the adoption of information and communications technologies (ICT) for conventional farming practices (Makondo, Kobo, Mathonsi, et al., 2023). As a result of this evolution, a new concept, known as smart farming, has emerged (Berckmans, 2017). The arrival of smart farming represents a ground-breaking concept in which ICT plays an important role in improving agricultural practices. Smart farming uses ICT technologies to optimise harvesting processes; and uses advanced tools to efficiently monitor and track livestock, representing a huge step forward in agriculture innovation (Veissier et al., 2019). Farmers gain the ability to consistently monitor the welfare of their livestock in real-time (Makondo, Kobo, Mathonsi, et al., 2023). This enhanced monitoring facilitates early detection of potential hazards that could pose catastrophic risks to livestock, such as fire outbreaks (Perakis et al., 2020). Cutting-edge technologies such as the internet of things (IoT) and fifth generation mobile technology (5G) hold the potential to revolutionize agriculture (Saiz-Rubio & Rovira-Más, 2020).

The IoT stands as a cutting-edge wireless technology that facilitates seamless communication among objects across a broad spectrum, leveraging the current internet communication infrastructure (Sadeeq et al., 2021). At its foundation, the concept relies on enabling interaction between many physical objects or devices that are linked together via the internet. In addition, IoT technology is used in many industries today, including mining, transport, agriculture, healthcare, and entertainment. However, delivering low-latency connections for IoT applications to enable real-time communication can be challenging (Laghari et al., 2021). As demand for low-latency connectivity grows, so does the rise of latency-sensitive industrial IoT applications, making 5G a potential alternative to addressing these latency requirements (Slalmi et al., 2021).

5G, the most recent generation of wireless mobile technology, promises to significantly improve on the performance of its predecessor, long term evolution (LTE) networks (Hassan et al., 2019). Standardisation of the 5G network is part of the third-generation partnership project (3GPP); and the specifications of its requirement models are described in the framework of the international telecommunications union (ITU) in international mobile

telecommunications 2020 (IMT-2020) (Ateya et al., 2018). The ITU and 3GPP have classified 5G services as ultra-reliable low-latency communication (uRLLC), massive machine-type communication (mMTC), and enhanced mobile broadband (eMBB) (Makondo, Kobo, Mathonsi, et al., 2023). eMBB consumers need Gbps-level data speeds, while mMTC users prioritise connection density and energy efficiency; and uRLLC traffic prioritises high dependability (99.999%) and low latency (1~20 ms/packet) (Huawei, 2016). Furthermore, the 5G uRLLC is designed for services that must meet stringent latency and availability standards. Thus 5G mobile networks that enable uRLLC must have low latency, with minimal packet loss or packets arriving out of sequence (Hamidi-Sepehr et al., 2021).

Previously, 5G core (5GC) network functions were deployed as proprietary software running on dedicated hardware in remote data centres (Ghosh et al., 2019). This type of implementation had numerous limitations, including excessive costs, overuse of resources, and complex deployment and maintenance. Such required the adoption of a cloud-based 5GC network to overcome the drawbacks of the old architecture (Al-Quzweeni et al., 2019).

Cloud-based 5GC network deployments provide numerous advantages, including improved scalability, optimised resource utilisation, deployment flexibility, and agility (Ghosh et al., 2019). This aligns with the 3GPP specifications of the 5GC that aim to provide high-performance, flexible, and on-demand services (Dogra et al., 2020). Adopting cloud-based 5GC network architectures enables network operators to reduce costs, build networks more efficiently, introduce new services more rapidly, and respond to changing technology trends through virtualisation of 5G network functions (Al-Quzweeni et al., 2019). The 5G architecture has since then embraced network function virtualisation (NFV) and cloud computing (CC) to streamline network and service deployment, operations, and management in order to fulfil the requirements for scale, throughput, latency, and reliability (Ali et al., 2021).

Although cloud-based 5GC network deployments overcome the known limitations of deploying core network functions as proprietary software running on dedicated hardware in remote data centres, there are many unsolved problems such as network congestion, latency, throughput, and bandwidth that still need to be optimised (Tang et al., 2021). As the number of applications of the IoT enabled by latency sensitive 5G increases, the centralised cloud becomes inefficient. This is because all queries from IoT devices are routed to a central cloud server, which is often located far away from the data source, resulting in several network hops,

which increases latency (Al-Quzweeni et al., 2019). As a result, uRLLC requirements become difficult to achieve. Therefore, computing capabilities must be at the edge, closer to the source where data are being generated, so that the stringent latency requirements for 5G-enabled IoT applications can be realised (Cao et al., 2020).

In recent years, several computing technologies have been proposed to work at the network edge; these technologies include edge/fog/mobile computing (Sadeeq et al., 2021). Although they exist under different names, they all serve as a middle layer between IoT devices and a centralised cloud to provide storage and computational resources with minimal delay (L. Lin et al., 2019). Edge computing (EC) then emerged in response to the growing demand for real-time data processing and the growing number of latency-sensitive 5G-enabled IoT applications (Deng et al., 2020). EC is a distributed computing platform that brings computing applications and resources closer to data sources such as IoT devices (L. Lin et al., 2019).

Therefore, the core network functions of 5G such as the user control plane (UPF) can be offloaded to the edge. The UPF manages data-traffic processing and routing inside the 5GC network (Kundel et al., 2022). The deployment of UPF at the edge is expected to meet the demanding requirements of uRLLC for 5G-enabled IoT applications (Y. Li et al., 2021). As an intermediary for data packets between user equipment (UE) and the external network, UPF becomes a crucial feature of the 5GC network (Ghosh et al., 2019), with its placement and set-up significantly influencing network speed, latency, and overall user experience (Yan et al., 2019). Therefore, there is significant interest in shifting UPF from the centralised 5GC network to distributed edge servers to ease these limits and leverage the potential benefits of edge computing (Chen & Ran, 2019).

The application of edge in 5G and IoT is also receiving much attention, especially because of its vital role in latency-sensitive applications (Hossein Motlagh et al., 2020). Edge UPF is regarded as a potential solution to overcoming some of the challenges that plague the application of CC in 5G while meeting the demands of IoT (Navarro et al., 2020). With the edge UPF, the sensor nodes would be able to meet the latency requirements while sending data for processing to the distributed edge servers along the edge network.

The primary objective of this study is to develop an efficient model for offloading UPF from centralised cloud 5GC to the edge of network, with the objective of improving overall quality of service (QoS). The idea is to overcome the issues of latency, network efficiency, and

scalability by placing the UPF on the edge of the network, closer to IoT devices. The proposed model will be tested using a 5G testbed compliant with 3GPP standards. Furthermore, this study proposes the software-defined networking (SDN)-based backhaul with an open network operating system (ONOS) controller for efficient traffic management between the 5G new radio (5GNR) and 5GC network.

1.2 Motivation

This research is motivated by the growing importance of 5G-enabled IoT applications in the field of smart farming. Smart farming has developed as a transformational agricultural technique, with the ability to increase production, minimise resource consumption, and ensure sustainable farming practices. The combination of 5G technology and IoT applications creates new opportunities for real-time monitoring, control, and analysis in smart farming systems.

However, latency optimisation is a substantial obstacle to achieving all the benefits of 5G networks. As a result, an effective model is required to migrate the UPF from the centralised 5GC network to edge servers. This shift allows edge computing and 5G networks to operate with the low network latency required for real-time IoT applications. The two entities can collaborate to improve the reliability of latency-sensitive applications such as smart farming, augmented reality, virtual reality, video and voice analytics, and video security monitoring, among many others (Navarro et al., 2020).

1.3 Problem Statement

The emergence of cloud-based 5GC networks has revolutionized network architecture, offering significant advantages such as scalability, resource optimisation, and deployment flexibility (Ghosh et al., 2019). This aligns perfectly with the 3GPP specifications that emphasize high-performance, flexible, and on-demand services for the 5G era (Dogra et al., 2020). However, the centralized nature of cloud-based 5GC deployments introduces a critical challenge namely increased latency (Motlagh et al., 2020).

This research investigates the latency bottleneck inherent in centralized cloud-based 5GC architectures for real-time applications in smart agriculture. While 5G offers immense potential for precision agriculture through real-time monitoring (Dogra et al., 2020), the distance between data generating IoT devices (e.g., soil moisture sensors) and centralized cloud processing introduces significant delays (Figure 1.1). These delays hinder the efficiency and

reliability of critical agricultural operations, such as real-time fire detection and perimeter security in livestock farms.

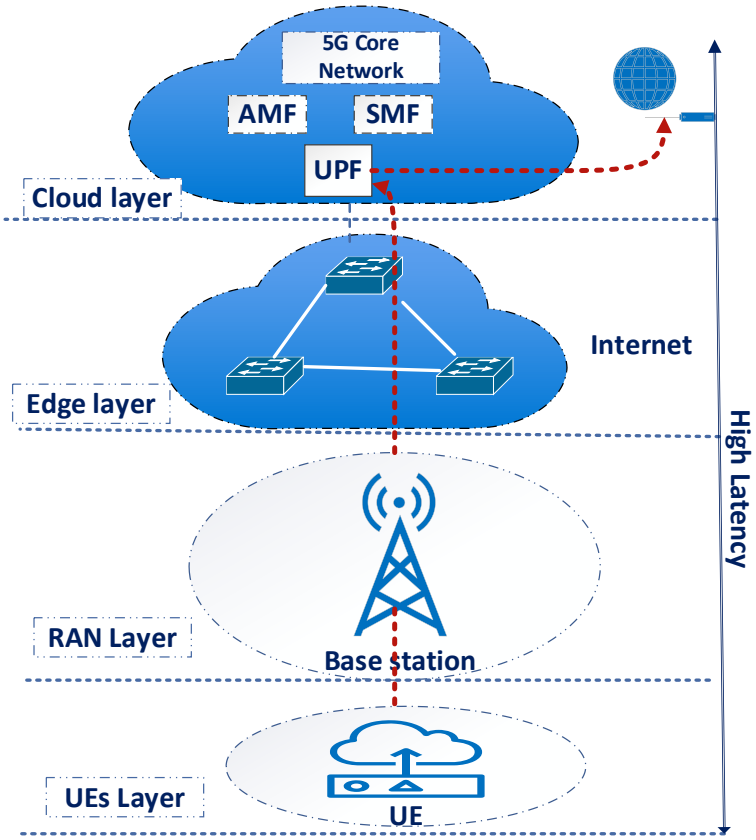

Figure 1.1. Cloud-based 5G-enabled IoT networks

In this study, we consider a livestock farm that has a 5G-enabled IoT system for monitoring perimeter fences. The system uses IoT sensors, such as distributed smoke detectors, throughout the farm to gather and send data to the internet via UPF, which is in the centralised cloud 5GC network, for immediate processing and decision-making which is paramount in smart agriculture. For instance, delayed fire detection due to high latency can have catastrophic consequences, leading to property damage, animal loss, and jeopardizing human safety. Therefore, addressing latency becomes an imperative for the successful implementation of 5G-enabled smart agriculture solutions. This study aims to develop an efficient approach that mitigates the latency challenge in the current 5G network architecture for smart agriculture applications. The proposed solution focuses on reducing the data travel distance by processing data closer to the source, at the edge of the network, where IoT sensors reside (Tang et al., 2021).

By overcoming the latency barrier, this research seeks to unlock the full potential of 5G for real-time applications in smart agriculture. This will contribute to improved efficiency, reliability, and ultimately, the success of precision agriculture practices.

1.4 Research Questions

The main question of this research study is as follows: How can an efficient model be developed which reduces latency for 5G-enabled IoT applications in smart farming?

Sub-questions.

1.4.1 What are the factors that affect the latency for 5G-enabled IoT applications in smart farming?

1.4.2 How can an efficient model be designed leveraging the edge computing to minimise the latency for 5G-enabled IoT applications in smart farming?

1.4.3 How can the 5GC and the 5GNR be integrated using an SDN-based backhaul network to optimise the performance of 5G-enabled IoT networks?

1.4.4 How can the proposed model be implemented by offloading the UPF to the edge of the network?

1.4.5 What is the effectiveness of the proposed model for 5G-enabled IoT applications in smart farming?

1.5 Research Objectives

The aim of this research study is to develop an efficient model for latency-sensitive 5G-enabled IoT applications in a smart farming environment.

1.5.1 To identify the factors that affect the latency in 5G-enabled IoT networks.

1.5.2 To design the proposed model by leveraging the edge computing, thus minimising latency for 5G-enabled IoT applications in smart farming.

1.5.3 To integrate the 5GC and the 5GNR using an SDN-based backhaul network.

1.5.4 To implement the proposed model by offloading the UPF to the edge of the network.

1.5.5 To demonstrate through experimental evaluations that offloading the UPF to the edge does reduce latency for 5G-enabled IoT applications in smart farming.

1.6 Research Methodology

This research study will use experiential methodology to achieve the overall objective of the study, which will be divided into the methods given below. Experiential methodology has been chosen for this research study due to its effectiveness in providing practical insight into solutions to real-world problems.

1.6.1 Literature review

A comprehensive literature of related work was conducted to gain more understanding of what other researchers have done to reduce latency for 5G-enabled IoT applications in smart farming; this includes conference papers, internet sources, journal articles and more.

1.6.2 Modelling

In this study, a mathematical model was used to design the proposed model. The research study demonstrated through experimental evaluations how the proposed model reduces latency for 5G-enabled IoT applications in smart farming.

1.6.3 Implementation

In this work, the proposed model was developed and implemented on a 3GPP-compliant testbed, leveraging open-source tools such as Open5GS and srsRAN, as well as an SDN-based backhaul with an ONOS controller. Furthermore, a Python code was written to design the SDN backhaul.

1.6.4 Experimentation

The experiments of the proposed model were conducted on a 3GPP-compliant 5G testbed. This testbed provided a realistic environment for evaluating the performance of the proposed model in reducing latency for 5G-enabled IoT applications in smart farming.

1.7 Significance and Benefits of the Study

This study has several significant implications and benefits. The following are some of the areas in which this research study can benefit mobile operators.

1.7.1 Deployment of UPF on the edge

The deploying of UPF on the edge provides considerable benefits and is a viable solution for mobile carriers. Mobile operators can gain various benefits by deploying the UPF at the network edge, including lower latency, higher network performance, and a better user experience. This technique offers faster data processing and more effective routing, allowing mobile carriers to provide high-speed connectivity while also supporting current and future uRLLC applications.

1.7.2 Deployment of SDN-based backhaul

Incorporating SDN technology into the backhaul infrastructure improves the operational efficiency and flexibility of mobile networks. SDN transforms network administration by providing centralised control and programmability for network devices. SDN enables operators to dynamically allocate bandwidth, optimise routing paths, and prioritise traffic according to real-time demand and network conditions. SDN-based backhaul solutions give operators greater flexibility in responding to changing traffic patterns and service requirements. By centralising administration and control, operators can quickly alter network settings, ensuring optimal resource utilisation and service delivery. This mobility is especially useful in dynamic contexts, such as fluctuating traffic loads or new service needs.

1.8 Limitations of the Study

The proposed model focuses on the optimisation of propagation latency, leaving out aspects such as processing and queueing delays, which are equally important contributors to overall latency. Furthermore, the experiment uses an emulated SANReN network, which may not fully

replicate real-world network situations. This constraint suggests that the conclusions from this network set-up may not be immediately applicable to real-world network situations. Furthermore, the unavailability of 10 Gb/s interfaces for the servers poses a constraint on the study, 10 Gb/s traffic from the backhaul being throttled to just 1 Gb/s on the servers.

1.9 Contribution of the Study

The main contributions of this research study can be summarised as follows:

1.9.1 In this study, an extensive review of the current state of the art is provided, focusing on the factors that impact latency in 5G networks in the context of smart farming. This involves evaluating the advantages and limitations of different models proposed to handle the positioning of 5GC network functions.

1.9.2 Building upon the insights gained from the literature review, this study develops an efficient model which will reduce latency for 5G-enabled IoT applications in smart farming. This model is developed by offloading the UPF to the edge.

1.9.3 This study is primarily intended for emerging markets, with a particular emphasis on South Africa, where poor IoT performance due to excessive latency is common. Therefore, the present study shows the practicality of offloading the UPF at the network edge.

1.10 Ethical Considerations

This study did not require ethics approval. This is because the data that was used in the study was based on computer programmes and experiments that neither directly nor indirectly involve any information about humans or animals.

1.11 Dissertation Organisation

The remainder of this dissertation is structured as follows: Chapter 2 presents an overview of the enabling technologies of smart agriculture and examines various related works. Chapter 3 presents the proposed model for reducing latency in 5G. Chapter 4 describes the detailed steps for the implementation of the proposed model on a 5G testbed. The results and analysis of the proposed model are evaluated and presented in Chapter 5. Lastly, the conclusion and future direction of the study are presented in Chapter 6, as indicated in Figure 1.2.

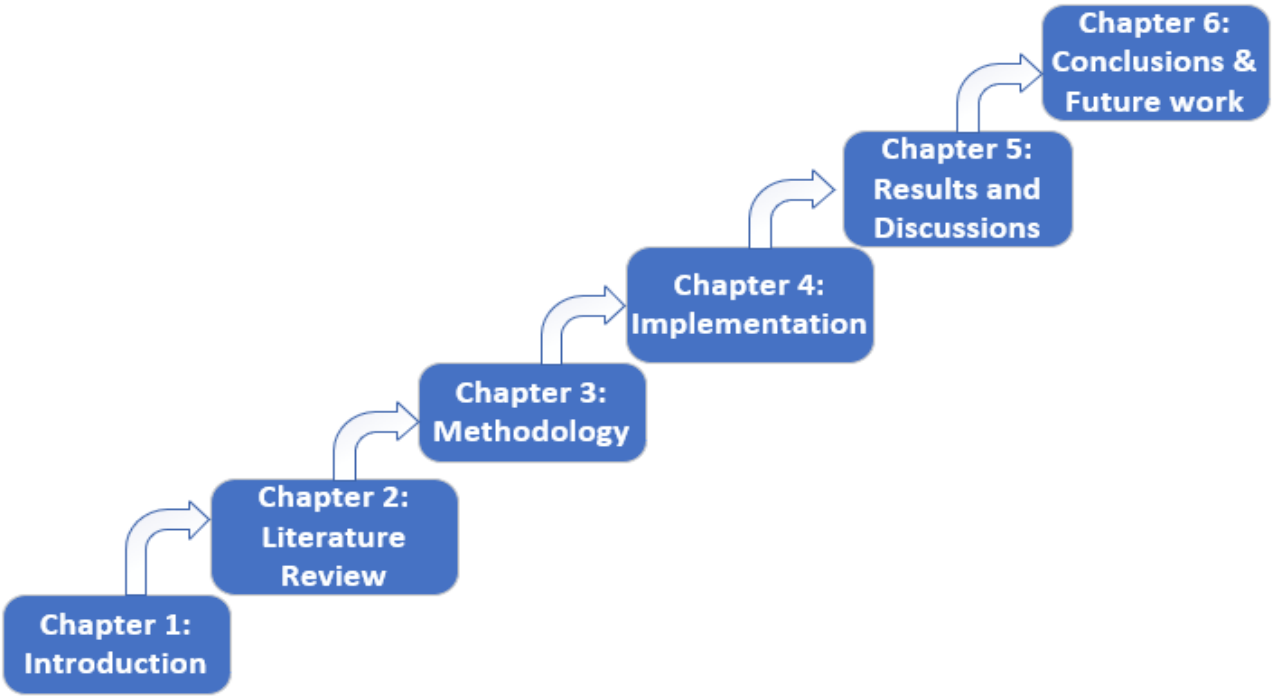

Figure 1.2. Dissertation organisation

1.12 Summary

This chapter provided a summary of the proposed study, emphasising its importance and relevance to the existing body of information. Most crucially, the chapter offered the research aims in the form of technical objectives, which must be satisfied by providing relevant responses to the chapter's research questions. The technique used to answer the study questions and the findings has been presented. The following chapter provides a comprehensive examination of existing literature and relevant studies.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

Smart farming is an emerging concept that results from the application of ICT to agriculture. Smart farming applications require a low-latency environment to operate efficiently. This chapter discusses technologies that have the potential to revolutionise and bring efficiency to agricultural operations, such as the 5G, IoT, cloud- and edge-computing platforms (Makondo et al., 2023).

The rest of the chapter is organised as follows. The chapter starts by providing an overview of smart farming in Section 2.2. This is followed by an overview of the IoT in Section 2.3. In Section 2.4 an overview of 5G networks is presented. The related works and the summary are presented in Sections 2.5 and 2.6, respectively.

2.2 Overview of Smart Farming

The agricultural sector is currently undergoing a 4IR concept known as Farming 4.0, because of the adoption of ICT into traditional methods of agriculture (Makondo, Kobo, Mathonsi, et al., 2023). As the world population continues to grow, so does the demand for food. Farmers are under constant pressure to produce more food with fewer resources. Therefore, farmers are turning to technology for answers and finding new ways to increase yields and decrease inputs. One of these solutions is called smart farming (Triantafyllou et al., 2019).

Smart farming is the application of intelligent ICT systems such as IoT, cloud-based processes, machine learning, artificial intelligence, and networking to farming systems, such as crop cultivation, livestock farming, aquatic and snail farming, to name a few (Triantafyllou et al., 2019). According to the food and agriculture organisation (FAO), by 2050, smart farming applications are expected to boost farmers' agricultural output by an estimated 70%. In South Africa, smart farming has significant potential to address agricultural challenges, increase productivity, and promote sustainable practices. The integration of technology and data-driven solutions is likely to appreciably improve the agricultural sector of the country, contributing to food security, economic growth, and environmental sustainability. South Africa's foremost research institutes, such as the technology innovation agency (TIA), innovation hub (IH), the council for scientific and industrial research (CSIR) and the agricultural research council (ARC), are leading active research and development in agricultural science and technological

advancements (Monamodi, 2020). Smart agriculture technologies that have the potential to digitise the agricultural sector are presented in Figure 2.1.

Figure 2.1. Smart farming technologies (Kimogol, 2019)

2.2.1 The applications of smart farming

Smart farming applications can increase agricultural output, eliminate resource waste, and encourage environmentally friendly farming methods. As technology advances, these applications will become increasingly vital to the addressing of modern agriculture issues. Some of the smart farming applications are depicted in Figure 2.2 (Ali et al., 2023).

2.2.1.1 Solar radiation: Solar radiation levels must be monitored to be used in digital models that predict plant growth, health, and disease risk.

2.2.1.2 Weather stations: The capacity to monitor wind, rain, temperature, and humidity is critical in both arable and cattle production.

2.2.1.3 Livestock tracking: global positioning system (GPS)-enabled asset tracking can be fitted to a collar, allowing the farmer to track the whereabouts of their cattle.

2.2.1.4 Agricultural drones: make it easy for farmers to monitor their fields, using image technology to detect disease, pests, and other growing issues.

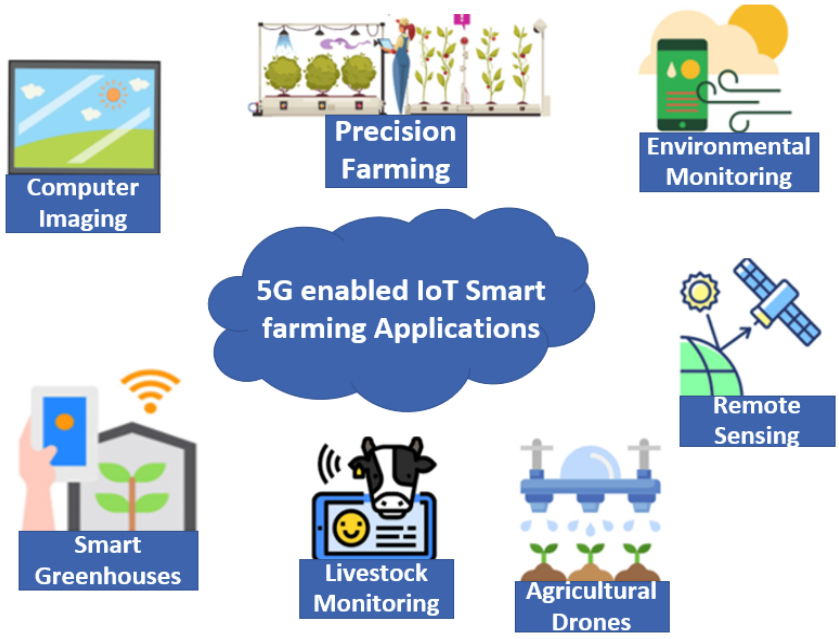

Figure 2.2. Applications of smart farming (Kimogol, 2019)

This research study focuses on the environmental monitoring applications that require ultra-low latency. According to (Jiang et al., 2018), latency refers to a measurement of the delay between the source and destination. In smart farming, latency refers to delay in data transmission between sensors and central monitoring or control systems (Saqib et al., 2020). Environmental monitoring applications rely on IoT. The IoT is one of the smart agricultural applications that allows producers to collect data such as which seed to use on their land, the required amount of fertiliser, changing climate conditions, and monitoring all external conditions from the machinery and computers they use while producing (Friha et al., 2021).

2.2.2 Real-time environmental monitoring applications

Smart agriculture monitoring systems use IoT technology to track and manage several aspects of farming, such as crop health, soil conditions, weather patterns, among many more. Using a network of sensors, actuators, and communication technologies, these systems collect and analyse real-time data from agricultural fields (Quy et al., 2022). The data that have been processed are then used to optimise agricultural operations, allowing for informed decision-making. The growing demand for financially feasible and environmentally sustainable farming methods serves as the foundation for an IoT-powered smart agriculture-monitoring system (Goel & Yadav, 2023). One significant advantage of IoT in agriculture is the capacity to remotely monitor many aspects of farm operations. This allows for the creation and gathering of critical data-monitoring applications closer to their source, with some data-processing duties managed directly on edge devices (Quy et al., 2022). IoT applications are discussed below.

2.2.2.1 Real-time detection and prevention of wildfires

In the context of IoT-driven smart farming, wildfires play a critical role in environmental monitoring. While wildfires may not be directly related to farming activities, they can profoundly impact the environment, including animals and crops. For example, in a livestock farm, an IoT system can be installed for monitoring purposes along the perimeter fence (Quy et al., 2022).

2.2.2.2 IoT geofencing

In smart farming, IoT-based geofencing is a latency-sensitive application. IoT geofencing involves creating and managing virtual barriers with IoT devices, known as geofences, around specific regions. Furthermore, such a technology can help safeguard animals and prevent livestock theft by immediately warning farmers when animals go outside their assigned borders (Sheham et al., 2022).

2.2.3 The impact of high latency for environmental monitoring in smart farming

High latency in IoT for smart farming, particularly when detecting fires, can have serious and potentially fatal consequences. In agricultural settings, the timely detection and response to fires is crucial to preventing crop damage, property damage, and even loss of life. Some of the consequences of high latency in environmental monitoring for fire detection in smart farming are as follows (Sinha & Dhanalakshmi, 2022):

2.2.3.1 Fire detection delay

High data-transfer latency can cause considerable delays in gaining valuable information from fire-detection sensors deployed throughout the agricultural area, such as temperature and smoke sensors. As a result of the delay in fire detection, the system may not alert farmers and authorities as quickly as needed to deal with a fire outbreak. Therefore, flames can rapidly gain speed and spread before any response measures can be implemented, converting a manageable situation into a full-fledged calamity (Ali et al., 2018).

2.2.3.2 Increased damage

High latency can cause delayed detection and response, which can lead to more severe damage. Crops, livestock, and equipment could be destroyed by fires if they spread quickly. In these circumstances, fires can also cause additional environmental damage, such as air pollution and damage to nearby ecosystems. The magnitude of the disaster increases with the time it takes to discover and respond to a fire, increasing the financial losses for farmers and the environmental effect.

2.2.3.3 Safety risks

The high latency in fire detection creates substantial safety risks for field workers and anyone around them. When alerts are delayed, there may not be enough time to implement evacuation preparations or perform critical safety procedures. As a result, persons in the vicinity of the fire, particularly farm labourers, may be exposed to dangerous circumstances and probable injury.

2.2.3.4 Resource inefficiency

Another result of high latency in fire detection is inefficient resource allocation. Due to delayed alarms, firefighting resources, including personnel and equipment, may not be delivered in a timely manner, or may be sent to the wrong place. This inefficiency can lead to a waste of time, effort, and resources, reducing the effectiveness of firefighting efforts and allowing flames to spread (Ali et al., 2018).

2.3 Overview of Internet of Things

The IoT is one of the most advanced wireless technologies now that enables seamless communication between objects in a wide range using the existing internet communications

infrastructure (Sadeeq et al., 2021). At its foundation, the concept relies on enabling interaction between many physical objects or devices that are linked together via the internet. In addition, IoT technology is applied in many industries today, including mining, transport, agriculture, healthcare, and entertainment. However, delivering low-latency connections for IoT applications to enable real-time communication can be challenging (Laghari et al., 2021). As the demand for low-latency connectivity grows, so does the rise of latency-sensitive industrial IoT applications, making 5G a potential alternative to addressing these latency requirements. The components of the IoT platform are displayed in Figure 2.3.

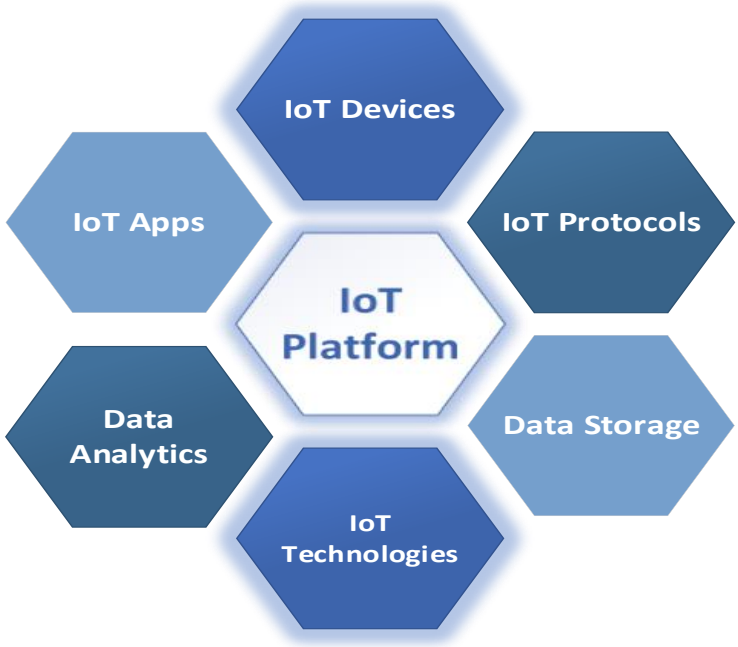

Figure 2.3. Components of IoT platform (Motlagh et al., 2020)

2.3.1 IoT architecture

Figure 2.4 shows a fundamental architecture in IoT that was introduced during the initial stages of research in this field. The architecture is composed of four layers, namely, perception, network, processing, and application (Mrabet et al., 2020).

2.3.1.1 Perception layer

In an IoT architecture, the perception layer is the first layer that collects data from numerous sources. It is made up of sensors and actuators placed throughout the environment to collect data on physical characteristics such as humidity, smoke, and temperature. These devices communicate with the subsequent levels of the IoT architecture, usually either through wired

or wireless communication methods, to send the acquired data for further processing and analysis (Mrabet et al., 2020).

2.3.1.2 Transport layer

The transport layer oversees the establishment of connectivity and communication amongst IoT system components. This layer consists of protocols and technologies that allow connections and communication between devices and the rest of the internet. The IoT uses a variety of communication network technologies, such as Wi-Fi, Bluetooth, Zigbee; and cellular networks such as 4G and 5G. Furthermore, the transport layer could consist of routers and gateways that serve as intermediaries between devices and the internet; as well as security features such as encryption and authentication to protect against illegal access (Mrabet et al., 2020).

2.3.1.3 Processing layer

The IoT architecture's processing layer includes both software and hardware components responsible for collecting, analysing, and interpreting data obtained from IoT devices. Here, raw data is received from these devices, processed, and then made available for further analysis or action. This layer incorporates various technologies and tools such as data-management systems, analytical platforms, and machine-learning algorithms to extract valuable information and make data-driven decisions. For example, a data lake serves as a centralised storage repository for storing raw data collected from IoT devices (Mrabet et al., 2020).

2.3.1.4 The application layer

This layer communicates directly with the end user (Goel & Yadav, 2023). The application layer oversees the creation of user-friendly interfaces and features that allow people to access and operate IoT devices. This layer contains a variety of software and applications, including mobile apps, online portals, and other user interfaces that connect to the underlying IoT infrastructure. The application layer also comprises middleware services, which enable various IoT devices and systems to communicate and exchange data in real-time. This layer also offers processing and analytics tools to analyse and transform data into relevant insights. Machine-learning techniques, data-visualisation tools, and other advanced analytical skills can be included (Mrabet et al., 2020).

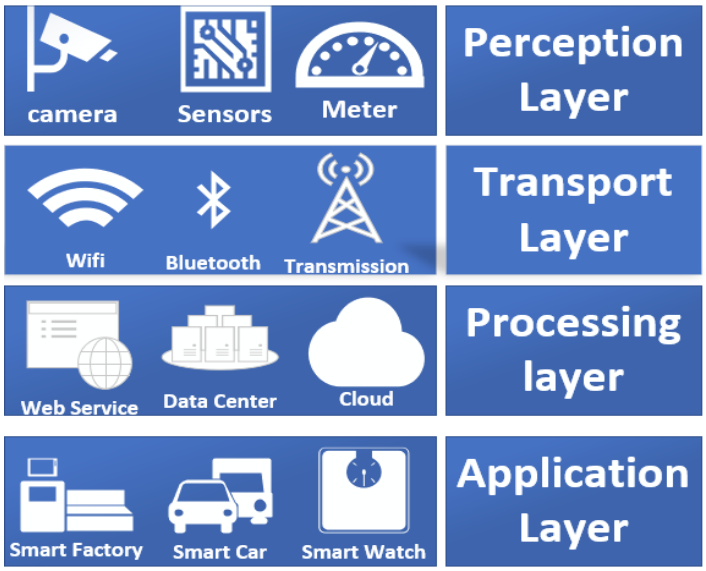

Figure 2.4. Layers of IoT architecture (Antao et al., 2018)

2.3.2 IoT building blocks

As depicted in Figure 2.5, IoT comprises four main components. These include things, gateways, network infrastructure (NI) and cloud infrastructure (CI). Things are physical items equipped with sensors, software, and other technologies that allow them to communicate and exchange data across a network. Gateway is an intermediate block between things and network or cloud infrastructure, used for connectivity purposes. The network infrastructure helps to control the information provided and ensures a safe and smooth flow. CI is equipped with information storage and computing skills (Kumar & Mallick, 2018).

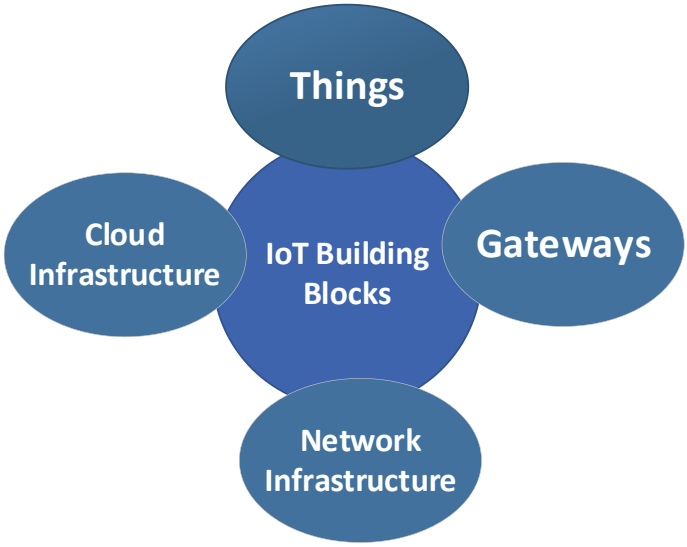

Figure 2.5. Building blocks of IoT (Prasad & Chawda, 2018)

2.3.3 Communication technologies in IoT

Over the years, IoT connectivity solutions have been dependent on three separate approaches: short-range communications, low power wide area network (LPWAN), and cellular networks (Ikpehai et al., 2018). These technologies are further discussed in this section.

2.3.3.1 Short-range communications

Short-range communication solutions include using technologies such as Zigbee, Bluetooth, and Wi-Fi to establish wireless connections between IoT devices (Cao et al., 2022). These short-range technologies are often cost-effective and are available to small and medium-sized agricultural enterprises in emerging markets. These solutions are energy efficient and suitable for battery-powered IoT devices (Feng et al., 2019). Furthermore, they are suitable for applications in small areas such as greenhouses, small farms, and sensor networks (Ojha et al., 2015). However, these solutions have limited coverage and are not suitable for large and widespread agricultural enterprises. Furthermore, these technologies operate in unlicensed frequency ranges and can be exposed to interference from other devices, affecting reliability (Azari & Masoudi, 2021).

2.3.3.2 Low-power wide-area network

According to (Akpakwu et al., 2017), low-power wide-area network (LPWAN) leverages unlicensed radio bands such as 2.4 GHz, 868/915 MHz, 433 MHz, and 169 MHz. These radio bands are notable for their low power consumption and large area coverage; hence they are referred to as LPWAN technologies. LPWAN includes technologies such as long-range wide area network (Lora WAN), narrow band IoT (NB-IoT), and Sigfox that prioritise long-distance wireless communication while minimising energy consumption (Raza et al., 2017). NB-IoT is built on existing LTE capabilities (Ayoub et al., 2018). The long-distance connectivity provided by LPWAN technologies is ideal for large-scale agricultural areas. LPWAN devices have energy-efficient capabilities and allow IoT sensors and devices to have a longer battery life. Furthermore, these technologies reduce operating costs and are attractive for high-cost applications in emerging markets. However, the bandwidth of LPWAN networks is limited. As a result, they are not suitable for high-speed data transmissions for IoT applications (Wang et al., 2020).

2.3.3.3 Cellular networks

The cellular network is a well-known global system that can offer ubiquity of access. These networks include second-generation (2G) cellular technology, third generation (3G) cellular technology, fourth generation (4G) cellular technology, and 5G (Raghunandan, 2022b). Cellular networks offer reliable broadband communication for a variety of voice calls and video streaming applications. This technology is considered a leading choice in providing connectivity to IoT (Biral et al., 2015).

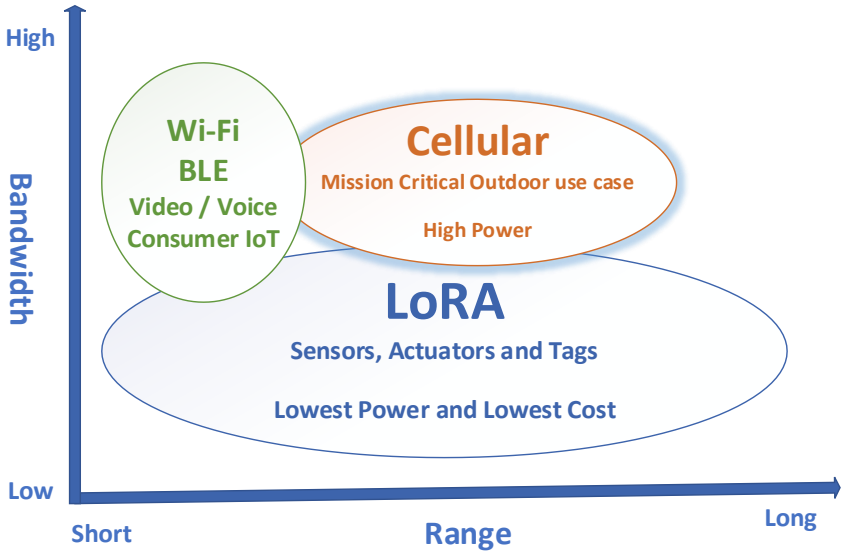

Figure 2.6. Communication technologies in IoT comparison (Qin et al., 2021)

Figure 2.6 compares many aspects of wireless communication technology, including transmission range, data throughput, energy usage, and cost. The choice depends on the application context because each of these technologies has both strengths and disadvantages.

The IoT has become an integral component of our increasingly interconnected environment. On the other hand, the success of IoT is strongly dependent on efficient and reliable communication between devices (Quy et al., 2022). This is where 5G technology comes in, 5G being the fifth generation of cellular networks. 5G provides much faster speeds, lower latency, and the ability to connect to many devices at the same time. 5G can unlock the full potential of IoT connectivity with these improvements (Liu et al., 2023).

2.4 Overview of Fifth Generation (5G) of Cellular Technology

5G is the fifth generation of a service-based architecture (SBA) for mobile networks. The main components of 5G are UE, 5GNR, and the 5GC network (Singh et al., 2017). The 5GC is placed at the heart of the 5G system architecture; the 5GC enables the 5GNR to serve the 3GPP defined use cases, namely, eMBB, uRLLC, and mMTC (Ali et al., 2021).

The uRLLC traffic prioritises high dependability (99.999%) and low latency (1~20 ms/packet) (Huawei, 2016). Furthermore, the 5G uRLLC is designed for services that must meet stringent latency and availability standards. 5G mobile networks that enable uRLLC must have low latency, with minimal packet loss or packets arriving out of sequence (Hamidi-Sepehr et al., 2021). With the 4IR under way, the uRLLC is poised to play a key role in shaping its future. As the world continues to adopt and integrate this cutting-edge technology into industries, we can expect significant advancements in automation, efficiency, and sustainability. By staying informed about the latest trends and leveraging the power of 5G uRLLC, businesses are well positioned to thrive in the rapidly evolving landscape of Industry 4.0 and beyond (Ma et al., 2019).

2.4.1 5G core network architecture

The 3GPP-standardised 5GC network architecture supports increased throughput demand, lower latency, and increased dependability to meet the requirements of the numerous applications and services that 5G must offer. 3GPP has defined 5GC to use cloud virtualisation and SBA across all 5G services and processes, including security, authentication, traffic aggregation, and session-management from end devices. The 5GC network allows the virtualisation of software functions to leverage a network function virtualisation that encompasses the physical resources as well as the virtual instantiations in the architecture of 5G networks (R. Mohamed et al., 2021). This architecture introduces the control and user plane separation (CUPS) concept that separates user plane (UP) and control plane (CP) functions, allowing for independent scaling of CP and UP functions, as shown in Figure 2.7. The CP of the 5GC network oversees network control, signalling, and service management; whereas the UP deals with transmitting user data while maintaining QoS (Yazıcı et al., 2014).

This separation of planes allows for more scalability, flexibility, and resource allocation within the 5G network, ensuring that both control and user data functions can be optimised independently, thus satisfying the varying requirements of several services and applications.

For example, a mobile network operators (MNO) can add more UP functions without adding more CP functions, and vice versa, depending on deployment strategies (Haile et al., 2020). To reduce user plane latency, the user plane functions could be spread geographically close to 5GNR, while CP services could be centralised to maximise the benefits of virtualisation (Condoluci & Mahmoodi, 2018).

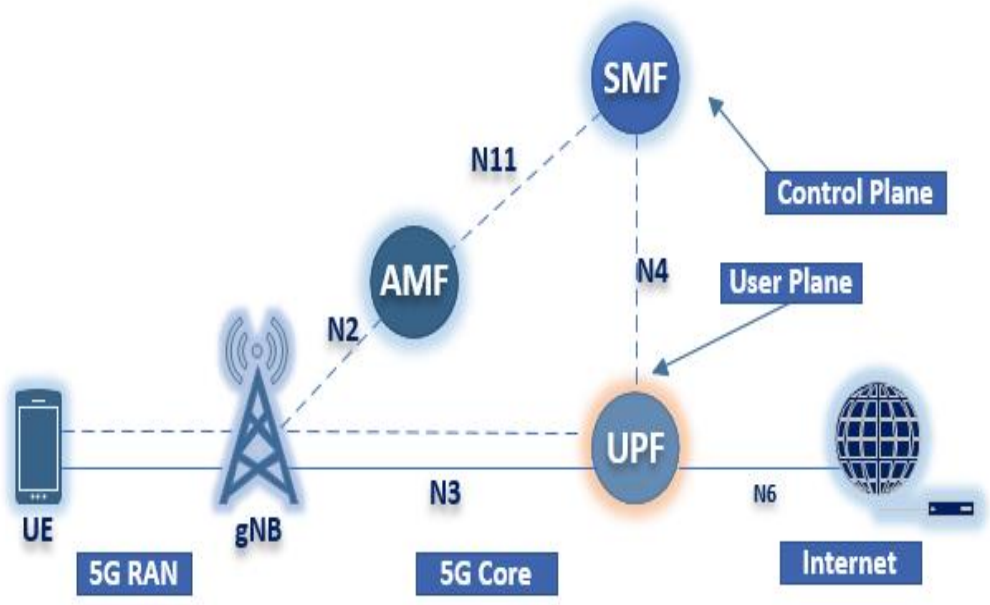

Figure 2.7. The 5G SBA architecture (Kunz et al., 2019)

2.4.2 5G core network functions

The 5GC network consists of several key functions that enable the operation of the 5G network and the delivery of numerous services. These functions are designed to support high data speeds, low latency, and a wide range of applications, including enhanced mobile broadband, exceptionally reliable low latency communications, and large-scale machine communications. The following are the main functions of the 5GC network (Choi & Park, 2017).

2.4.2.1 The UPF transports IP data traffic between UE and external networks. The UPF is an integral part of the 3GPP 5GC architecture. As part of the advancement of the UP and CP separation architecture, the UPF represents the decoupled UP. This development decentralises the UPF data-forwarding component while keeping CP functions in the core network. The UPF can analyse and aggregate traffic near network edges or subscriber locations. This technique improves bandwidth efficiency, decreases network complexity, and reduces traffic latency to satisfy 5G needs for numerous use cases, including uRLLC (Ahmad, 2023).

2.4.2.2 The access and mobility management function (AMF) acts as the primary entry point for the UE connection. The appropriate session-management function (SMF) responsible for overseeing the user session can be selected, depending on the service requested by the UE. SMF is responsible for handling user-data sessions. The SMF manages QoS for user services, establishes, releases data connections, and manages the routing of user-data packets.

2.4.2.3 The authentication server function (AUSF) enables the AMF to authenticate the UE and access 5G core services.

2.4.2.4 Other functions, such as the policy-control function (PCF), the application function (AF), and the unified data-management (UDM), provide the policy-control framework for controlling network behaviour by applying policy decisions, and accessing subscription information.

2.4.3 5G cellular networking deployment models

Several types of 5G deployment options exist, namely, the 5G standalone (SA) network and the 5G non-standalone (NSA). The 3GPP has introduced the 5G NSA options as transition stages for MNOs seeking to transition to a complete end-to-end 5G SA network (Attaran, 2023). Based on available information, most MNOs in South Africa, including Vodacom and MTN, have implemented 5G NSA networks for commercial purposes, primarily emphasising the eMBB service class. However, RAIN is the only mobile MNO in the country to have deployed a 5G SA network, covering specific regions and targeting future deployment of IoT use cases (Admire, 2022).

2.4.3.1 5G NSA

The telecommunications service provider can use a pre-existing 4G core network architecture to oversee CP and signal traffic within the 5G NSA solution. Thus, 5GNR technology can be incorporated into the current 4G infrastructure by using a dual connection configuration, deploying both 5G and 4G base stations. The NSA architecture is prominent in the early stages of 5G adoption and is preferred by many MNOs, particularly those who are not yet prepared to make major upfront expenditures, or manage the costs associated with moving to comprehensive 5G networks (Attaran, 2023). The 5G NSA offers a cost-effective solution for 5G deployment; however, it delays the fulfilment of 5G networks. As it employs two distinct cellular technologies, the NSA requires significant power consumption. Furthermore, it does not provide ultralow latency, which is a critical 5G capability (Attaran, 2023).Figure 2.8 shows the architecture of the NSA. The 5G NSA introduces latency due to the complexities associated with the integration of 4G and 5G networks.

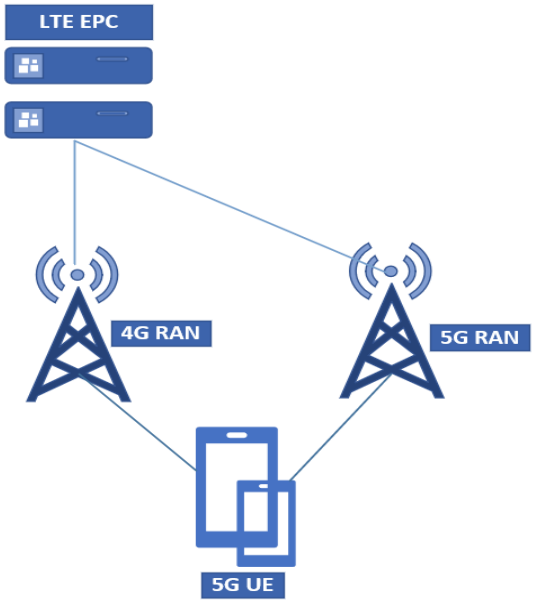

Figure 2.8. 5G NSA deployment (Liu et al., 2020)

2.4.3.2 5G SA

The 5G SA network offers a substantial advancement in telecommunications technology, with an innovative core design that distinguishes it from 4G networks, having no dependency on the 4G network infrastructure. The 5G SA architecture is distinguished by its fully virtualized network, which is meant to maximise efficiency and deliver crucial 5G services independently. Unlike the NSA, SA has additional features such as reduced latency and centralised network-

management operations (Attaran, 2023). However, adopting 5G SA has several challenges, including increased costs and a steep learning curve for experts who manage complicated infrastructure. Despite these challenges, many carriers are exploring relocation to South Africa to fully realise their potential. Compared with 5G NSA, SA provides new business prospects, improved user experiences, increased network efficiency, and reduced complexity. The focus of this research will be on building a 5G SA network to explore its capabilities and benefits. 5G SA is depicted in Figure 2.9.

Figure 2.9. 5G SA deployment (Liu et al., 2020)

2.4.4 5G key technologies

5G technology is a much-anticipated breakthrough that will help to meet the evolving demands of the industry. The 5G, which stands out for its increased mobile capacity and remarkably low latency, represents a significant advance over previous 4G networks. With increased capacity and bandwidth speeds up to four hundred times faster than its predecessors, 5G is poised to alter the wireless network environment. The emergence of 5G has the potential to broaden the mobile ecosystem by facilitating countless connections with rapid speeds, enhanced reliability, and minimal latency (Li et al., 2020). The following section delves into key technologies within 5G that have the ability to diminish latency.

2.4.4.1 Massive multiple-input multiple-output

Massive multiple-input multiple-output (MIMO) technology is a well-established wireless communications technique that allows several data signals to be sent and received over the same radio channel at the same time (Eze et al., 2018). MIMO techniques are widely used in wireless communications, as well as in 3G, 4G, and 5G networks. Furthermore, massive MIMO can help to reduce latency by reducing the amount of time it takes for data to travel between the base station and the device (Mamushiane & Dlamini, 2017).

2.4.4.2 Network slicing

Network slicing (NS) refers to the configuration of a network to support multiple logical networks on a single physical infrastructure, facilitating traffic division (Zhang, 2019). NS is important in 5G, as it accommodates a wide range of services with differing performance needs. The 5GC network supports end-to-end (E2E) network slicing, which allows for the splitting or 'slicing' of physical network resources into many independent networks across various frequency band segments (Esmaeily et al., 2020). Thereafter, these network slices can be devoted to certain use cases, such as uRLLC applications with stringent latency requirements. Network slicing is a critical use case for 5G SA because it optimises network capacity and ensures that resources are deployed efficiently (Esmaeily et al., 2020). Figure 2.10 shows how NS is implemented and used in a variety of sectors and companies.

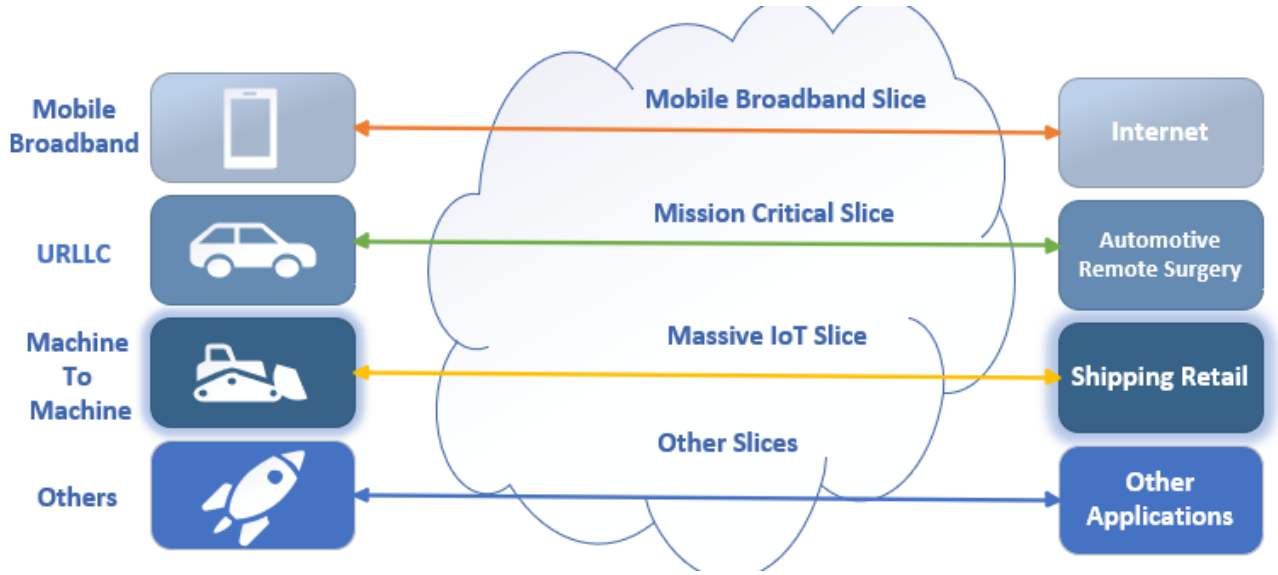

Figure 2.10. Network-slicing architecture (Guan et al., 2018)

The network slice as a service (NSaaS) concept enables operators to build customised network slices as a service to their customers (Alotaibi, 2021). This results in the creation of a new business model for MNOs. However, in this study network slicing was not considered when optimising the latency.

2.4.4.3 Spectrum

Spectrum pertains to the radio frequencies allocated to the mobile sector for communication and wireless data transfer (Khan & Pi, 2011). Spectrum governs the capacity and extent of coverage in a mobile network. 5G NR networks are engineered to function across a broad spectrum range, supporting mobile-data speeds spanning from a few Kb/s to several Gb/s (Lin & Lee, 2021). 5G represents the first mobile communication technology with the potential to reach peak speeds through millimetre wave operations. 5G can take advantage of both the current spectrum used by older mobile technologies and the newly defined cellular frequency bands. Various spectrum frequency ranges are used for wireless communication, depending on the 5G use cases (Kharel, 2022). Depending on the application case, 5G operates on a combination of frequency bands. The spectrum used by the 5G services is separated into three distinct bands, as illustrated in Figure 2.11.

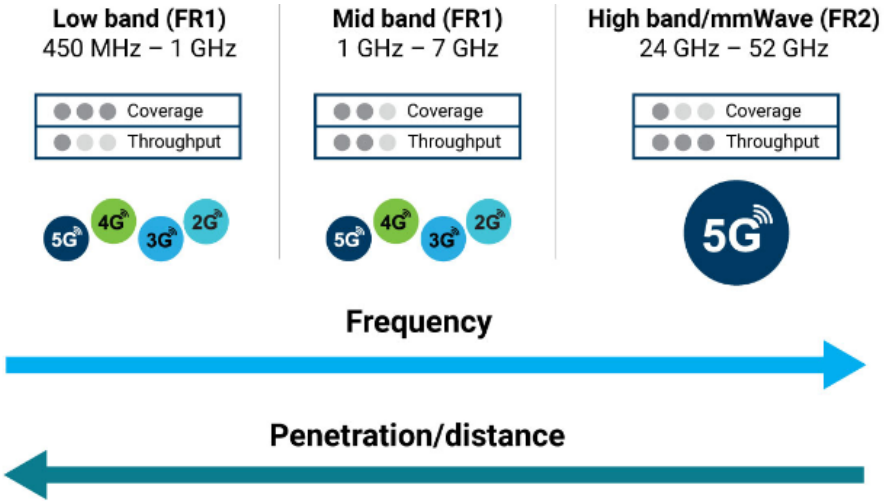

Figure 2.11. 5G Radio-frequency spectrum (Kharel, 2022)

2.4.4.4 Low-band spectrum

In this frequency range, devices operate slightly below 1 GHz. This spectrum band is crucial for providing wide coverage and penetrating indoor areas with minimal signal interference. Moreover, the channel bandwidth for this spectrum is 20 MHz. Low bands are highly reliable

in wide-area networks and can support essential communications. However, although many current wireless networks use the low-band spectrum, these wireless networks may not be ideal for applications requiring low latency (Kharel, 2022).

2.4.4.5 Mid-band spectrum

The 5G mid-band spectrum, ranging from 1 to 7 GHz, offers channel bandwidths from 50 to 100 MHz. Mid-band networks provide high capacity, significant data throughput, and minimal latency. Compared with the low-band spectrum, the mid-band spectrum enables faster speeds and broader coverage in urban, suburban, and indoor areas, albeit less than the high-band spectrum. Therefore, mid-band spectrum is ideal for urban centres, small towns, and suburban regions. The mid-band spectrum's adaptability and availability for private spectrum use have made it popular among businesses. However, applications requiring strict latency requirements, such as 1 ms, may find mid-band spectrum less suitable (Kharel, 2022).

2.4.4.6 High-band spectrum

The high-band spectrum in 5G operates at frequencies greater than 24 GHz. The high-band spectrum has the fastest speeds and the largest storage capacity. In densely populated areas, the high band is typically deployed to offer high-throughput and low-latency services. However, achieving widespread coverage proves challenging due to the propagation characteristics of millimetre-wave frequency (Kharel, 2022).

2.5 Latency in a 5G Network

The focus of this study is on uRLLC services, which drive the need for extremely low latencies. The uRLLC was first specified by 3GPP in Version 15; and is a key feature of 5G services. The uRLLC and mMTC are the most significant differences between 5G and previous generations of mobile networks. The uRLLC is optimised for mission-critical wireless communication applications, focusing on key performance indicators (KPIs) such as latency, reliability, and service availability. Other valid performance measures include jitter and synchronisation. Furthermore, uRLLC can handle wireless applications with high latency, dependability, and availability requirements (Li et al., 2018). This study focuses on the latency part of uRLLC. The key requirements for latency are discussed below.

2.5.1 Latency

3GPP defines E2E latency as the duration required to transmit data from a sender to a receiver, measured at the communication interface. This duration covers from the time the data are sent by the sender to the time the receiver successfully receives the data (Ibrahim et al., 2018). The total of radio, transport, and core latency is equal to end-to-end latency, which can be as low as 1 milliseconds (ms) (Parvez, 2018). The latency in 5G systems is split into two parts, namely, the UP and the CP.

2.5.1.1 User-plane latency

The latency of the UP is the length of time it takes a packet to travel from its source to its destination, measured in milliseconds (Ma et al., 2019). In an ideal world, where there is only one user on each uplink and downlink, and a few Internet protocol (IP) packets are exchanged, the uRLLC requirement for UP latency is 1 ms from source to destination (Ma et al., 2019).

2.5.1.2 Control-plane latency

CP latency can be defined as the length of time it takes the CP to move from its most battery-efficient state to an active data-transferring state. For the uRLLC CP, a maximum delay of 20 ms is permitted; however, smaller latencies are both conceivable and recommended (Ma et al., 2019).

2.5.1.3 5G End-to-end latency

E2E latency in 5G refers to the length of time it takes for a data packet to travel from its source (such as a user device or sensor) to its destination (application server, or the internet), receiving a response. This includes the total latency experienced by all network segments involved in packet transmission, processing, and delivery. The E2E latency ought to be 10 ms for most cases, although it should be reduced below 1 ms for some latency-sensitive applications. The overall E2E latency of a packet transmission in a 5G cellular network can be attributed to the 5GNR, fronthaul, backhaul, 5GC, and internet (Parvez et al., 2018), as depicted in Figure 2.12.

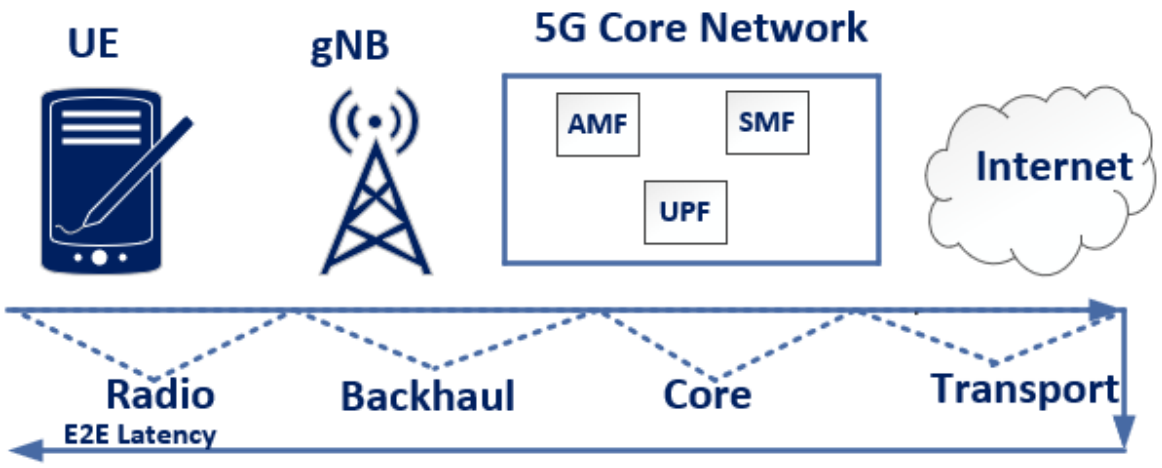

Figure 2.12. End-to-end latency in 5G (Berzin, 2021)

2.5.1.4 Sources of latency in 5G networks

The sources of latency in the 5G private network as defined by the 3GPP standards are explained in this section. These sources of latency are depicted in Figure 2.13.

a) UE

In the mobile network ecosystem, each component has a significant impact on latency and network performance. Beginning with the UE, which in this case is an IoT smoke detector, latency is influenced by several factors, including processing delays, signal-acquisition time, and protocol overhead. As the network's initiating endpoint, the UE analyses sensor data to detect events such as smoke or fire; and then connects to the nearest base station or access point in the 5GNR (Parvez, 2018).

b) 5G new radio (5GNR)

The latency in the 5GNR, which allows the transmission of the date wirelessly between the UE and the 5GC network, is influenced by signal propagation, modulation/demodulation, and scheduling protocols. Significant advances have been made in the reduction of latency from 2G to 4G networks, with greater progress projected in 5G networks. Furthermore, the 5GNR has components such as remote radio units (RRU) and baseband units (BBUs), which oversee radio signal transmission, reception, and baseband processing at the cell site. Collocating RRUs and BBUs reduces backhaul latency, optimises resource allocation, and improves network performance (Parvez, 2018).

c) Backhaul

Moving on to the backhaul that transports data between the 5GNR and the 5GC network, latency is determined by distance, link type, and congestion. Traditional configurations with RRUs and BBUs located at cell sites have low backhaul latency; however, queueing delays can arise as data packets await delivery over network links (Parvez, 2018). In general, microwave connections typically have higher latency, while optical fibres have comparatively lower latency.

d) 5G core

In 5GC, latency is influenced by various entities of the core network such as UPF, AMF, and SMF. Processing steps within the 5GC include UE authentication, 5GC bearer control, idle state mobility handling, mobility anchoring, UE IP address allocation, and packet filtering (Parvez, 2018).

e) Transport

The transport of data between the 5GC network and the internet or external networks introduces additional latency, influenced by link technology, network congestion, and distance. Processing delays in internet routers and switches, along with queueing delays due to network congestion, can impact E2E latency. Additionally, propagation delays may occur as signals travel through the internet infrastructure, particularly over long distances. In summary, each component within the mobile network ecosystem, including UE, 5GNR, backhaul, 5GC network and the internet, plays a vital role in shaping latency and network performance, highlighting the importance of optimising network architecture and technologies to ensure timely and reliable communication for various applications and services (Paolini & Fili, 2018).

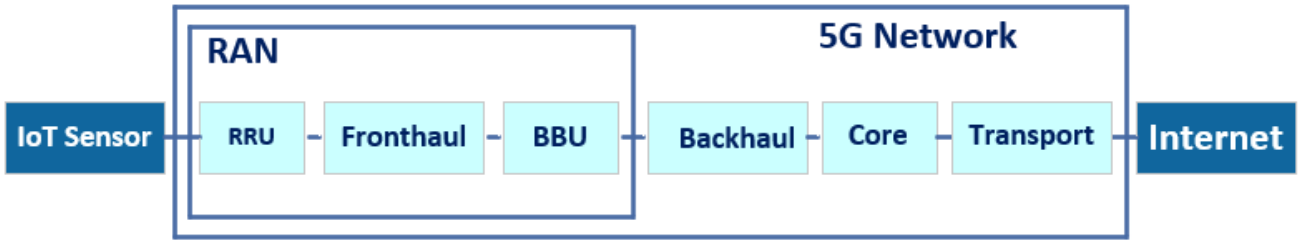

Figure 2.13. Sources of latency in 5G (Paolini & Fili, 2018)

2.5.1.5 Factors that contribute to latency in 5G

In 5G, latency can be caused by various factors, including propagation, transmission, routing, processing, and storage delays (Haile et al., 2021). A few of these are briefly described below:

a) Propagation

In 5G networks, propagation delay is the time it takes for signals to travel from the UE, which serves as the source, to the nearest base station (transmitter), and then to the internet via the backhaul and the 5GC. Propagation delay is mainly affected by the distance between the UE and the internet. When the UE and the 5GC are separated by a long distance, limiting propagation delay is critical to providing reliable connectivity, reducing latency, and maintaining quality of service in 5G networks. Operators can provide consumers with uninterrupted and high-quality network experiences regardless of their geographical location by addressing the problems associated with long-distance signal propagation (Goonatilake & Bachnak, 2012).

b) Transmission

Transmission delays occur when data packets are sent from the UE to the base station and then routed across the network to the internet. This consists of encoding, modulating, and transmitting data over the air interface and network architecture. Transmission latency between the UE and the base station can be reduced thanks to 5G's advances in data rates and modulation techniques. However, transmission delays inside the network infrastructure, particularly in reaching the internet, may vary depending on network congestion and routing efficiency (Goonatilake & Bachnak, 2012).

c) Queuing

Queuing delay occurs when data packets from the UE are queued at various network nodes, including base stations, routers, and switches, awaiting processing, and forwarding to the internet. Queuing delays in 5G networks can occur because of network congestion, especially at places where traffic converges on the internet. High volumes of traffic or congestion could cause significant queueing delays, which affect the overall latency experienced by the UE when transferring data to the internet (Goonatilake & Bachnak, 2012).

d) Processing

Processing delay is the length of time it takes network devices, such as base stations, routers, and core network elements, to process data packets bound for the internet. This comprises operations such as routing, switching, and protocol processing. Processing latency in 5G networks varies, depending on the complexity of network functions and the effectiveness of hardware and software implementation. Optimising processing performance is critical to reducing latency and ensuring timely processing and sending of data packets from the UE to the internet (Goonatilake & Bachnak, 2012).

Latency in 5G depends on four components, which are explained by the following mathematical equation defined in Equation (2.1).

$$Latency = L_{PT} + L_{TT} + L_{QT} + L_{PD} \quad (2.1)$$

where

- L_{PT} refers to the propagation delay.
- L_{TT} refers to transmission delay.
- L_{QT} refers to queuing delay.
- L_{PD} refers to processing delay.

2.6 5G-related Technologies for Ensuring Low Latency

Various technologies and communication protocols can reduce latency across the entire network. However, the reduced latency of the tactile internet creates several issues. These solutions are introduced in several 5G network segments, including access, core, and customer premises equipment. This section highlights strategies that ensure low latency between network tiers. The tactile internet will enhance human-machine interaction with reduced latency, enabling real-time systems. The tactile internet is a communication infrastructure that prioritises low latency, minimal transmission delays, high availability, and ultra-reliability while maintaining high security and service quality (X. Li et al., 2021).

In emerging markets, SDN and NFV technologies have emerged as the key enablers of 5G (Mamushiane & Dlamini, 2017). Furthermore, the 5G system is intended to provide data connections and services that will facilitate industry adoption via these innovative

technologies. The need for these technologies has arisen due to the numerous data-service profiles that must be supported by the 5G network.

2.6.1.1 Software defined networking

SDN is an emerging networking paradigm that seeks to simplify network management and configuration (Nguyen et al., 2017). SDN is a complete paradigm shift from traditional networking in that it tries to dramatically increase network efficiency through high-level innovative abstractions. SDN separates network intelligence, or the CP, from the packet-forwarding engine, or the data plane (Makondo, Kobo, & Mathonsi, 2023). This separation of planes enables centralised network intelligence to be delivered to the controller, giving it a comprehensive view of the network. Some of the advantages that SDN provides include vendor independence, heterogeneous network administration, reliability, and improved security that are not available in traditional networks (Ahmad & Mir, 2021). In this study, SDN was implemented on the backhaul to facilitate efficient traffic management, resulting in high throughput and low latency.

SDN is structured as a multitier architecture with three operational planes: the data plane, CP, and application planes (Mamushiane, 2019a), as depicted in Figure 2.14.

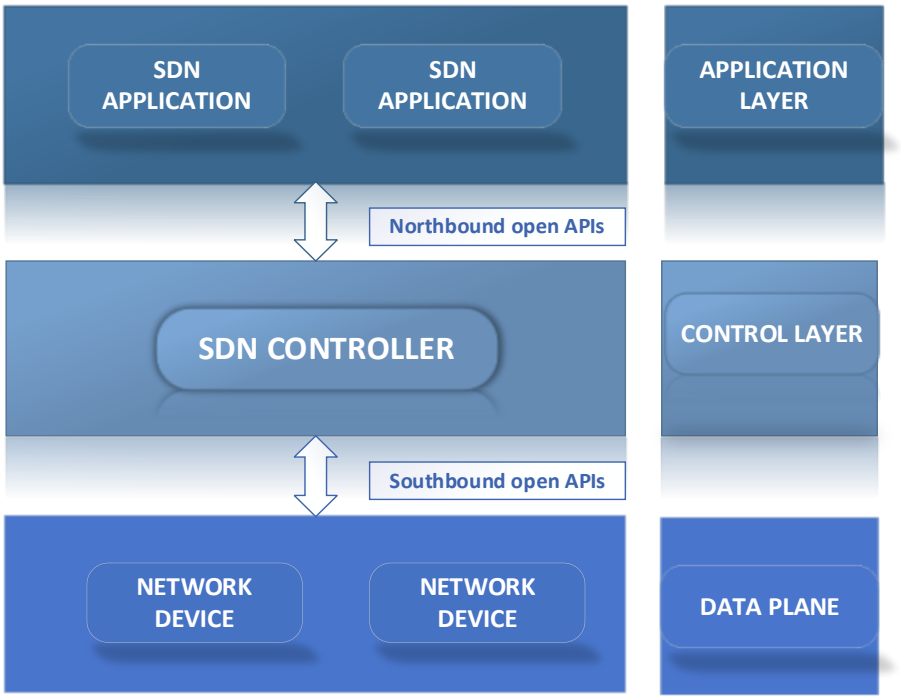

Figure 2.14. Software-defined network architecture (Nguyen et al., 2017)

The data plane is made up of various network elements such as switches, routers, and firewalls that expose their capabilities to the CP via the controller's southbound programmable interface (Xia et al., 2014). CP plays an essential role in network architecture, serving as a centralised entity that provides comprehensive visibility and control over all components within the network (Mamushiane, 2019a). The northbound programmable interface allows applications to transmit specific requirements and instructions to the controller (Mamushiane, 2019b). The controller then reads and translates the application requirements into low-level flow instructions required to configure the data plane. The controller's northbound interface is used to offer an abstracted view of resource use and state to the application layer. This is essential in hiding irrelevant features and only providing useful information to the application plane (Xia et al., 2014).

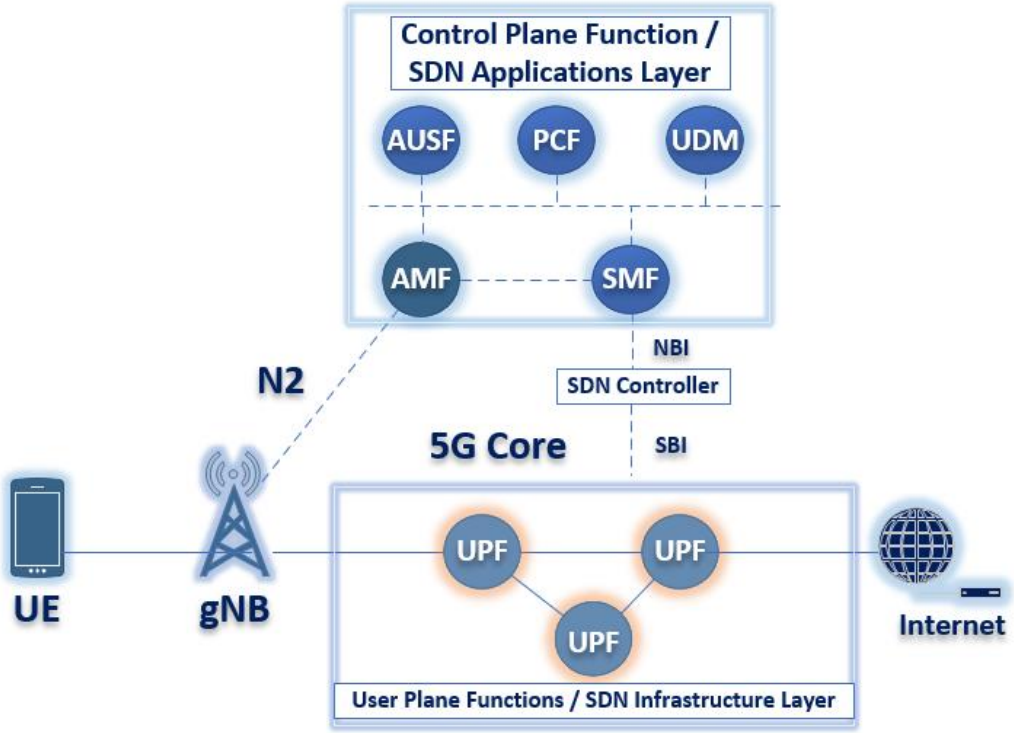

Figure 2.15. The SDN paradigm applied to the 5G core (Lin et al., 2019)

Figure 2.15 shows the application of the SDN paradigm to the 5GC. Inclusion of an SDN network controller into the mobile core architecture results in a three-tier architecture that can easily be transferred to a SDN paradigm. The SDN is used in 5GC networks to increase flexibility and efficiency. The SND is made up of an infrastructure layer for forwarding devices, such as SDN switches, a controller layer, which has a comprehensive logical view of

the physical network, and an application layer, which forwards decision logic to the controller. When incorporated into the 5G core, the application layer can be assigned to the CP, and the infrastructure layer to the nodes in the user plane. This provides a strong abstraction between layers (Mamushiane & Dlamini, 2017).

2.6.1.2 Network function virtualisation

The NFV concept is closely related to the SDN which virtualises network functionalities for flexible provisioning, deployment, and management (Kobo et al., 2017). The European Telecommunications Standards Institute (ETSI) proposed the NFV architecture, which provides standards for the deployment of NFV. Unlike traditional physical devices with tightly connected software and hardware, the NFV intends to integrate diverse network tasks on a general-purpose hardware device using standard software. The NFV aims to replace conventional, expensive, and specialised network operations such as switches, routers, and firewalls (Mamushiane & Dlamini, 2017). The NFV enables virtualised operations within virtual servers, which successfully manage large data volumes while increasing network flexibility, automation, and scalability. Depending on the application or service, network requests may be handled in the cloud or on the edge (Rafea, 2021).

In the realm of 5G, NFV serves the purpose of virtualizing diverse network functions, allowing the operation of multiple logical networks on a single physical infrastructure (Yousaf et al., 2017). Furthermore, the integration of NFV into the 5G architecture allows for the segmentation of a network into several logical networks, each capable of supporting multiple 5GNRs. Furthermore, NFV can also solve 5G challenges by optimising resource provisioning of virtual network functions (VNFs) for price and energy, scaling VNFs, and ensuring that VNFs constantly operate correctly (Yousaf et al., 2017).

The use of 5G NFV can provide 5G operators with several benefits, including:

a) Capex reduction: decreases the need for hardware purchase, and supports the pay-as-you-grow concept, thus avoiding unnecessary costs for 5G development (Basu et al., 2020).
b) Opex reduction: reduces the need for space, power and cooling equipment, thus simplifying the development and management of 5G services (Basu et al., 2020).

c) Time-to-market acceleration: decreases the time to develop new networking services by taking advantage of new market opportunities (Basu et al., 2020).
d) Flexibility and speed: rapid service escalation to accommodate shifting needs. By allowing software services to be delivered using any industrial server equipment, innovation is also supported (Basu et al., 2020).
e) Automation: the 5G SDN implementation can be supported by the NFV platform (Basu et al., 2020).

2.6.1.3 SDN/NFV-based 5GC network

To access services on a cellular network, all data traffic on the mobile network passes through the core network. In 5G, there will be an increase in mobile data traffic due to usage scenarios such as enhanced mobile broadband for high definition (HD) films, virtual reality, and augmented reality, ultra-reliable low latency sensitive applications (Nguyen et al., 2017). To address these 5G problems, we can use an architecture based on NFV and SDN for the development and management of the 5GC network, as depicted in Figure 2.16 below.

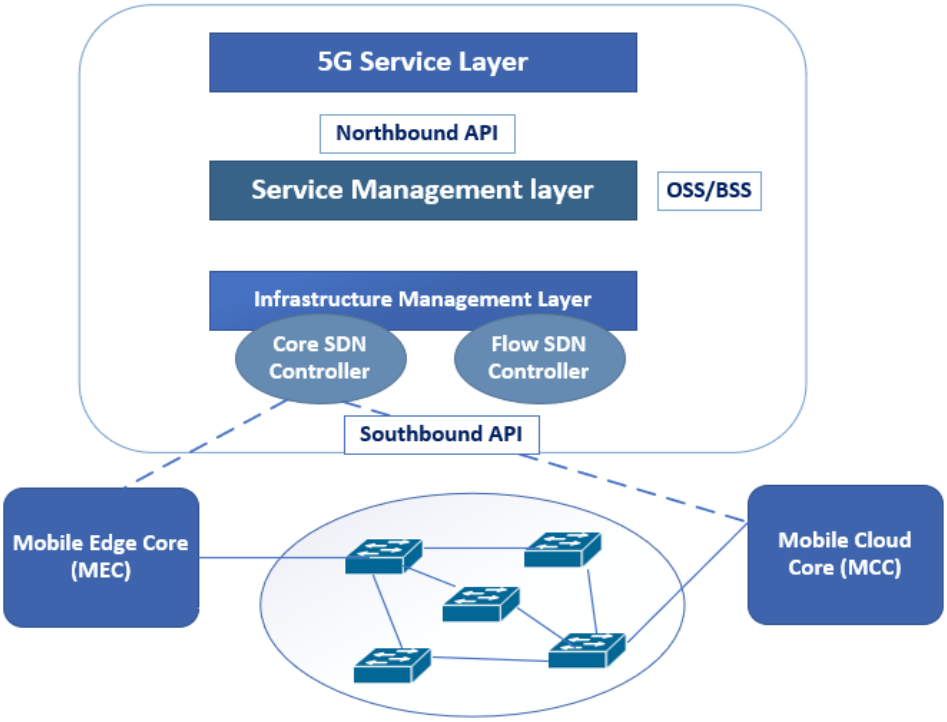

Figure 2.16. SDN/NFV-based 5G core (Nguyen et al., 2017)

The 5G service layer is the entry point for NFs that will provide numerous services to users. The service-management layer encompasses many features such as registration, authentication,

orchestration, security, and QoS, which oversee the setting of network function modules according to policies established by the business support system (BSS) and the operation support system (OSS) (Pereira & Karia, 2018). The infrastructure management layer oversees the infrastructure of the 5GC, including flow scheduling, network slicing, and NF deployment. There are two types of controllers: core SDN controllers, which manage service migration and deployment, and flow SDN controllers, which oversee the efficiency of backhaul network traffic routing. The 5GC UP and some applications are implemented within the network edge, while the CP is placed in a cloud data centre (Pereira & Karia, 2018).

2.6.1.4 Cloud computing

CC is a centralised computing architecture that allows businesses to access in-demand computer system capabilities such as computational power and massive data storage over the internet without direct user intervention, as depicted in Figure 2.17 (Sunyaev & Sunyaev, 2020). Data in cloud can be stored in more than one place, including third-party servers. Dropbox, Microsoft Azure, and Rack space are some examples of CC services. Large organisations employ cloud technologies to boost elasticity, agility, processing ease, and accessibility (Sadeeq et al., 2021).

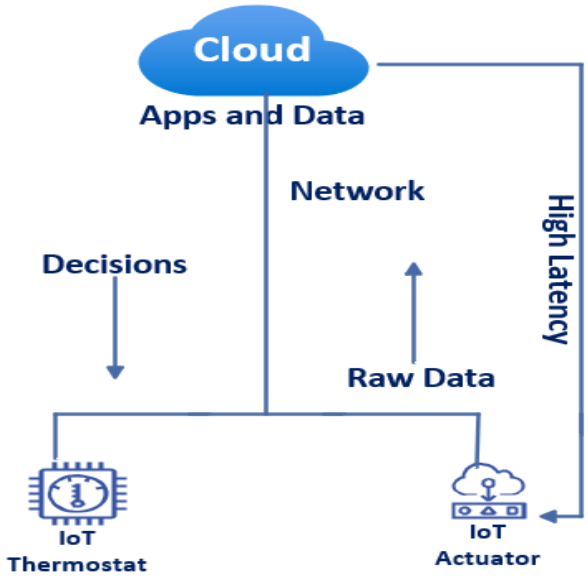

Figure 2.17. Cloud-computing architecture (Sadeeq et al., 2021)

In practice, 5GC network functions (NFs) are deployed as a native cloud function instantiated on an appropriate platform, for example, a cloud infrastructure (Alliance, 2021). Leveraging

cloud software technologies, the 3GPP NF architecture leads to greater flexibility, programmability, automation, and significant cost and energy reduction (Shah et al., 2021). 3GPP, introduced that 5GC NFs become cloud and virtualisation-based applications. The 5GC platform will be more programmable, allowing many distinct functions to be built, configured, connected, and deployed at the scale needed (Shah et al., 2021). Wireless providers are deploying their 5GC SA network functions utilising one of the three cloud delivery options listed below.

a) Private cloud: utilise your internal private cloud to deploy essential functions on-premises, as they have done in the past.
b) Public cloud: use a cloud service provider (CSP) like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud to deploy the network entirely on the public cloud.
c) Hybrid cloud: combines an on-premises data centre (or edge location) with a public cloud. Here, the services are an instance of the cloud CSP, through products such as AWS Outposts, Azure Stack Edge from Microsoft, or Anthos from Google Cloud.

Cloud-based 5G wireless networks provide several benefits and much potential in a wide range of sectors. However, due to its distance from the IoT devices and sensors responsible for data gathering, this architecture introduces additional latency. Such latency reduces the effectiveness and dependability of smart agriculture. As a result, EC has emerged as a viable solution to the latency dilemma in smart farming (Makondo, Kobo, Mathonsi, et al., 2023).

2.6.1.5 Edge computing

EC differentiates itself from the typical CC in that it operates on a revolutionary computing paradigm located at the network's edges. EC's core idea is to perform computational operations close to the data source (Cao et al., 2020). Edge technology processes data close to its source. EC has a wide range of applications, including healthcare, retail, transportation, and manufacturing. EC devices can make autonomous decisions and respond in real time utilising data received by IoT devices (Cao et al., 2020). The EC architecture is depicted in Figure 2.18.

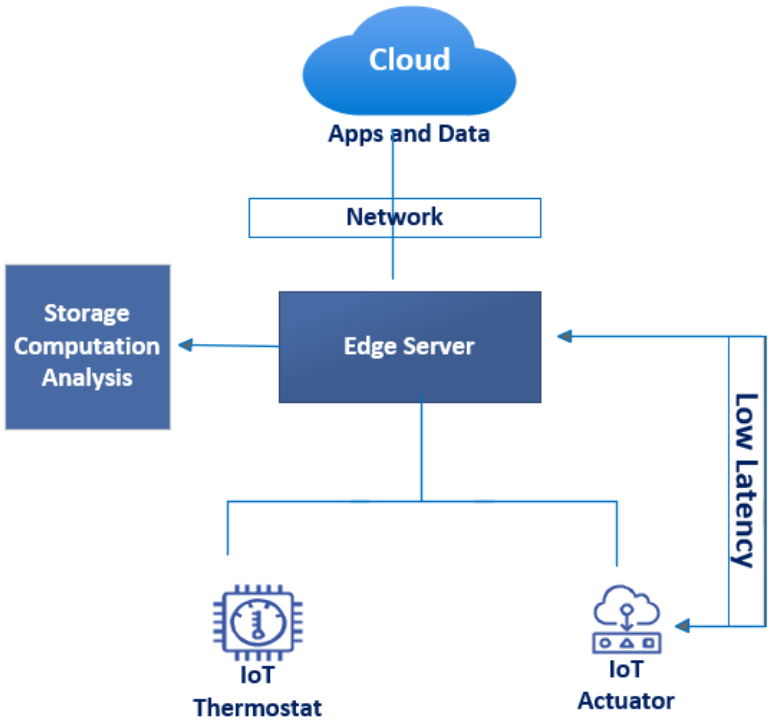

Figure 2.18. Edge-computing architecture (Sadeeq et al., 2021)

The deployment of EC varies depending on use-cases' requirements (Liu et al., 2022). Below are some implementations of EC.

2.6.1.6 Fog computing

Fog computing extends CC functionality to the network edge using intermediary computing nodes known as fog nodes. These nodes sit between the cloud and end-consumers or IoT devices. The basic goal of fog computing is to bring computer processes and services closer to the data source, reducing latency, and conserving bandwidth. Fog nodes located near the network edge, close to end users or IoT devices, allow real-time processing capabilities, making them suitable for applications demanding low latency and quick data analysis (Liu et al., 2022).

2.6.1.7 Multi-edge computing

Multi-edge computing (MEC) is a different specialised version of EC that primarily targets mobile network infrastructures, particularly 4G and 5G networks. MEC aims to move computing resources, such as servers and storage, closer to mobile-network base stations or access points. By doing so, MEC enables the delivery of low-latency services directly from the mobile edge, eliminating the need for data to pass through the 5GC network. MEC is

particularly useful in real-time gaming, location-based services, video streaming, and IoT deployments, all of which require fast responses and low latency to provide an excellent user experience (Zhu et al., 2020).

The 5G mobile architecture amalgamates IT technology and CC components to construct a software network framework at the peripheries of a 5G cellular network (Goel & Yadav, 2023). The operational model remains efficient, scalable, and sustainable, as data do not have to travel long distances (Hassan et al., 2019). In real-world scenarios, the deployment of the 5G UPF on the edge can meet the uRLLC requirements of 5G-enabled IoT applications in smart farming. As a result, the UPF will differentiate between users based on IP addresses and transfer user traffic in accordance with the traffic offloading policy. The UPF will then perform local forwarding and routing for data traffic that requires local processing, thus reducing delays, lowering data-forwarding latency, and improving user throughput and experience (Leyva-Pupo et al., 2022). As a result, the dream of having a latency between 1-20 ms will become a reality. The main differences between cloud and edge platforms are summarised in Table *2.1*.

Table 2.1. The Differences between Cloud and Edge Computing

Criterion	Edge Computing	Cloud Computing
Location	At the edge of the network	At the Internet
Geographical distribution	Localised	Centralised
Latency	Low	High
Distance	Few network hops	Multiple network hops
Computing power	Limited	High
Providing real-time services	Perfect	Provide delays

2.6.2 Exploring the advantages of edge computing in 5G networks

The efficacy of future 5G IoT networks is dependent on EC, which is now a major focus in IoT research. The primary objective is to reduce latency by limiting the amount of data transferred to the cloud. In 5G IoT contexts, EC performs a variety of critical functions to ensure seamless functioning (Kong et al., 2022). Edge provides the following benefits.

2.6.2.1 Local storage

UEs have the capability to offload significant amounts of data to edge servers through EC. Unlike the edge, which has limited storage capacity, the cloud has large storage resources. Servers store a variety of data types, including computational data, metadata such as location and timestamps, and monitoring data. Edge servers offer various storage options tailored to different data types. For example, ephemeral storage provides temporary storage for specific networked IoT applications (Kong et al., 2022).

2.6.2.2 Local computation

Edge servers manage the computational and processing tasks for sensor-connected devices. Unlike previous computational paradigms, the edge represents an intelligent computing system that brings these capabilities closer to the user's equipment (Liu et al., 2022).

2.6.2.3 Local Data Analysis

Regional data analysis enabled by edge computing involves the real-time collecting and processing of large datasets collected from UEs closer to the edge clouds. This method dramatically reduces the latency normally involved with delivering data to centralised cloud servers and waiting for results. Regional data analysis reduces the time required to send data across great distances to central cloud servers. As a result, the latency between data collection and analysis is minimised, allowing for faster decision-making and more responsive applications. This reduction in latency improves the overall efficiency and efficacy of edge computing systems, especially in scenarios requiring real-time insights and quick responses (Rafea, 2021).

2.6.2.4 Locally based decisions

The ability to process data locally minimises the dependency on external components, such as the cloud, making systems, especially those based on the cloud, more accessible. With increased bandwidth availability, many entities can make judgments locally. For example,

intelligent factories with automated processes can use the increased bandwidth to make autonomous judgments. This change to local decision-making improves system efficiency and responsiveness, which aids in the optimisation of numerous industrial processes (Rafea, 2021).

2.6.2.5 Faster data processing

EC accelerates data processing and allows activities to be completed more quickly. Previously, cloud servers encountered loading delays, which limited client interactions. However, with data being increasingly processed on edge servers, gateways, and even cell phones, enterprises can participate at unprecedented speeds (Rafea, 2021).

2.6.3 Factors slowing down the adoption of edge computing in 5G

In this section, we discuss some of the factors that hinder the adoption of EC in 5G-enabled IoT networks.

Integration of EC into 5G-enabled IoT networks presents several issues and considerations. According to (Y. Liu et al., 2020), one key hurdle is a lack of infrastructure. In many regions or remote areas, the infrastructure required for edge implementation may be unavailable or poorly built, impeding the deployment of edge solutions. The introduction of edge raises further security risks. Moving key data-processing operations to edge servers may increase security risks, as observed by (Zeyu et al., 2020). Edge nodes may be more sensitive to physical manipulation and attacks, necessitating the deployment of strong security mechanisms to ensure data privacy and protection.

Designing, installing, and operating edge computing systems involves knowledge of both network and distributed systems (Qiu et al., 2020). However, there may be a shortage of qualified people with the appropriate skills to do these jobs. Lastly, (Liu et al., 2020) pointed out that high investment prices provide a substantial barrier to the implementation of edge frameworks. Due to their complexity, edge solutions can be expensive and complicated, necessitating the use of additional equipment and assets. While these investments may increase efficiency, careful thought and critical thinking are required to justify the cost.

2.7 Related Works

The deployment of 5GC network functions in the cloud does not adequately address the latency problem for latency sensitive IoT applications. The fundamental problem that must be

addressed during the deployment of 5GC NFs is the placement of the UPF (Leyva-Pupo, et al., 2019). To our knowledge, only a few studies have focused on placing the UPF on the edge. Therefore, we had to peruse other techniques that were proposed to reduce latency.

2.7.1 Positioning of UPF techniques

Research was conducted into the positioning of the UPF, which is the critical component of 5G networks. The emphasis was on optimising the placement of the UPF to reduce latency, reduce relocations, and balance traffic load. Santoyo-González, et al. (2019) presented a paradigm for optimising the placement of UPFs based on the previously optimal position of edge nodes (ENs). The primary goal was to find the appropriate number and position of ENs and UPFs to meet the latency requirements of 5G cellular networks. The concept of anticipatory user-plane management to reduce latency was introduced by (Cervelló-Pastor, et al., 2022). In their work, these researchers leveraged forecasts of individual user activities to determine when to include intermediate user plane functions (I-UPFs) in the session data flow (Leyva-Pupo & Cervelló-Pastor, 2022). Furthermore, a simplified model overview to improve the position of I-UPFs was introduced by (Peters & Khan, 2019). Santoyo-González, et al. (2019) investigated the positioning of Anchor UPFs (AUPFs). The key goal of their work was to minimise the deployment and operating costs while meeting the 5G requirements. However, all researchers focused only on simulating their work without evaluating a model on a real 3GPP-compliant testbed. Such leaves a research gap because the results could not be validated in a real environment.

Service latency and bandwidth needs can be satisfied by deploying UPFs and applications at the network edge, as this reduces the length of the routing path under the concept of control and user-plane separation (3GPP, 2018; Jun et al., 2020). UPFs represent the evolution from traditional serving gateways (SGW) and packet gateways (PGW) to 5G networks. In contrast to existing networks, which handle a packet data unit (PDU) session through a single PGW and SGW, 5G networks allow a PDU session to be assigned to many UPFs at the same time (Leyva-Pupo & Cervelló-Pastor, 2022). This is because UPFs can be split into several functional responsibilities (microservices) that can be connected and directed as needed, according to 5G standards. However, since there are many variables and trade-offs involved, placing and chaining UPFs in an MEC environment is difficult (Leyva-Pupo & Cervelló-Pastor, 2022). Although moving network functions and applications such as UPFs to the network edge

frequently results in lower latency, better reliability, throughput, and fault tolerance, doing so still raises costs because this move requires a larger number of deployed instances. Furthermore, edge-based systems, which may include hundreds or thousands of embedded compute nodes, are often much more expensive than ordinary cloud services (Leyva-Pupo & Cervelló-Pastor, 2022).

Costa-Requena et al. (2018) proposed an SDN-based method to analyse the traffic flows of the user plane that connects the mobile core network to the radio access network. Layer 2 (L2) connections are established using either real-time or virtual ports in either a real or virtual switch with a programmable 5G UPF. Using Ethernet virtual local area networks (VLANs), the above researchers have set up L2 links as virtual connections. User-plane traffic delivered from base stations via general packet radio service (GPRS) tunnelling protocol/internet protocol (GTP/IP) culminates at the unified packet flow processor within the network core. The UPF routes traffic to the preceding switch using dedicated VLANs that replicate dedicated Layer 2 links for low-latency applications. Costa-Requena et al. (2018) proposed method leveraged SDN principles to improve traffic control and enable network slicing in 4G/5G networks. Their method involves establishing traffic-flow rules within the UPF switch using SDN solutions to tailor actions and priorities for traffic flows based on latency. Instead of typical GTP/IP tunnels, the UPF manages dedicated Layer 1/Layer 2 (L1/L2) links through VLANs, resulting in more efficient resource allocation. This methodology allows for finer traffic management within VLANs than per traditional methods, resulting in improved customisation and optimisation for varied applications or services. Furthermore, by operating as an independent component capable of local breakout for the user data plane, the UPF supports MEC, which brings computational resources closer to end users, lowering latency. The deployment results show that this approach successfully meets the performance requirements of uRLLC applications, which are critical in industries such as industrial automation and augmented reality.

Wang et al. (2023) presented a framework for integrating several UPF implementations. This framework complies with normal UPF but separates the CP from the UP. The authors combined UPF-CP using a packet-forwarding control protocol (PFCP) agent with two UPF-UP implementations: one with an Intel data-plane development kit (DPDK); and the other with a smart network interface card (SmartNIC). The researchers linked the framework with Free5GC, an open source 5G core network, and tested it using experiments. The results

confirmed their work's interoperability with Open5GS; and showed that the hardware-accelerated UPF outperforms the pure software solution in terms of packet-processing speed on the UP. Their work provided a heterogeneous UPF integration framework that enables the rapid creation of new 5G UPF user-plane implementations, while being overseen by a common UPF control plane. The messages between UPF and other 5G network services comply with the 3GPP specifications. Furthermore, SDN data-plane offloading technology is used to implement the functionality of the heterogeneous 5G UPF user planes. Compared with a typical VNF-based UPF, SNF reduces packet processing time and increases network throughput.

The work conducted by (Wang et al., 2022) demonstrated a proof-of-concept for an optical central office with MEC, UPF, and disaggregated optical line terminal (OLT) sharing an X86-server. The convergence of these technologies enables the mobile network to meet the strong latency and jitter limitations, while making use of the fibre-to-the-home (FTTH) central office location. By co-hosting the UPF from the mobile network and edge with the optical line terminal (OLT), Wang et al. (2022) demonstrated an end-to-end latency reduction of 93% and 80% in their real-time test bed with optical point-to-point (PtP) and point-to-multipoint (PtMP) topologies, respectively, as well as 90% and 11% in packet jitter.

The traffic-control mechanism for delay-sensitive IoT services can be implemented only in 5G UPF or edge routers in the data network (DN), without altering IoT devices or regulating the wide area network (Amemiya & Nakao, 2021). This mechanism regulates the latency of delay-sensitive traffic by distinguishing between delay-sensitive and delay-tolerant traffic, adding delays, and adjusting the receive window size of delay-tolerant traffic packets. The above authors further proposed an implementation architecture for programmable white-box switches that takes advantage of Berkley packet filter/eXpress data path (BPF/XDP) capability. The evaluation results showed that their proposed solution kept the delay-sensitive traffic latency within the required latency for single and multiple local 5G locations sharing an obscure wide-area network. However, the responsiveness of their proposed solution, particularly when numerous edge locations share the wide area network was not tested. Additionally, the authors indicated that they still needed to address the issue of scalability as delay-sensitive flows increase.

2.7.2 Network slicing techniques

Zacarias et al. (2023) presented a method for vertical radio access network slicing that allows the development of several isolated campus networks to accommodate uRLLC services. The researchers modelled the placement of the radio access network function as a mixed-integer linear programming problem with uRLLC constraints. In addition, unlike other similar approaches, their model considered user traffic flow from a known source node on the network's edge to an unknown a priori destination node. The authors indicated that their model could be experimented with on industrial campus networks by allowing UPFs to be dynamically placed to serve the uRLLC. The researchers further explained that they were also intending to investigate slice isolation at the server and node levels as their future work. In addition, replication of network functions could be investigated to improve the reliability of next-generation radio-access networks while supporting applications that use uRLLC.

In their study (Esmaeily et al., 2023) these researchers investigated the radio-resource slicing problem for uRLLC and eMBB use cases. In 5G new radio, uRLLC and eMBB traffic is multiplexed across several numerologies based on their distinct service requirements. The above authors introduced their optimisation approach, mixed-numerology mini-slot-based resource allocation (MiMRA), to reduce the impact on eMBB data rate of puncturing by distinct uRLLC traffic classes. Esmaeily et al.'s (2023) technique mitigated this consequence by instituting a puncturing rate barrier. The researchers also developed a scheduling approach that maximises the average rate of all eMBB users while keeping to each eMBB user's minimal data-rate need. The authors' simulation findings validate the practicality of our proposed resource allocation algorithm. However, their study focused more on eMBB use cases.

Kwabena (2018) designed, developed, and deployed a testbed that demonstrated how network resources can be dynamically sliced to allow ultra-reliable low-latency communication in a 5G network. This researcher's design was implemented using the TAKE5 testbed, which is constructed on the X-Network LTE network located on the Aalto campus, allowing easy integration. Kwabena's (2018) project focused solely on the slicing of the 5GNR and backhaul network. The EPC slicing was not considered in this work. The project's goal was to simplify network slicing by first defining it and then implementing it using practical network devices that support virtualisation to offer isolation between slices. Implementation also made use of indoor base stations and several virtual evolved packet cores (vEPCs). The implementation was

evaluated, and the results outperformed previously published network slicing techniques. However, Kwabena's (2018) experiment was focused on 4G rather than 5G networks.

2.7.3 Other latency optimisation techniques

Several studies have concentrated on the placement of the gateway in the context of 5G network deployment. Kiess & Khan (2019) proposed a gateway-location cost model that did not account for latency. Costa-Requena et al. (2018) used SDN to create 5G UPF components but did not specifically address latency reduction. For session management and intermediate placement of UPF (Peters & Khan, 2019) used a learning-based technique. Leyva-Pupo et al. (2020) investigated ideal UPF placement configurations while considering cost and latency; however, these researchers did not consider cooperative deployment with edge servers.

Nsiah et al. (2019) studied the feasibility of using the strategies proposed by 3GPP for latency reduction in cellular networks such as semi-persistent scheduling (SPS) and short transmission time interval (sTTI). Nsiah et al. (2019) first evaluated these strategies before putting them into practice in an open-source network simulator-3 (NS3). The researchers explained that they carried out simulations to analyse the uplink user-plane latency focusing on category-NB1 UE. According to their findings, SPS and sTTI demonstrated the potential to significantly reduce latency in NB-IoT systems. Furthermore, the authors indicated that, by incorporating both strategies into NB-IoT systems, NB-IoT will be positioned as a desirable technology for low-data-rate uRLLC applications. However, Nsiah et al.'s (2019) solution was not tested using several UEs in the uplink latency to determine the impact of multiple UEs on performance.

In their study, (Van Huynh et al., 2022) addressed the issue of minimising latency in computation offloading in digital twin (DT) wireless edge networks for industrial IoT environments using uRLLC links. The DT-based edge networks provide a powerful computing framework which enables computation-intensive services, in which the DT is utilised to simulate the computing capabilities of edge servers and to optimise the resource allocation of the entire system. The objective function consists of local processing delay, uRLLC-based transmission latency, and edge-processing latency, all of which are subject to communicational and computational resource budgets. Therefore, minimal latency is achieved by concurrently optimising transmit power, user association, offloading sections, user-processing rate, and edge-server-processing rate. Due to complicated non-convex constraints and strong coupling variables, the presented issue is extremely difficult. To address this computationally difficult

problem, researchers offered an iterative technique that divides the original problem into three subproblems; and solves it using an alternating optimisation strategy mixed with an inner convex approximation framework. The simulation results show that the proposed strategy is effective at reducing latency compared with alternative benchmark systems. In closing, the researchers indicated that they were still to address the problem of heterogeneous computation.

2.8 Summary

In this chapter, this research study provided a detailed overview of smart farming. This chapter further presented the overviews of the smart farming enabling technologies such as IoT, 5G, and other 5G-related technologies such as SDN, NFV, and CC, among many others. Finally, this chapter has reviewed the related works that were conducted to reduce latency in 5G-enabled IoT applications for smart farming using ED. Subsequently, the research study presented the research gaps identified from the existing solutions.

CHAPTER 3 SYSTEM MODELLING AND DESIGN

3.1 Introduction

The cloud-based core network is the conventional architecture proposed for 5G networks, characterised by centralising all core network functions. However, this model is inadequate to cope with the increasing number of IoT applications requiring low latency for a better user experience. On the other hand, edge architecture eases the heavy burden on the core network and minimises propagation delays. This chapter examines the edge architecture and provides justification for its adoption in 5G-enabled IoT in smart farming applications. This chapter defines the overall system architecture, including the network model based on edge computing in 5G cellular systems. Thereafter, the issue of placing UPF on the edge for 5G systems is formulated as a proposed solution.

The remainder of the chapter is structured as follows. The chapter begins by providing the system model in Section 3.2, followed by the latency model in Section 3.2.1, and the formulation and optimisation of the problem in Section 3.2.2. The summary is presented in Section 3.2.

3.2 System Model

The system model considers a four-tiered hierarchical architecture, which is composed of 5GC network, UEs, 5GNR, backhaul that is regarded as the edge, and central cloud layers. UEs connect to the UPF located at the edge nodes through wireless communication links using 5G base stations; while edge nodes access servers in the remote cloud data centre through fibre-optic based backhaul communication, as illustrated in Figure 3.1. This work focuses on smart farming applications that require ultra-low latency.

The model is based on a single radio connection using 5GNR. The network functions in the system are virtualised and executed on separate VMs using the NFV approach. The backhaul is based on SDN switches.

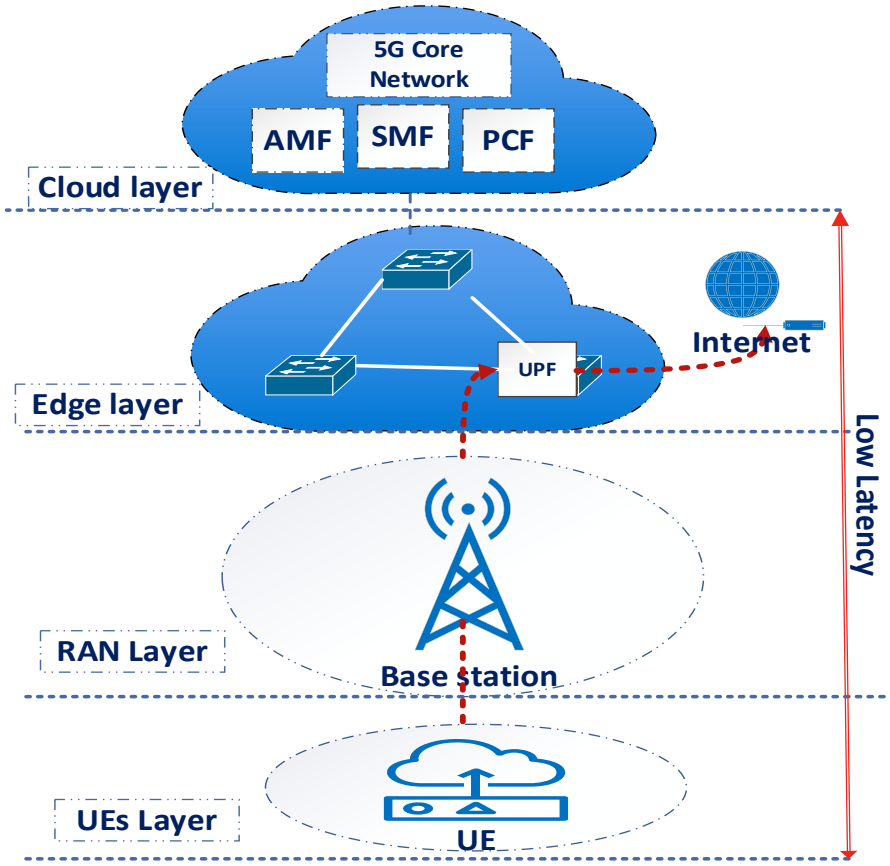

Figure 3.1. Proposed solution

In the modelling of the 5G network topology, a graph indicated by G = (N, E) is deployed, where N represents the set of edge nodes and E represents the fibre links that join them. The edge node ensemble consists of open virtual switches (OVS) located on the 5G systems backhaul. The OVS is an open-source virtual switch that is suitable for use in SDN systems. The UPFs are placed on the edge nodes. Let Lrc represent the minimum distance between the 5GNR and the UPF in terms of propagation delay, while $Lreq$ indicates the maximum acceptable latency between them.

The review of literature presented in Chapter 2 indicated that propagation delay plays a significant role in overall latency within 5G systems.

3.2.1 Latency optimisation model

The latency in 5G systems is split into two parts, namely, UP and CP. UP latency refers to the time it takes for a packet to be available in the IP layer between an evolved terrestrial radio access network (E-UTRAN) edge/UE of an evolved universal mobile telecommunications

service (UMTS) and an E-UTRAN node (Ritola et al., 2022). The CP latency refers to the time it takes for a UE to change from idle to active. In the idle state, a UE is disconnected from radio resource control (RRC). Once the RRC connection has been established, the UE transitions from idle to connected, and then to active, after entering dedicated mode. Low-latency communication is focused mostly on the UP because of its impact on application performance. Packet transmission delays in cellular networks can be caused by 5GNR, backhaul, 5GC, and the internet (Ritola et al., 2022). In 5G, latency is measured as round-trip time (RTT); and includes delay between the uplink (UL) and the down link (DL) (Håkegård et al., 2024). The total one-way latency in the 5G network is defined in Equation (3.1).

$$L = L_{RAN} + L_{backhaul} + L_{CoreNetwork} + L_{transport} \quad (3.1)$$

Each term represents the delays at UE, 5GNR, 5GC and Transport, respectively.

where

- L_{RAN} refers to the packet transmission time between the 5GNR and the UE which is caused primarily by the wireless communication medium.
- $L_{backhaul}$ refers to the time to link the 5GNR to the 5GC network function known as UPF. In most cases, copper wires, microwave, or optical fibre, connect the 5GC with the 5GNR.
- $L_{CoreNetwork}$ refers to the core network that connects the UE to the internet.
- $L_{transport}$ refers to the latency in data communication between the UPF of the 5GC and the internet.

Therefore, the RTT delay is approximated defined as $2 \times L$ in 5G networks. The edge may contain several forwarding nodes, such as routers, switches, concentration points, and UPFs. Each node contributes to the RTT delay. The latency is given by Equation (3.2).

$$L = L_{prop} + (N_i + 1)L_{Tx} + (N_i + 2)L_{process} + L_{TP} \quad (3.2)$$

where

- L_{prop} refers to the propagation delay between the UE and the internet.
- L_{Tx} refers to the transmission delay for each link.

- N_i refers to the number of forwarding nodes e.g., switches, routers, and points of concentration in the core network.
- $+2$ indicates a fixed processing delay or a consistent factor influencing processing latency.
- $L_{process}$ is the processing delay at each node.
- L_{TP} represents the latency associated with the tunnelling protocol between the internet and the UPF.

The latency of a 5GNR can be affected by several variables, including network architecture, transmission times, processing delays, and signal distribution. The following is a simplified formula for 5GNR latency (see Equation (3.3)).

$$L_{5GNR} = L_{prop} + L_{trans} \tag{3.3}$$

where

- L_{prop} refers to the propagation delay that is the length of time it takes for signals to travel through the air.
- L_{trans} refers to transmission delay, which takes into consideration the length of time required for signal encoding, modulation, and transmission.

The propagation delay depends on the transport medium used for the edge links, e.g., fibre, copper, or microwave. The L_{prop} can be represented using Equation (3.4).

$$L_{prop} = \frac{d_{CN}}{v} \tag{3.4}$$

where

- d_{CN} refers to the distance from the 5GNR to the internet via the edge nodes.
- v refers to the velocity of the transmission over the selected medium.

This research study considers the speed of light in fibre optics cables. The formula is given in Equation (3.5).

$$v = 2\ x\ 10^8 \text{ for fibre} \tag{3.5}$$

The transmission delay, L_{Tx} depends on the packet size and on the transmission rate of the link and is given by Equation (3.6).

$$L_{Tx} = \frac{P_{size}}{\alpha C_{link}} \tag{3.6}$$

where

- P_{size} refers to the packet size.
- α refers to the fraction of the link capacity.
- C_{link} refers to the link capacity allocated for the uRLLC use case.

3.2.2 Problem formulation and optimisation

Given the above-mentioned system model, the 5G network's primary performance metric is latency. Consequently, the problem of minimising the delay in the moving of UPFs, particularly to edge of the network, can be formalised as a mathematical optimisation problem, leading to the identification of optimal deployment configurations through established optimisation techniques. Optimisation ensures that uRLLC is achieved in 5G networks. As a result, 5G network systems can support low-latency, ultra-reliable, high-volume, and real-time data-forwarding services for 5G-enabled IoT applications.

Decision Variables:

- Let x_i be a binary decision variable indicating whether UPF is placed at the edge node i as seen in Equation (3.7).

$$x_i \in \forall i \in N \tag{3.7}$$

where

- N represents a set of nodes.
- $\forall i \in N$ indicates that the following condition carries all nodes i in the set N.

The reasons for using binary decision variables stem from their ability to capture the deployment status of the UPF at each edge node in a concise manner, facilitating an

optimisation process that focuses on achieving specific network objectives, such as minimising latency.

Objective Function:

- Minimise the total latency by optimising the placement of UPF as seen in Equation (3.8).

$$\text{Min } L_{total} = \Sigma_{i \epsilon N} w_i \cdot d_i \quad (3.8)$$

where

- L_{total} refers to the total latency that needs to be optimised.
- w_i refers to the weight associated with latency between UE and UPF at the edge node i.
- d_i represents the distance between UE and the UPF at the edge node i.

Constraints:

- Ensure that only one placement of UPF goes to the internet (see Equation (3.9));

$$\Sigma_{i \epsilon N} x_i = 1 \quad (3.9)$$

- Binary decision variables (see Equation (3.10));

$$x_i \epsilon \{0,1\}, \forall i \epsilon \text{ N} \quad (3.10)$$

Decision Variables:

- x_i represents a binary decision variable that indicates whether the edge node i is where UPF is placed (1 if true, 0 otherwise).

3.3 Summary

This chapter presented the design of the proposed edge-based 5G model for optimising latency in 5G networks. The proposed model is designed by offloading the UPF from the central cloud to the edge to shorten the physical distance between the UE and the internet. Furthermore, this model allows IoT sensors, such as smoke detectors that are distributed across the farm, to

collect and transmit data to the UPF located at the edge for real-time processing and decision-making in smart farming. In the next chapter, we illustrate the implementation of the proposed model introduced in this chapter.

CHAPTER 4 IMPLEMENTATION

4.1 Introduction

Chapter 4 focuses on the detailed implementation of the edge-based 5G model introduced in Chapter 3, specifically addressing latency optimisation for latency-sensitive applications in the context of smart farming. This chapter describes the software tools used to construct the testbed, followed by the hardware implementation and the methodology used to measure latency performance. The implementation of the proposed solution is structured into four (4) sections. The initial part involves the implementation of the open source 5GC network, Open5GS. The second part is the deployment of srsRAN, followed by the emulation of backhaul; and lastly, the configuration of UEs.

The remainder of the chapter is organised as follows. This chapter starts by providing an overview of the testbed in Section 4.1, followed by the experimental set-up in Section 4.2; and lastly, the summary is presented in Section 4.3.

4.2 Testbed Overview

4.1.1 5G core network

The 5GC is the fundamental component responsible for network functions such as user authentication, session management, and mobility management. The 5GC consists of the AMF, SMF and UPF components (Shetty, 2021). The 5GC is connected to the 5GNR and the internet.

4.1.2 Backhaul

Backhaul is the network segment that connects the 5GNR to the 5GC network and is essential for data transfer in the 5G network (Raghunandan, 2022a).

4.1.3 5G New Radio

The 5GNR includes base stations, such as gNodeBs (gNBs), that connect consumer devices to the 5G network. The 5GNR is instrumental in ensuring dependable and swift wireless connectivity. 5GNR introduces notable advancements compared with its forerunners, particularly through the integration of MIMO technology and beamforming techniques. MIMO uses multiple antennae to enhance spectral efficiency and to augment network capacity, thus contributing to the overall performance and reliability of 5GNR networks (Ahmadi, 2019).

Beamforming allows accurate signal focusing in certain directions, reducing interference, and increasing coverage (Dludla et al., 2022).

4.1.4 5G UEs

UE refers to devices used by end users to access and use 5G network resources. This category includes a wide range of devices such as smartphones, tablets, laptops, and IoT sensors. These devices have hardware that is compatible with 5G, allowing them to communicate with both 5GNR and the larger 5GC network (Dludla et al., 2022). 5G UEs are designed to be adaptable, catering to a wide range of applications, from high-definition video-streaming to mission-critical IoT connectivity. These devices have advanced features such as eMBB for rapid data transmission, uRLLC for real-time applications, and mMTC to efficiently manage many IoT devices. The versatility of 5G UEs positions them as critical components in unleashing the full potential of the 5G network, catering to end users in a variety of sectors and circumstances.

4.3 Experimental Set-up

This section outlines the implementation of the proposed model described in Chapter 3. This model is implemented in a 3GPP-compliant 5G testbed. Furthermore, this experiment features a standalone 5G deployment to realise full 5G capabilities such as low latency and faster speeds. The primary objective is to shorten the propagation delay by placing the UPF node on the edge. The propagation delay can be defined as a delay between the UE and the internet. Our solution includes SDN switches at the edge near the gNBs that are configured to route each user's traffic to the UPF. An open network operating system (ONOS) SDN controller controls the SDN switches.

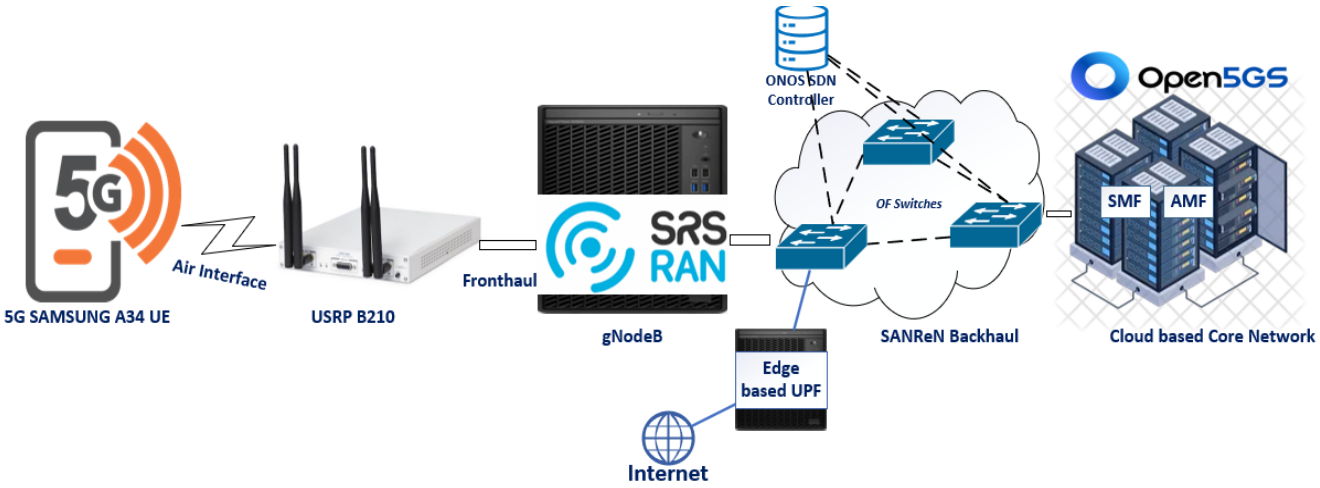

Figure 4.1. Proposed model experimental set-up

As shown in Figure 4.1, the proposed model is tested using an open source 5G testbed that conforms to 3GPP standards. This experimental set-up consists of the srsRAN operating on the Ubuntu computing system; and a software-defined radio (SDR) known as the universal software radio peripheral (USRP) model B210. In this set-up, the BBU is served by the srsRAN, and the RRU is operated by the USRP. The reception, transmission, and amplification of radio frequency (RF) signals are undertaken by the USRP unit, enabling an in-depth evaluation of the proposed model's performance (Mamushiane et al., 2023).

In this experiment, the USRP was utilised due to its ability to reconfigure and programme specific functionalities such as gain settings, modulation techniques, operating frequencies, and radio protocols, as well as its ability to address current conditions such as interference, noise, and traffic volume. The USRP was set up to operate in a single-input single-output (SISO) state without beamforming, leading to average throughput for a 5G network (Mamushiane et al., 2023). To improve data-transfer rates and reduce latency, the fronthaul connecting srsRAN and USRP uses a peripheral component interconnect express (PCIe) connector. The 5GC is deployed using Open5GS in an OpenStack Cloud. The SDN-based backhaul connection between SrsRAN and Open5GS uses 10 Gb/s emulated fibre lines. This backhaul was chosen to improve the planned 5G test bed for superior performance, with the goal of maximising throughput while minimising latency (Mamushiane et al., 2023).

4.3.1 Hardware used for modelling

The experiments were conducted using Dell precision 3660 workstations equipped with 8th Generation Intel® Core™ i7 processors, specifically the Intel (R) Core (TM) i7-8700 CPU

running at 3.20 GHz, with 6 cores, 32 GB of RAM, and a 250 GB solid state drive (SSD) storage disk.

4.2.2 Software technologies and tools used for modelling

This section introduces the essential technologies that enable and support our 5G testbed. These technologies were carefully chosen based on their common integration capabilities, as well as their popularity and record of accomplishment in the industry. Other significant characteristics were also considered, such as their stability, security, scalability, ease of use, cost effectiveness, and community support. Furthermore, the open-source nature of many of these technologies allows us to tailor the technology to our individual needs, which can lead to cost savings and lower vendor lock-in.

4.3.2 Open5GS deployment

Our experiments leverage the implementation of the Open5GS mobile core network. Open5GS is an open source 5GC implementation developed in the C programming language (Open5GS). Open5GS includes a web application to maintain UE membership information, ideal for testing. Open5GS is based on 3GPP standards Release 17 and supports IPv6, multiple PDU sessions, handover capabilities, and voice-over LTE and 5G new radio integration. Open5GS has a larger contributor base than other 5GC projects, leading to more frequent software releases and weekly repository contributions (Open5GS). Open5GS provides better performance in terms of data-plane bit rates, with an expansion of up to approximately 1 Gb/s with multiple UE connections (Lando et al., 2023).

Open5GS 5GC can be deployed directly on the metal or in a virtualised environment utilising hypervisors such as VMware or VirtualBox. The deployment entails installing the necessary database for 5G network services such as network repository function (NRF), PCF and UDM, configuring dependencies, and building a tunnel interface to route-user traffic (Mamushiane et al., 2023). The tunnel interface should be automated for persistence after system reboots. Users then clone the Open5GS repository, compile the software, and deploy critical 5GC functions such as AMF, SMF, UDM, and UPF. It is critical to disable firewall rules that may block communication, and in the event of a service failure, users should ensure that the tunnel interface is enabled, terminating all ongoing operations. This procedure allows for the effective deployment of Open5GS for the functionality of the Open5GS 5GC network.

In this experiment, the Open5GS CUPS architecture is used to separate the UPF from CP functions. As a result, the UPF functions as a standalone on a separate server that is deliberately placed at the network edge, closer to the UEs. Other fundamental network operations, such as AMF and SMF, are in the central cloud. This set-up improves network efficiency by decreasing latency and increasing bandwidth consumption, thus improving the performance of 5G services closer to end users.

Therefore, the PFCP connection is established between the UPF on the edge server and the SMF in the central cloud to provide efficient session management and data forwarding in the 5G network architecture (Amponis et al., 2023). This connection allows the UPF to communicate with the SMF, exchanging control messages and session-related information that is required for dynamic session creation, modification, and release. By establishing this PFCP connection, the UPF can receive SMF instructions on session policies, QoS parameters, and routing decisions, assuring optimal resource allocation and network performance. Furthermore, the PFCP connection enables real-time coordination between the network's edge and cloud components, resulting in seamless data transmission and administration across distributed network components. Figure 4.2 shows the PFCP status between the UPF (10.0.0.40) and SMF (10.0.0.30).

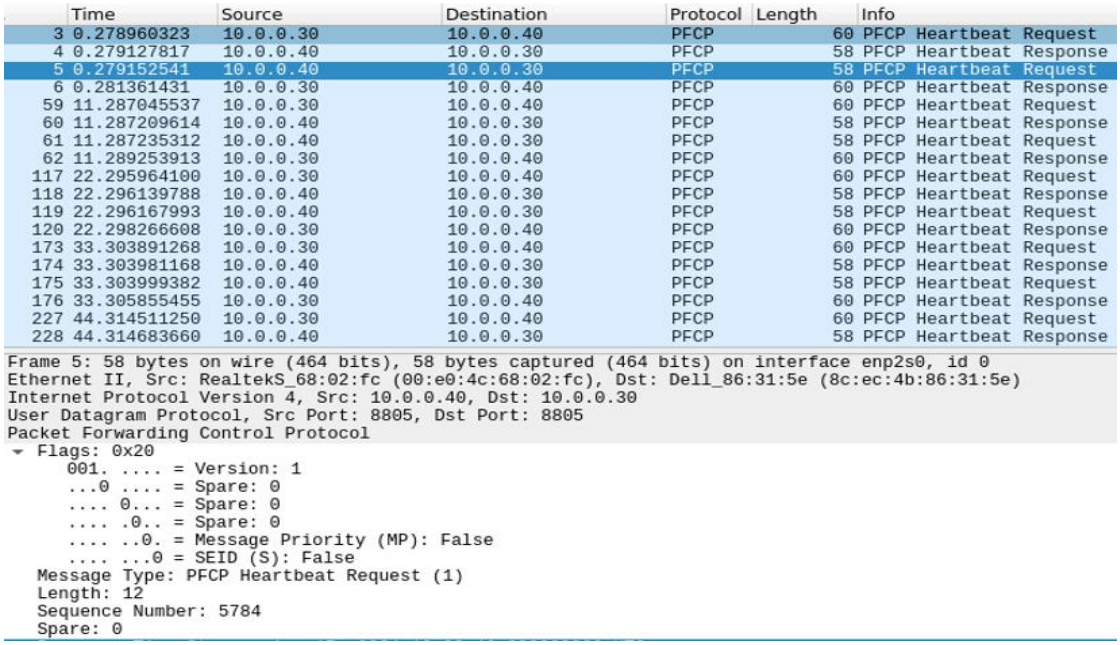

Figure 4.2. PFCP status between UPF and SMF

After establishing the PFCP connection, the next step is to establish the GPRS, and the general tunnelling protocol (GTP) between the UPF and the gNB in the 5G network. The GTP tunnel

provides a secure and efficient channel to send user data between the 5GC network and the UE. Upon successful registration of the UE on the 5G network, data requests are routed to the uplink tunnel endpoint identifier (TEID) associated with the uplink direction of the GTP tunnel. This tunnel encapsulates and transmits data packets from the gNB to the UPF, ensuring a reliable and seamless connection between the core network components and the UEs (Cruz, 2023).

Figure 4.3 shows the list of subscribers registered on the network. Open5Gs includes the web user interface (WebUI) capability together with dynamic updating of subscriber information, allowing UE data to seamlessly integrate into the 5G network. Users can easily add and manage subscribers by accessing the WebUI at http://127.0.0.1:9999 (Cruz, 2023). Open5GS stores information about subscribers in the Mongo database (Cruz, 2023).

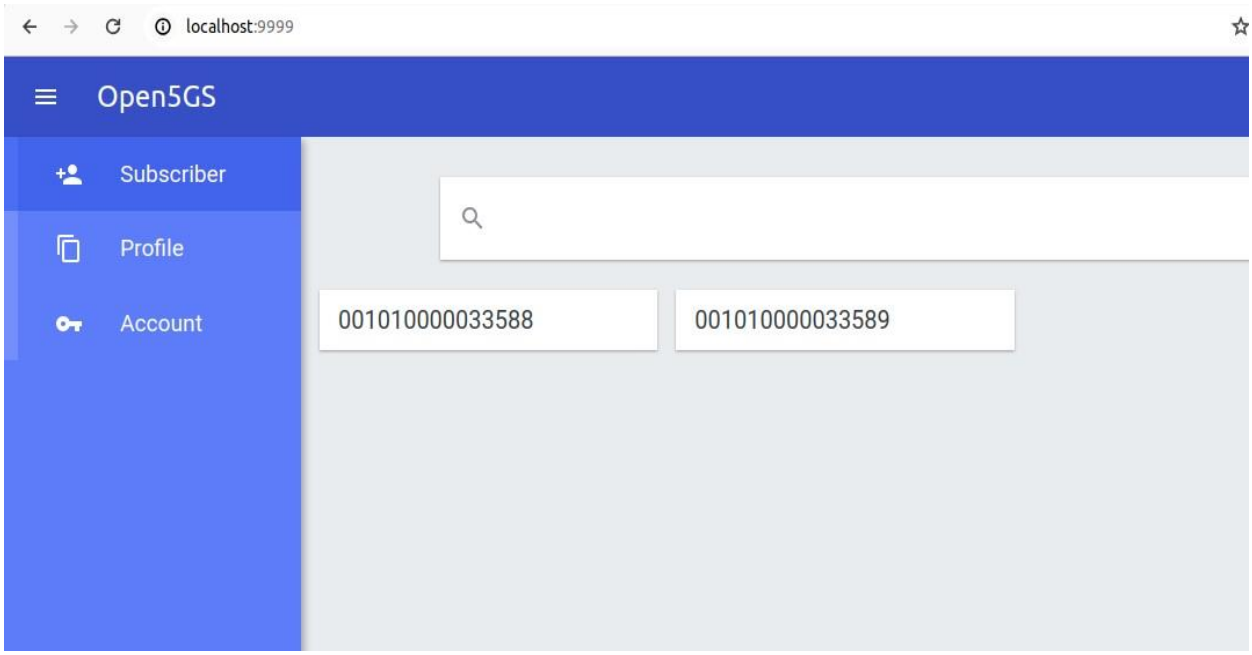

Figure 4.3. Subscribers in Open5GS database

Registration is a prerequisite for subscribers to access and use network resources efficiently. Therefore, to streamline the UE registration procedure within the network, it is imperative that the network has the required data to authenticate and verify the UE, as shown in Figure 4.4. During the registration process, crucial information must be provided, such as subscriber key (K), operator key (OPc/OP) and international mobile subscriber identity (IMSI). The provided values must align with the configuration parameters established for the UE, ensuring smooth

integration and maximising performance within the 5G network.

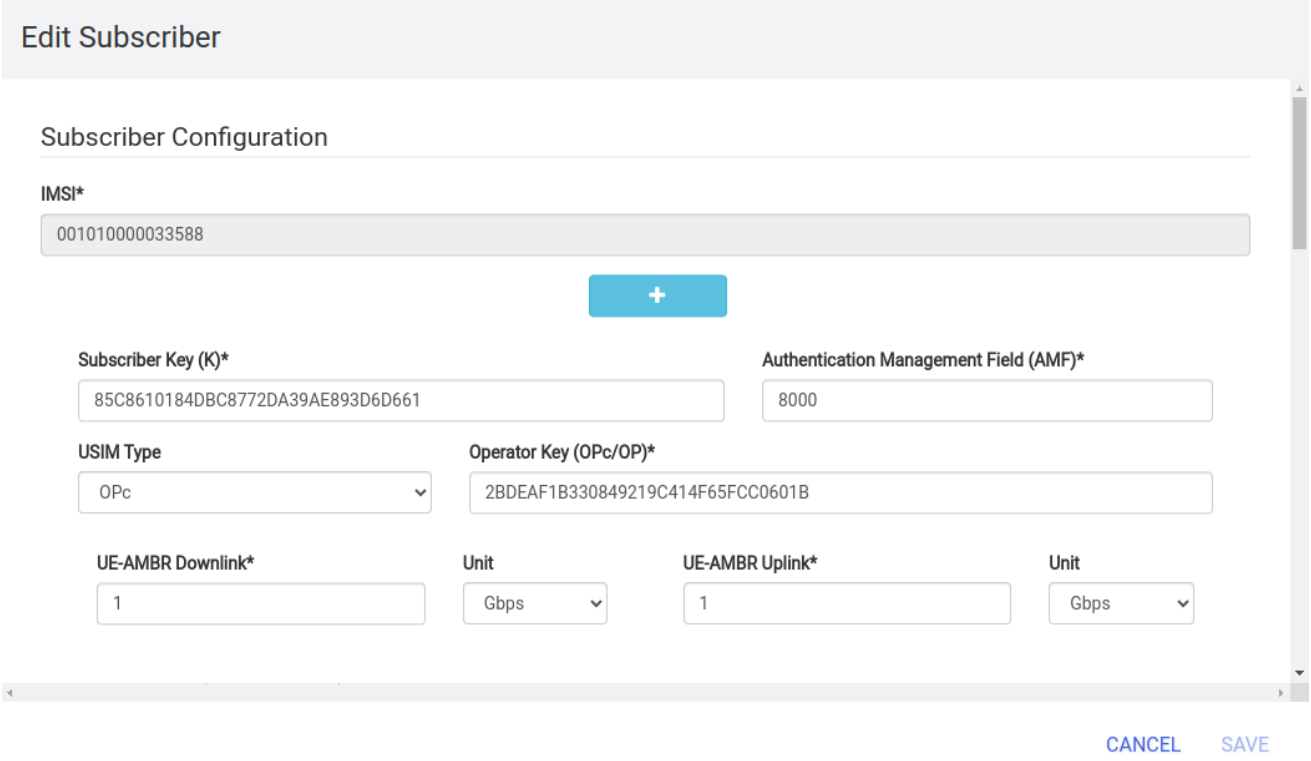

Figure 4.4. Adding subscriber data to database

Figure 4.5 shows two essential session parameters that must be configured, namely, data network name (DNN)/access point name (APN). These settings must correspond with those configured on the UE, including its type, which, in our case, is internet protocol Version 4 (IPv4). Furthermore, retaining the default settings for the other parameters is important to note.

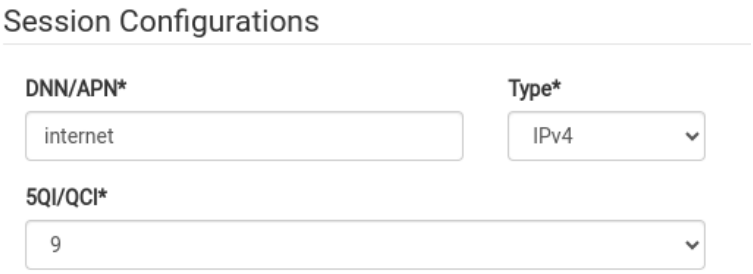

Figure 4.5. APN and IP information database

Another key step is to change the *sample.yaml* file located in the Open5Gs directory. It is critical to change the values of the mobile country code/mobile network code (MCC/MNC) parameters of the public land mobile network (PLMN) to match those of the UE IMSI and tracking area code (TAC). In this experiment, we set the MCC to 001, TAC to 1 and MNC to

01. It is also mandatory to configure the NG application protocol (NGAP) address to be consistent with that of the 5GC.

4.3.3 srsRAN deployment

The srsRAN is an open-source project developed for implementing the centralised unit of the radio access network that is software based (srsRAN). The srsRAN can be hosted on a Linux-based operating system and can apply the 5G SA connected to a SDR for running the distributed unit (DU) of the (srsRAN). It is critical to maximise the computing node's performance to match the USRP's real-time communication needs when deploying the srsRAN gNB/BBU (Mamushiane et al., 2023). The procedure involves deactivating hyperthreading and virtualisation technologies, enabling thread priority scheduling, adjusting socket-buffer sizes, modifying ethernet maximum transmission unit (MTU) values, and configuring network interface card (NIC) ring buffer sizes. Furthermore, it is recommended that C-state support be disabled to prevent the 5GNR from entering sleep mode; and using a low-latency kernel can reduce overhead while improving efficiency (Mamushiane et al., 2023).

4.3.4 SANReN backhaul deployment

Our proposed model is implemented on a world-class SDN-based wide area network (WAN) known as the South African research network (SANReN), which is managed by the CSIR. The SANReN network was used to demonstrate our proposed model in a developing market environment. This backhaul connects Open5GS 5GC and srsRAN 5GNR.The SANReN WAN is made up of seven (7) nodes spread throughout South Africa; and linked together via fibre optic cables.

As depicted in Figure 4.6, SANReN WAN functions as the backbone, with each point of presence linking the metropolitan area networks (MANs). The topology data set was obtained from the internet topology zoo (Knight et al., 2011). The data set is in geography markup language (GML) format. The geographical locations of each node and the topological configuration of the SANReN network are contained within the GML file. In addition, this backhaul serves as our edge in this study.

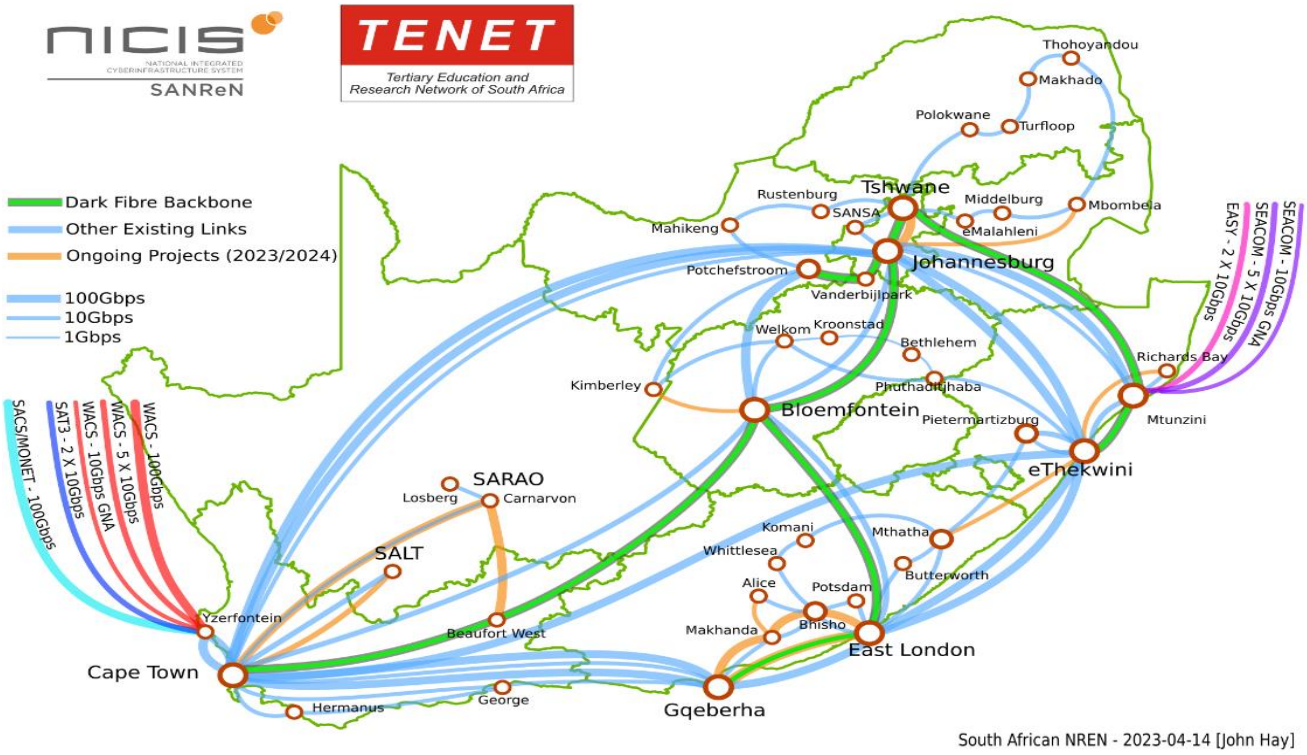

Figure 4.6. SANReN geographical map (SANREN)

The SDN-based backhaul network is emulated using Mininet. Mininet is a network-orchestration tool that is often used to create realistic virtual networks on a single machine (virtual, cloud or local) for rapid prototyping of OpenFlow- and SDN-related solutions. Furthermore, Mininet runs OpenFlow controllers, and simulates the SDN data plane using OpenFlow virtual kernel switches, connections, and end hosts. Mininet hosts can execute the underlying Linux and file system operations (Team, 2017). A Python code is used to define the network topology, configure switches and hosts, and establish connection with the OpenFlow controller. Figure 4.7 displays the proposed experiment set-up with the ONOS controller. The edge UPF and the srsRAN are connected to one switch that is in Pretoria, and the Open5GS core is connected to the switch in Cape Town. The diagram was designed using Miniedit. Miniedit is a simple Python software that provides a graphical user interface (GUI) for creating and managing various network topologies. This is an experimental toolkit designed to show how Mininet may be extended to build a network and accelerate simulation. Miniedit is a minimal user interface (UI) that consists of a canvas with a row of small tool icons on the left side of the window, and a menu bar across the top (Romanov et al., 2022).

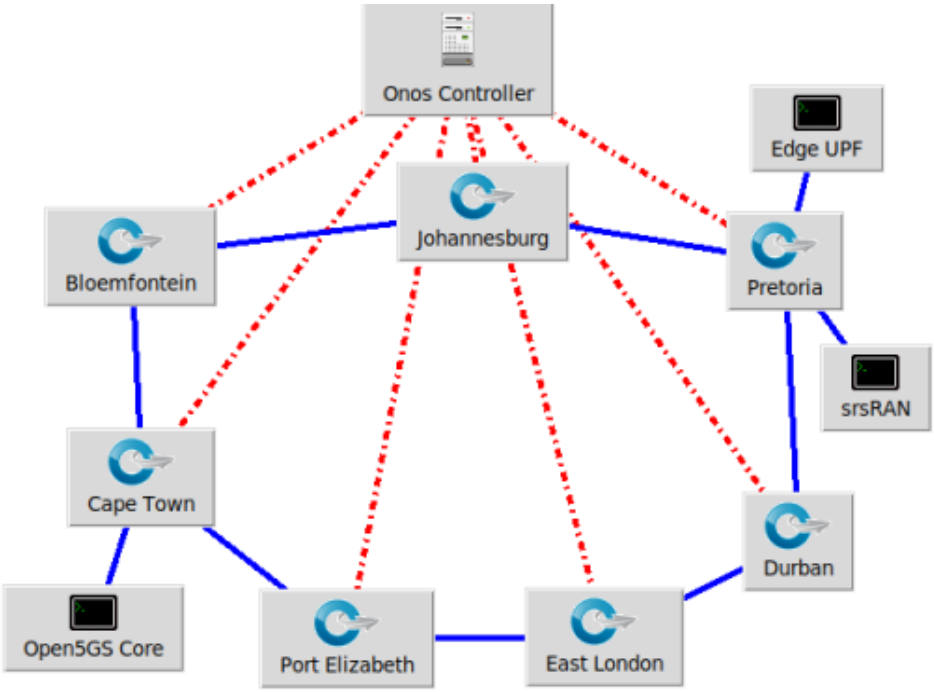

Figure 4.7. SANReN with ONOS controller

In this experiment, we deployed the ONOS controller, which is an open-source distributed SDN controller developed by ON.Lab. ONOS's northbound interface incorporates two abstraction frameworks known as the intent framework and the global network topology view, tailored for scalability, dependability, and best performance in real-world network scenarios (Kumar et al., 2016). Figure 4.8 displays the topology of the proposed backhaul on the ONOS WebUI. The green lines show the live traffic between Open5GS, srsRAN, and the proposed edge UPF when the UE is connected to the 5G private network. In this topology, the shortest path is used to send data between the source and destination.

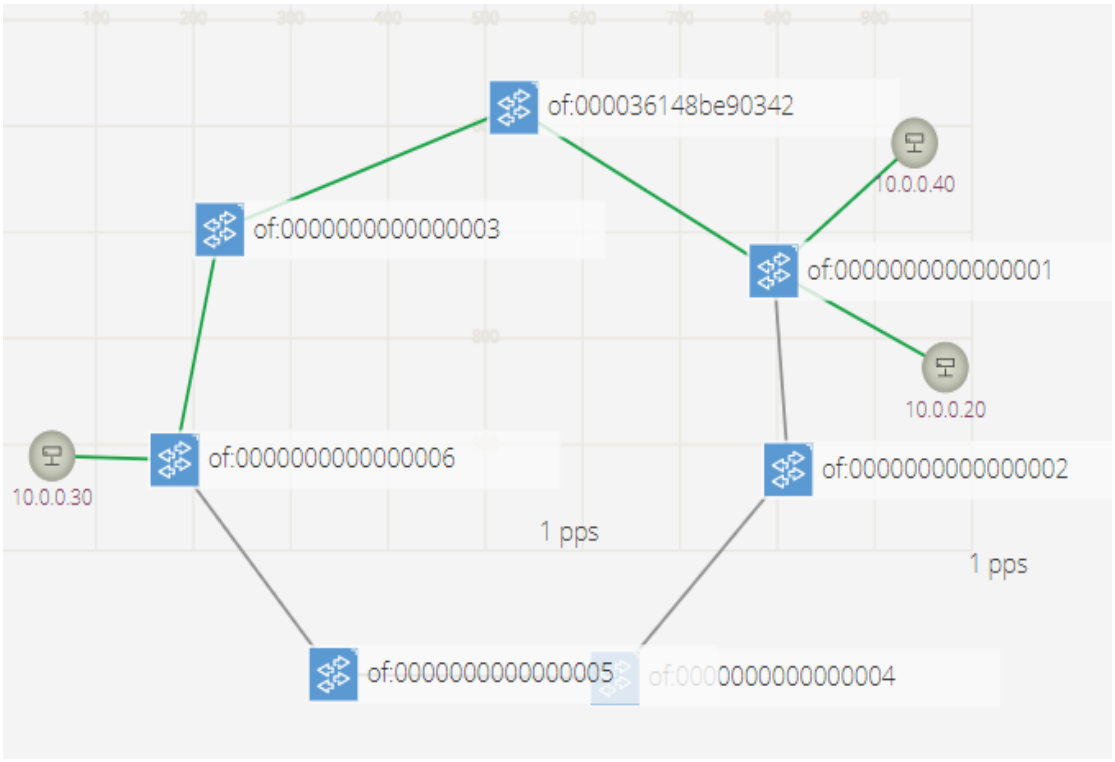

Figure 4.8. Proposed topology over ONOS GUI

The ONOS controller enhances the resilience of the control plane by providing backup control in case of network failures. Additionally, its distributed core is self-coordinating and enables load-sharing through data-plane fragmentation. The ONOS controller also features an improved east/westbound interface to facilitate effective communication between controllers (Mamushiane, 2019b). The UPF is then strategically placed in the backhaul network for optimisation of latency for latency-sensitive applications in smart farming.

4.3.5 UE deployment

Throughout our testing, we used Samsung A34 as the UE, which performed well in our test environment. The UE communicates with the internet through a data-plane function, the UPF, that masquerades as a GTP tunnel.

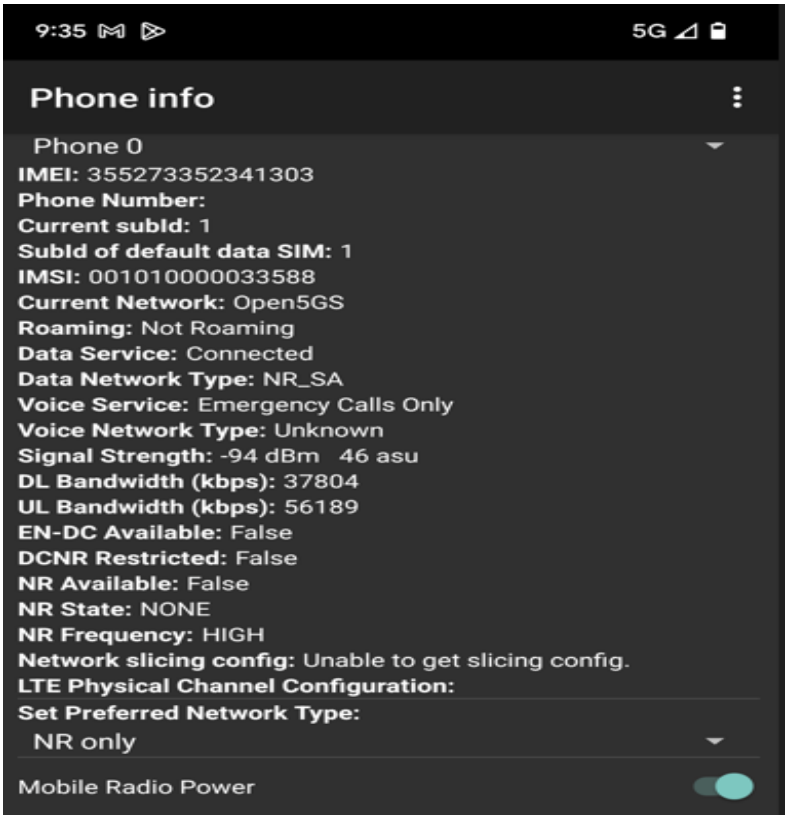

Figure 4.9. 5G information connection on the UE

Figure 4.9 shows the connection information after the UE has connected to the 5G private SA taken from the 5G smart switch android application. The 'phone information' feature of the 5G switch application provides users with a detailed overview of their device's network connectivity status. This contains information such as the current network type, e.g., 5G, and signal strength, which provides users with useful information about their network coverage. Furthermore, the application indicates the exact network to which the user UE is connected, in this case the Open5GS network, giving users more information about their present network environment. In addition, the application displays information on the status of the subscriber identity module (SIM) card, such as whether it is inserted and active. Crucially, users can obtain IDs such as the international mobile equipment identification (IMEI) and IMSI numbers, which uniquely identify their device and SIM card, respectively.

4.3.5.1 Programming the UE's SIM card

In this experiment, Sysmocom commercial-off-the-shelf (COTS) SIM cards (SJSl and SJA2) were used. Sysmocom SIM cards meet 3GPP standards, such as the 3GPP MILENAGE authentication procedure. These cards were reprogrammed to synchronise authentication and

operator-specific parameters with our Open5GS 5GC network set-up (Mamushiane et al., 2023). Pysim was used to programme the SIM cards. Pysim is a popular application for reading and writing SIM cards (Uchenna et al., 2019).

4.3.6 UE-to-internet connection establishment

Configuring the public land mobile network (PLMN) and network slice selection assistance information (NSSAI) settings alongside the Open5GS 5GC is necessary in establishing a complete end-to-end connection. It is critical to ensure that the chosen PLMN values are available and have not been assigned by the mobile network carriers. Both srsRAN and the Samsung UE must allow access to Open5GS 5GC. The 5GNR and 5GC exchange data and control messages through a specialised private network. Meanwhile, communications between the UE and the 5GC involve using 5GNR private networks to control traffic, and GTP tunnels to handle data traffic. The reception of a successful 'PDU Session Establishment Response' message from the Open5GS 5GC network indicates the creation of a GTP tunnel. Once a PDU session has been created, the UE gains access to internet services. Activating IP forwarding and applying network address translation (NAT) rules to IP tables are necessary procedures for connecting the 5GC to the internet. Failure to do so will result in the UE not receiving internet resources (Mamushiane et al., 2023), as depicted in Figure 4.10.

```
upf@upf-M11U-OPS:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
upf@upf-M11U-OPS:~$ sudo iptables -t nat -A POSTROUTING -s 10.45.0.0/16 -o ogstun -j MASQUERADE
upf@upf-M11U-OPS:~$
```

Figure 4.10. IPV4 forwarding over the NAT

4.4 Performance Testing Tools

4.4.1.1 Iperf3

IPerf3 is a tool for testing local area network (LAN) and wireless local area network (WLAN) speed and throughput. IPerf3 operates through generating traffic from a client computer and routing it to the IP address of a server machine. Transmission control protocol (TCP) tests are then used to measure bandwidth, resulting in a precise estimate of the network speeds achieving. IPerf3 can also be used to assess the quality of a link or to measure latency, jitter, and datagram loss (Zieliński, 2023).

4.4.1.2 Ping

Ping is a measure of how long it takes a data packet to go from one computer to another and back to the original computer. This statistic is frequently used in networking and online games to assess latency or delay in communication between machines. Ping is measured in milliseconds (ms) and represents the packet's round-trip time. A lower ping suggests faster and more responsive communication, which is especially useful in real-time applications such as online gaming, video conferencing, and other interactive online activities. High ping can result in delays and delayed experiences, which are uncomfortable for consumers who require real-time interaction (Gautam et al., 2022).

4.4.1.3 Wireshark

This is a robust network packet analyser that enables users to capture and view traffic on a computer network. Wireshark contains precise information on the data packets being transferred across the network, such as their source and destination addresses, protocols, and content. Wireshark's capacity to record and analyse network traffic makes it an essential tool for network administrators, security experts, developers, and anybody else who needs to troubleshoot or monitor network activities. Its open-source nature and free availability make it accessible to a diverse range of users, democratising the ability to effectively analyse network data (Soepeno, 2023).

4.4.1.4 Open signal speed test application

Open signal is a smartphone application that allows users to accurately monitor the performance of their mobile networks. Open signal provides information on various elements of network performance, including speed, video-streaming quality, and latency. Open signal collects real-time data from users all over the world to provide users with a true knowledge of their network experience, rather than relying on estimates or best-case scenarios. This information is useful for consumers seeking a mobile network provider, as well as individuals who want to analyse and improve their mobile-connectivity experience (opensignal).

4.5 Summary

This chapter provided a procedural framework for developing a 3GPP-compliant 5G testbed using open-source software tools such as srsRAN, SDN-based backhaul with ONOS controller, and Open5GS. The next Chapter 5, presents the study's findings.

CHAPTER 5 RESULTS

5.1 Introduction

In this chapter, the results of the experiments conducted in Chapter 4 are presented. A series of experiments was carried out to evaluate the impact of implementing UPF on the edge along the mobile backhaul rather than on the cloud. These experiments aimed to evaluate the performance of deploying the UPF node on the edge of the network closer to the data sources. The performance of the proposed model is evaluated using network metrics, namely, throughput, RTT, and jitter.

The chapter opens with an overview of the 5G testbed evaluation in Section 5.1. This is followed by the presentation of the results in Section 5.2; and the summary is presented in Section 5.3.

5.2 Evaluation of the 5G Testbed

As stated in Section 4.2 of Chapter 4, the experiments were carried out on the 5G testbed consisting of Open5GS core network, SDN backhaul, srsRAN 5G 5GNR, and 5G UE. i.e., Samsung A34 smartphone. The Open5GS server is connected to a separate SDN switch located in Cape Town; while the UPF node and the srsRAN servers are connected to a different SDN switch located in Pretoria. The UPF node of the data plane is deployed on the edge of the network for propagation latency optimisation; while the CP functions remain on the central Cloud.

In this study, throughput, RTT and jitter metrics are used to evaluate the performance of proposed model. Throughput is the amount of data that can be sent over a network in each period of time; and is typically measured in bits per second (b/s) or megabits per second (Mb/s) (Iranmanesh et al., 2021). RTT refers to the time it takes for a network request originating from a UE to travel to its destination on the internet, then return to the UE. This metric is measured in microseconds; and is an important indicator for determining the speed and stability of connections within local networks and on the internet (Martinez et al., 2023). Yusuf (2023) defined jitter in networking as the variation in the time delay between packet arrivals caused by network congestion or route changes. Throughput and RTT are inversely related: high throughput results in lower RTT, and vice versa. Cruz (2023) used throughput, RTT and jitter to determine the performance of their proposed model for reducing latency in 5G networks.

Thus, these metrics were considered in this research study when evaluating the proposed solution. Some of the factors that can influence throughput and latency in 5G networks include network congestion, packet loss, link bandwidth, and the distance between UE and the internet (Iranmanesh et al., 2021).

Parameter values are numerical examples that represent as many realistic deployment scenarios as possible, as depicted in Table 5.1. However, depending on the situation that needs to be presented, any other value can be used. In this experiment, processing, queueing, and transmission delays are kept constant.

Table 5.1. Experiment Parameters

Parameter	Value
Base-station processing time	200 µs
Cell radius	6 m
Packet size	1500 Bytes
Capacity of each link between SDN switches	10 Gb/s
Transmission delay	200 µs
Node-processing delay	200 µs
Queuing delay	200 µs
SDN Controller-processing time	200 µs

5.3 Presentation of Results

This section presents the results that were obtained after evaluating the two UPF deployment scenarios in 5G networks, i.e., Cloud UPF and Edge UPF. First, this study measures the throughput between the SDN switches in the backhaul that connects the srsRAN, the UPF node, and the Open5GS servers. Furthermore, throughput, RTT, and jitter are measured between the UE and the internet in both scenarios. The comparison between the two UPF deployment scenarios seeks to provide information on the latency characteristics of each deployment scenario, which will help in selecting the best architectural options for the 5G uRLLC applications. The experimental results are presented and discussed in detail from the next paragraph.

The experimental results in Figure 5.1 show that the use of 10 Gb/s fibre links in the network infrastructure significantly improves throughput. The higher bandwidth provided by these links allows for more efficient data transmission, resulting in an average throughput of 9.27 Gb/s. Factors contributing to this high throughput include increased bandwidth, efficient network infrastructure, optimised protocols, and effective QoS management. These findings underscore the importance of high-speed fibre optic links in improving data transfer rates and overall network performance. The SDN improves traffic efficiency by enabling centralised control, dynamic traffic engineering, programmability, and automation. These capabilities enable networks to adapt to changing traffic patterns and optimise resource usage, ultimately improving overall network performance and user experience.

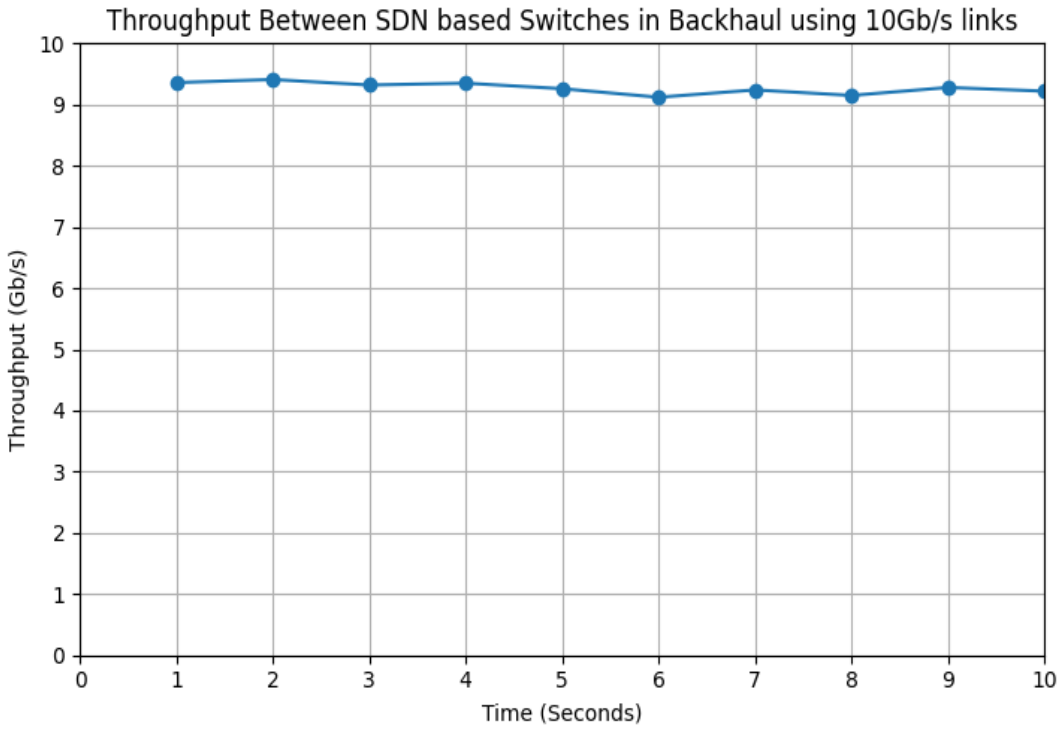

Figure 5.1. Throughput between the SDN-based switches in the backhaul

Figure 5.2 shows the throughput between srsRAN and Open5GS. The experimental findings showed that the obtained average throughput is 933 Mb/s. The reason for this difference is that the Ethernet interfaces for the srsRAN, UPF node, and Open5GS servers are limited to 1 Gb/s. This limitation poses a constraint on the study, as 10 Gb/s traffic from the backhaul is throttled to just 1 Gb/s on the servers due to the unavailability of 10 Gb/s interfaces.

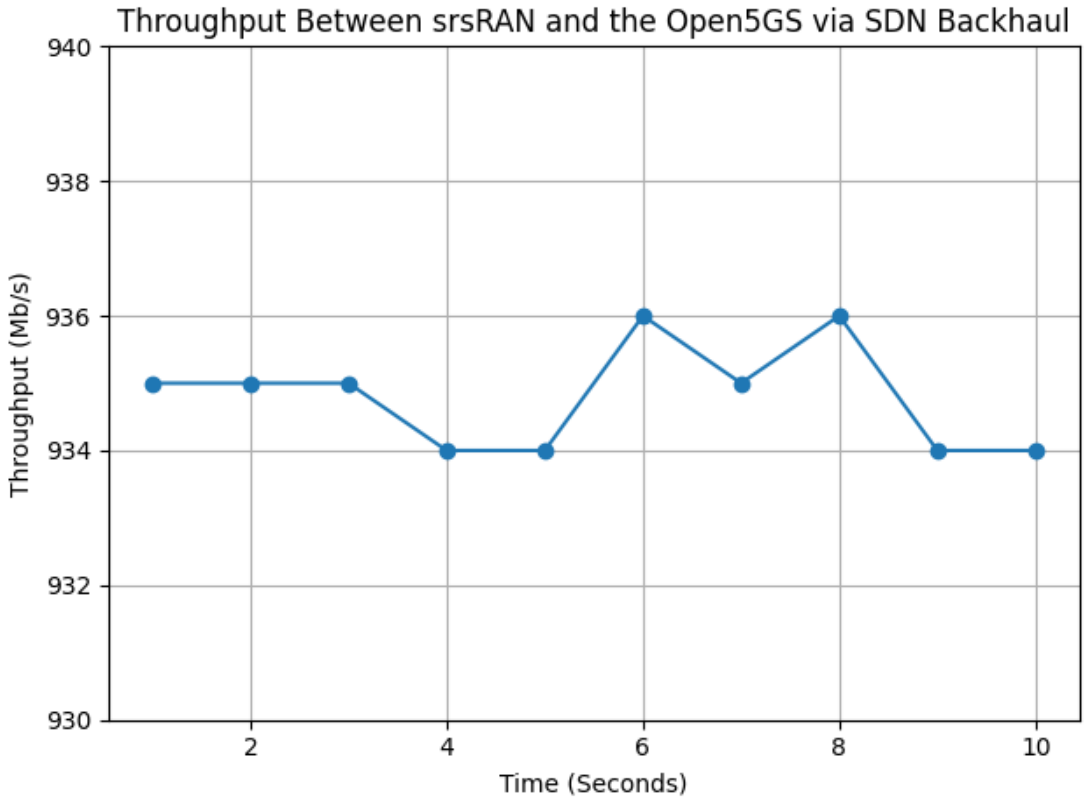

Figure 5.2. The throughput between the srsRAN and the Open5GS

In terms of throughput between the UE and internet, the edge UPF reaches 59.0 Mbps downlink and 18.8 Mbps uplink speeds, demonstrating its capacity to efficiently transport data. By contrast, the cloud UPF, although it has significantly lower throughput than the edge UPF, still delivers good speeds, with a downlink speed of 42.6 Mb/s and an uplink speed of 8.56 Mb/s, as depicted in Figure 5.3. These throughput values demonstrate the solid connectivity provided by both implementations. Furthermore, the latency factor is critical to determining real-time responsiveness and user experience. Edge UPF has low latency, measured at 16 ms, making it ideal for applications that require minimal delays and real-time interactions. Edge UPF's proximity to end users allows for efficient data processing, resulting in lower latency. Although experimental results demonstrated that cloud UPF offers competitive throughput, cloud UPF produces a little higher latency of 30 ms. However, this latency is still tolerable for many applications, while not optimal for latency-sensitive use cases.

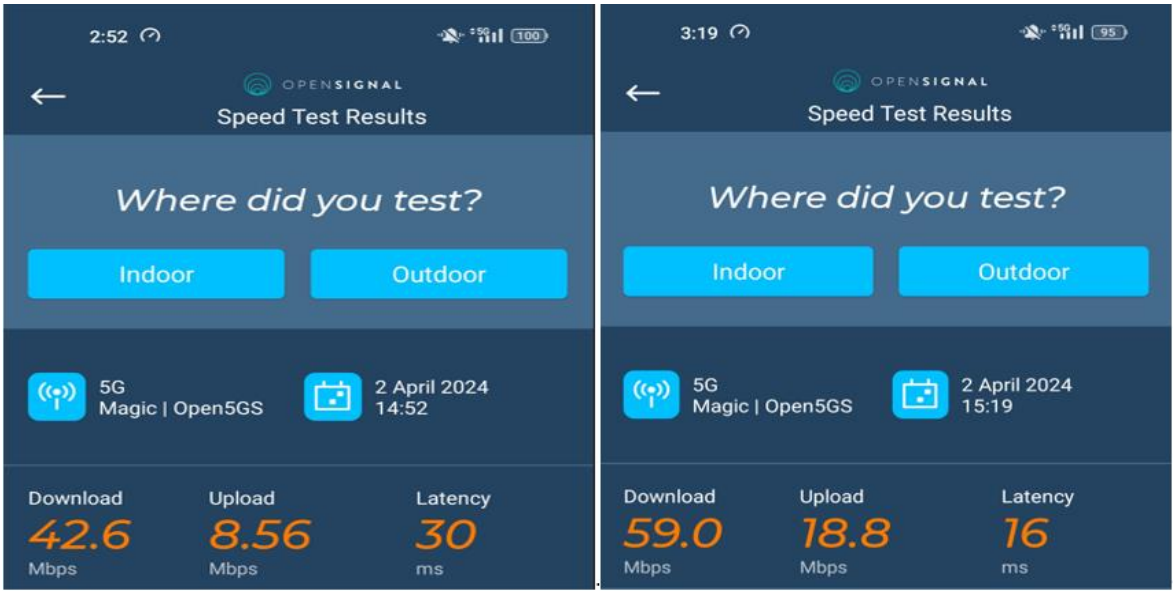

Figure 5.3. Latency and throughput comparison between cloud UPF (left) and edge UPF (right)

Finally, the deployment factors play an important role in assessing the viability of each UPF deployment. The edge UPF, when placed closer to end users, provides advantages in terms of reduced latency and improved user experiences. The proximity of edge UPF allows for more efficient data processing and shorter response times, which is especially useful for applications that require real-time responsiveness. By contrast, the cloud UPF, which features centralised processing, provides scalability and centralised management benefits. It can efficiently manage massive amounts of data and dynamically assign resources based on demand, making it ideal for scenarios with changing data requirements.

Figure 5.4 shows the latency comparison between edge UPF and cloud UPF. The experimental results demonstrate that placing UPF on the edge of the network using the CUPS approach is ideal for latency optimisation in 5G SA network deployment. These experimental tests were conducted 10 times under each scenario to ensure statistical consistency; and the results were compared.

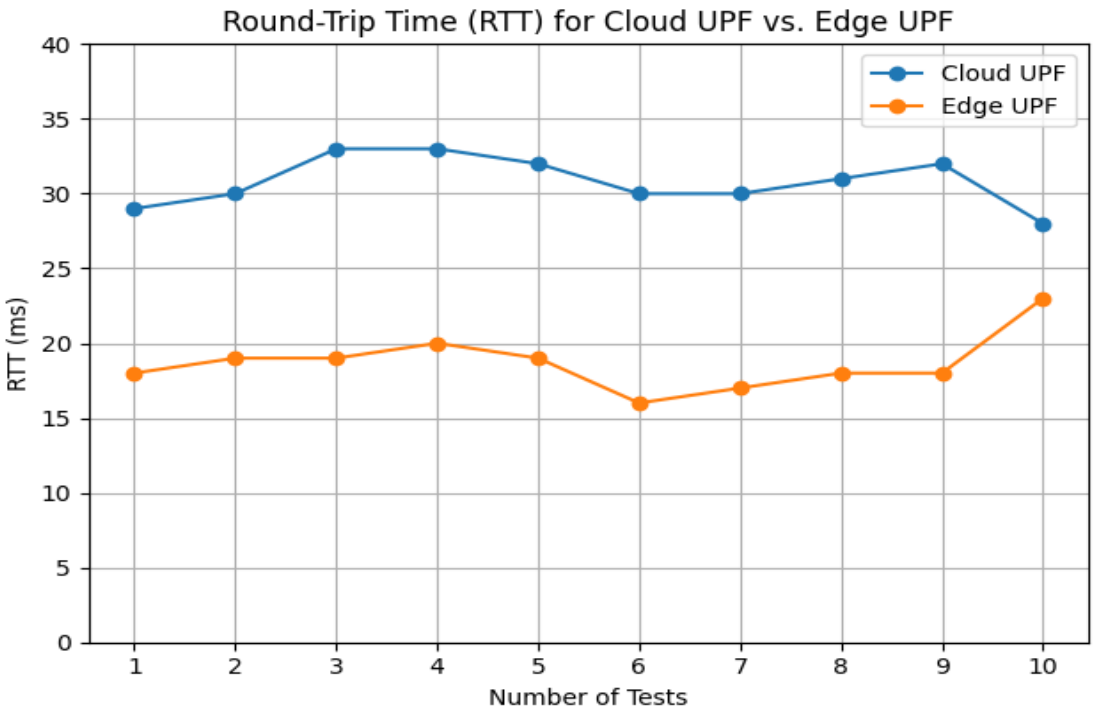

Figure 5.4. RTT Comparison between cloud UPF and edge UPF

Figure 5.5 shows the RTT comparison between cloud UPF and edge UPF installations. It is evident that the cloud UPF deployment has a greater average RTT than the edge UPF set-up. This disparity reflects the architectural differences between the two configurations. Cloud UPF implementations, which use centralised data processing and routing, entail more network hops and potential congestion in data-centre networks, resulting in greater RTT. Edge UPF solutions, on the other hand, use local processing at the network edge to reduce network hops, and hence achieve reduced RTTs. Furthermore, the UPF installed at the edge establishes a local internet breakout, substantially lowering latency by enabling direct access to internet resources without passing through centralised data centres. The experiment indicates that the UPF can be isolated and deployed at the network edge without degrading performance. Furthermore, the results demonstrated that with an efficient edge UPF, user plane traffic can be offloaded as a local breakout during network congestion, rather than being transmitted to the cloud UPF. The average RTT for placing UPF on the edge is 18.7 ms and 30.7 ms for cloud UPF.

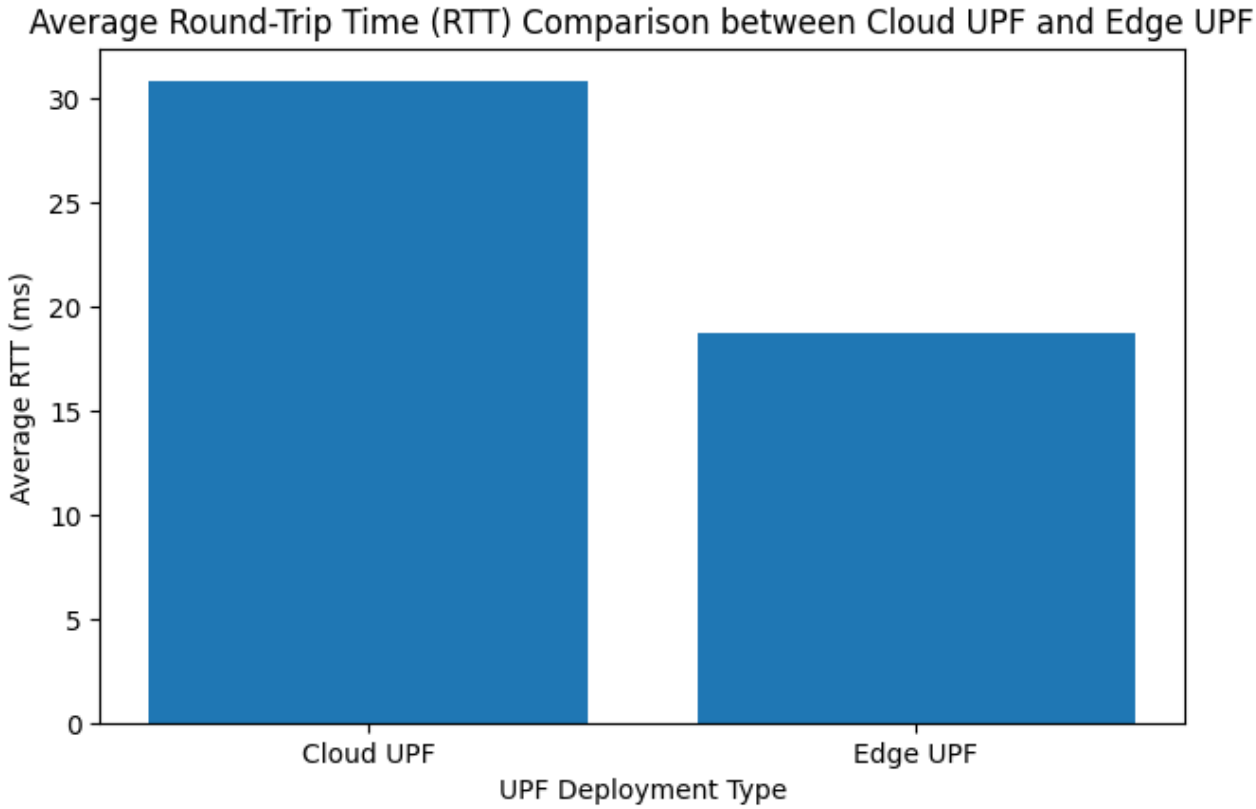

Figure 5.5. The Average RTT comparison between cloud UPF and edge UPF

Figure 5.6 shows the jitter values from the RTT results, revealing the major performance differences between the cloud UPF and edge UPF implementations. The cloud UPF's computed average jitter of roughly 3.67 ms suggests moderate latency variability, reflecting occasional fluctuations in latency performance. While the cloud UPF's latency characteristics are generally constant, the measured jitter values indicate a less predictable network environment than the edge UPF. On the contrary, the edge UPF has significantly lowered average jitter, around 1.56 ms, indicating greater consistency and predictability in latency performance. The lower jitter values between consecutive RTT measurements show less unpredictability and more consistency in latency, demonstrating the edge UPF's advantage in guaranteeing a dependable and responsive network infrastructure. These findings highlight the benefits of using the edge UPF architecture, especially in latency-sensitive applications in which constant performance is critical to providing excellent user experiences.

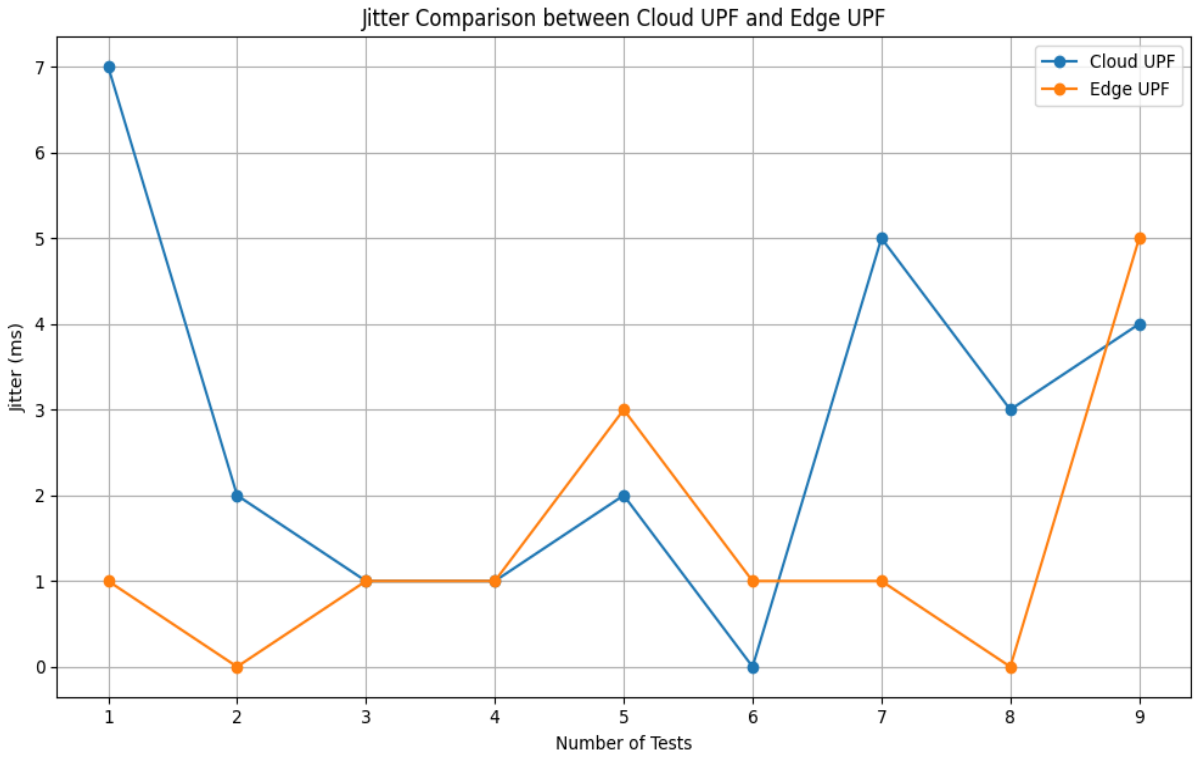

Figure 5.6. Jitter between cloud UPF and edge UPF

5.4 Summary

Our extensive experiments and analysis demonstrated that edge UPF deployment in 5G networks can support ultra-low latency for uRLLC use cases while also revealing the sensitivity of QoS to network dynamics such as throughput fluctuations. uRLLC latency requirements for both planes are low enough to enable the development of latency-critical applications in an optimal manner. Latency-sensitive systems can be built using the uRLLC use case of 5G wireless communication technology. Therefore, mobile network operators in South Africa and in the rest of Africa should consider deploying the UPF on the edge of the network to fully realise the benefits of 5G networks.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 Introduction

This study showed that the integration of IoT, 5G, edge computing, and SDN is critical for influencing the future of telecommunications and network architecture. 5G networks provide the high-speed, low-latency connectivity required for IoT devices to effectively transport data, allowing real-time communication. This connectivity extends to edge nodes that process data near its origin, lowering latency and improving response times for latency-sensitive applications. SDN improves 5G networks by centralising network administration and programmability, resulting in greater agility and flexibility to respond to changing demands. These technologies work together to create a cohesive ecosystem that fosters innovation and enables the deployment of advanced use cases such as self-driving cars, smart cities, and industrial automation.

This dissertation focused on latency optimisation in a smart farming use case with a 3GPP RTT delay of 20 ms. Latency optimisation was discussed in this dissertation because it was of interest to a telecommunications operator, there being many estimates for latency reduction in studies that lack real testing. An examination of existing research publications found that there are just a few studies covering uRLLC use cases that consider the entire 5G system. The objective was to explore 5G networks, determining whether an edge UPF is truly required for the use case, or whether it can be realised using cloud UPF. This concept shifts the UPF from the cloud to the network edge to overcome the high latency issue, which has been a major impediment to applications sensitive to IoT latency in smart farming. Several attempts to address these difficulties have proven ineffective over the years.

The remainder of the chapter is organised as follows. In Section 6.,2 the research study discusses how the technical objectives were achieved. The conclusion of the research is presented in Section 6.3. Future work is presented in Section 6.4, and the chapter summary is presented in Section 6.5.

6.2 Accomplishment of Research Objectives

This section of the research study examines how the set of aims and objectives were met, as outlined in Section 1.5 of Chapter 1.

6.2.1 To identify the factors that affect the latency for 5G-enabled IoT applications in the smart farming environment.

The review of literature conducted and presented in Chapter 2 indicated that the distance between the UPF and the UE influences the latency encountered by the UE when communicating with the internet in 5G networks. As a result, the position of the UPF is a critical determining factor of latency. The UPF acts as a vital intermediary, managing data-packet routing and processing between the UE and the internet. Its geographical location has a direct impact on the distance that data packets must travel, thus influencing latency. Placing the UPF closer to the UE can reduce the distance and network hops, resulting in lower latency and a more rapid response for UE applications and services.

6.2.2 To design the proposed model by leveraging the edge computing to minimise latency for 5G-enabled IoT applications in smart farming.

In Chapter 3, the proposed solution was designed to reduce latency for 5G-enabled IoT applications in smart farming using EC. This was accomplished by carefully placing the UPF at the network's edge, while other critical 5GC network operations remained housed in the cloud. This solution made use of the notion of EC, which allows computational operations and data processing to be conducted closer to the source of data collection and consumption. Data packets from IoT devices in smart farming could be processed and routed more efficiently by delegating computational activities to the network edge, resulting in lower latency and better real-time capabilities.

6.2.3 To implement the proposed model by offloading the UPF to the edge of the network.

In Chapter 4, a detailed 3GPP-compliant experimental set-up was used to realise the proposed model of offloading the UPF to the network's edge. This configuration used srsRAN as the 5GNR, Open5GS as the 5GC network architecture, an SDN-based backhaul, and Samsung A34 as the UE. Furthermore, both srsRAN and edge UPF were connected to an SDN switch in Pretoria, while the core network was connected to a different switch in Cape Town.

6.2.4 To integrate the 5GC and the 5GNR using an SDN-based backhaul network.

In Chapter 4, the Mininet emulator was used to establish a virtual SANREN network with strategic nodes in Cape Town and Pretoria, integrating 5GC and 5GNR. The ONOS SDN

controller handled these switch-configured nodes to provide dynamic data flow between the 5GC and 5GNR. The critical connection between these switches was established using high-speed 10 Gb/s fibre links, ensuring robust capacity to handle the demanding throughput and low latency requirements of 5G communications. This configuration demonstrated SDN's ability to successfully manage and optimise high capacity backhaul networks in a 5G environment.

6.2.5 To demonstrate through experimental evaluations that offloading the UPF to the edge does reduce latency for 5G-enabled IoT applications in smart farming.

In Chapter 5, the experimental tests were carried out on a 3GPP-compliant 5G testbed to evaluate the proposed model under two different scenarios: using a cloud-based UPF and an edge UPF. Experimental results showed that deploying the UPF at the network edge using the CUPS strategy reduced latency in a 5G standalone network. The tests were run ten times for each scenario to ensure statistical consistency; and the average RTT was measured. The edge UPF had an average latency of 18.7 ms, compared with 30.7 ms for the cloud based UPF. These findings show that the use of edge UPF resulted in a 60.7% reduction in latency compared with cloud UPF. It was also noted that higher throughput was achieved after shifting the UPF to the edge. This is because the network hops between UE and the internet were reduced.

6.3 Conclusion

The study focused on optimising latency for 5G-enabled IoT applications in smart farming contexts. The study presented a concept that takes advantage of edge computing by moving the UPF to the network's edge. The study used a 3GPP-compliant 5G testbed to compare the latency performance of a cloud-based UPF versus an edge UPF. When the UPF was placed at the edge, the latency was reduced by 60.7%, thereby improving the throughput by approximately 40.48%, demonstrating the efficiency of the proposed model. This study emphasises the role of edge computing in improving network performance and user experience for latency-sensitive applications in 5G deployments, particularly in smart farming scenarios.

6.4 Future Works

The edge UPF is still in the development stages; therefore, there is much work that must still be conducted, as highlighted in Chapter 2. The actual application of the edge UPF in real-world networks has been identified as one of the problems in telecommunication. Therefore, this

allows for further testing of the proposed model on real network testbeds by mobile network operators. This model requires more testing for scalability in a large network. The scalability of the ONOS controller is another aspect that must be tested in the future. Furthermore, it is recommended to use 10 Gb/s links instead of 1 Gb/s to connect the 5GNR and 5GC to the SDN switches, for better throughput.

6.5 Summary

This chapter discussed ways in which the objectives of the research study were accomplished. This research study considers that its objectives were successfully met. Furthermore, this study recommends future research, with a focus on testing the performance of the proposed model in a large environment.

References

3GPP. (2018). System architecture for the 5G system. *3GPP TS 23.501 V15. 3.0*.

Admire, M. (2022). *MTN, Huawei power Southern Africa's first 5G metal mine*. Retrieved 10/02/2024 from https://www.itweb.co.za/content/G98YdMLGojZ7X2PD

Ahmad, F. (2023). *Data collection for Machine Learning in 5G Mobile Networks* Oslomet-storbyuniversitetet].

Ahmad, S., & Mir, A. H. (2021). Scalability, consistency, reliability and security in SDN controllers: a survey of diverse SDN controllers. *Journal of Network and Systems Management*, *29*, 1-59.

Ahmadi, S. (2019). 5G network architecture. *5G NR*, 1-194.

Akpakwu, G. A., Silva, B. J., Hancke, G. P., & Abu-Mahfouz, A. M. (2017). A survey on 5G networks for the Internet of Things: Communication technologies and challenges. *IEEE Access*, *6*, 3619-3647.

Al-Quzweeni, A. N., Lawey, A. Q., Elgorashi, T. E., & Elmirghani, J. M. (2019). Optimized energy aware 5G network function virtualization. *IEEE Access*, *7*, 44939-44958.

Ali, A., Hussain, T., Tantashutikun, N., Hussain, N., & Cocetta, G. (2023). Application of Smart Techniques, Internet of Things and Data Mining for Resource Use Efficient and Sustainable Crop Production. *Agriculture*, *13*(2), 397. https://www.mdpi.com/2077-0472/13/2/397

Ali, R., Zikria, Y. B., Bashir, A. K., Garg, S., & Kim, H. S. (2021). URLLC for 5G and beyond: Requirements, enabling incumbent technologies and network intelligence. *IEEE Access*, *9*, 67064-67095.

Ali, T. A., Choksi, V., & Potdar, M. B. (2018). Precision agriculture monitoring system using green internet of things (g-iot). 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI),

Alliance, N. (2021). Cloud Native Enabling Future Telco Platforms. In: May.

Alotaibi, D. (2021). Survey on network slice isolation in 5g networks: Fundamental challenges. *Procedia Computer Science*, *182*, 38-45.

Amemiya, K., & Nakao, A. (2021). Enabling Delay-Sensitive IoT Application by Programmable Local 5G Edge. 2021 IEEE 10th International Conference on Cloud Networking (CloudNet),

Amponis, G., Radoglou-Grammatikis, P., Nakas, G., Goudos, S., Argyriou, V., Lagkas, T., & Sarigiannidis, P. (2023). 5G core PFCP intrusion detection dataset. 2023 12th International Conference on Modern Circuits and Systems Technologies (MOCAST),

Antao, L., Pinto, R., Reis, J., & Gonçalves, G. (2018). Requirements for testing and validating the industrial internet of things. 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW),

Ateya, A. A., Muthanna, A., Makolkina, M., & Koucheryavy, A. (2018). Study of 5G services standardization: specifications and requirements. 2018 10th international congress on ultra modern telecommunications and control systems and workshops (ICUMT),

Attaran, M. (2023). The impact of 5G on the evolution of intelligent automation and industry digitization. *Journal of Ambient Intelligence and Humanized Computing*, *14*(5), 5977-5993.

Ayoub, W., Samhat, A. E., Nouvel, F., Mroue, M., & Prévotet, J.-C. (2018). Internet of mobile things: Overview of lorawan, dash7, and nb-iot in lpwans standards and supported mobility. *IEEE Communications Surveys & Tutorials*, *21*(2), 1561-1581.

Azari, A., & Masoudi, M. (2021). Interference management for coexisting Internet of Things networks over unlicensed spectrum. *Ad Hoc Networks*, *120*, 102539.

Basu, D., Datta, R., & Ghosh, U. (2020). Softwarized network function virtualization for 5g: Challenges and opportunities. *Internet of Things and Secure Smart Environments: Successes and Pitfalls (2020)*, *147*.

Berckmans, D. (2017). General introduction to precision livestock farming. *Animal Frontiers*, *7*(1), 6-11.

Berzin, O. (2021). WONDER: Workload Optimized Network Defined Edge Routing. *arXiv preprint arXiv:2110.09563*.

Biral, A., Centenaro, M., Zanella, A., Vangelista, L., & Zorzi, M. (2015). The challenges of M2M massive access in wireless cellular networks. *Digital Communications and Networks*, *1*(1), 1-19.

Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An overview on edge computing research. *IEEE Access*, *8*, 85714-85728.

Cao, S., Chen, X., & Yuan, B. (2022). Overview of short-range wireless communication protocol. 2022 7th International Conference on Computer and Communication Systems (ICCCS),

Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. *Proceedings of the IEEE*, *107*(8), 1655-1674.

Choi, Y.-i., & Park, N. (2017). Slice architecture for 5G core network. 2017 Ninth international conference on ubiquitous and future networks (ICUFN),

Condoluci, M., & Mahmoodi, T. (2018). Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges. *Computer Networks*, *146*, 65-84.

Costa-Requena, J., Poutanen, A., Vural, S., Kamel, G., Clark, C., & Roy, S. K. (2018). SDN-based UPF for mobile backhaul network slicing. 2018 European Conference on Networks and Communications (EuCNC),

Cruz, D. E. R. (2023). *Design and Development of Laboratory Infrastructure for 5G LAN Network Environments*

Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, *7*(8), 7457-7469.

Dludla, G., Vilakazi, M., Burger, C. R., Lysko, A. A., Ngcama, L., Mboweni, L., Masonta, M., Kobo, H., & Mamushiane, L. (2022). Testing performance for several use cases with an indoor OpenAirInterface USRP-Based base station. 2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT),

Dogra, A., Jha, R. K., & Jain, S. (2020). A survey on beyond 5G network with the advent of 6G: Architecture and emerging technologies. *IEEE Access*, *9*, 67512-67547.

Esmaeily, A., Kralevska, K., & Gligoroski, D. (2020). A cloud-based SDN/NFV testbed for end-to-end network slicing in 4G/5G. 2020 6th IEEE Conference on Network Softwarization (NetSoft),

Esmaeily, A., Mendis, H. K., Mahmoodi, T., & Kralevska, K. (2023). Beyond 5G Resource Slicing With Mixed-Numerologies for Mission Critical URLLC and eMBB Coexistence. *IEEE Open Journal of the Communications Society*, *4*, 727-747.

Eze, K. G., Sadiku, M. N., & Musa, S. M. (2018). 5G wireless technology: A primer. *International Journal of Scientific Engineering and Technology*, *7*(7), 62-64.

Feng, X., Yan, F., & Liu, X. (2019). Study of wireless communication technologies on Internet of Things for precision agriculture. *Wireless Personal Communications*, *108*(3), 1785-1802.

Friha, O., Ferrag, M. A., Shu, L., Maglaras, L., & Wang, X. (2021). Internet of things for the future of smart agriculture: A comprehensive survey of emerging technologies. *IEEE/CAA Journal of Automatica Sinica*, *8*(4), 718-752.

Gautam, P., Jha, S., & Yadav, C. S. (2022). Performance Comparison of the Proposed OpenFlow Network with the Pox Controller and the Traditional Network in Software Defined Networks (SDN). In (pp. 537-552): Routledge: Londun, UK.

Ghosh, A., Maeder, A., Baker, M., & Chandramouli, D. (2019). 5G evolution: A view on 5G cellular technology beyond 3GPP release 15. *IEEE Access*, *7*, 127639-127651.

Goel, N., & Yadav, R. K. (2023). *Handbook of Research on Machine Learning-Enabled IoT for Smart Applications Across Industries*. IGI Global.

Goonatilake, R., & Bachnak, R. A. (2012). Modeling latency in a network distribution. *Network and Communication Technologies*, *1*(2), 1.

Guan, W., Wen, X., Wang, L., Lu, Z., & Shen, Y. (2018). A service-oriented deployment policy of end-to-end network slicing based on complex network theory. *IEEE Access*, *6*, 19691-19701.

Haile, B. B., Mutafungwa, E., & Hämäläinen, J. (2020). A data-driven multiobjective optimization framework for hyperdense 5G network planning. *IEEE Access*, *8*, 169423-169443.

Haile, H., Grinnemo, K.-J., Ferlin, S., Hurtig, P., & Brunstrom, A. (2021). End-to-end congestion control approaches for high throughput and low delay in 4G/5G cellular networks. *Computer Networks*, *186*, 107692.

Håkegård, J. E., Lundkvist, H., Rauniyar, A., & Morris, P. (2024). Performance Evaluation of an Open Source Implementation of a 5G Standalone Platform. *IEEE Access*, *12*, 25809-25819.

Hamidi-Sepehr, F., Sajadieh, M., Panteleev, S., Islam, T., Karls, I., Chatterjee, D., & Ansari, J. (2021). 5G URLLC: Evolution of high-performance wireless networking for industrial automation. *IEEE Communications Standards Magazine*, *5*(2), 132-140.

Hassan, N., Yau, K.-L. A., & Wu, C. (2019). Edge computing in 5G: A review. *IEEE Access*, *7*, 127276-127289.

Hossein Motlagh, N., Mohammadrezaei, M., Hunt, J., & Zakeri, B. (2020). Internet of Things (IoT) and the energy sector. *Energies*, *13*(2), 494.

Huawei, H. (2016). 3GPP TSG RAN WG1 Meeting# 87. *R1-166166, Gothenburg, Sweden, Aug*, *12*, 6.

Ibrahim, A. A., Kpochi, K. P., & Smith, E. J. (2018). Energy consumption assessment of mobile cellular networks. *Am J Eng Res (AJER)*, *7*(3), 96-101.

Ikpehai, A., Adebisi, B., Rabie, K. M., Anoh, K., Ande, R. E., Hammoudeh, M., Gacanin, H., & Mbanaso, U. M. (2018). Low-power wide area network technologies for Internet-of-Things: A comparative review. *IEEE Internet of Things Journal*, *6*(2), 2225-2240.

Iranmanesh, S., Abkenar, F. S., Raad, R., & Jamalipour, A. (2021). Improving throughput of 5G cellular networks via 3D placement optimization of logistics drones. *IEEE Transactions on Vehicular Technology*, *70*(2), 1448-1460.

Jiang, X., Shokri-Ghadikolaei, H., Fodor, G., Modiano, E., Pang, Z., Zorzi, M., & Fischione, C. (2018). Low-latency networking: Where latency lurks and how to tame it. *Proceedings of the IEEE*, *107*(2), 280-306.

Jun, S., Kang, Y., Kim, J., & Kim, C. (2020). Ultra-low-latency services in 5G systems: A perspective from 3GPP standards. *ETRI journal*, *42*(5), 721-733.

Khan, F., & Pi, Z. (2011). mmWave mobile broadband (MMB): Unleashing the 3–300GHz spectrum. 34th IEEE Sarnoff Symposium,

Kharel, U. (2022). Evolution of Mobile Wireless Communication Networks to 5G Revolution.

Kiess, W., & Khan, A. (2019). Centralized vs. distributed: On the placement of gateway functionality in 5G cellular networks. 2019 IEEE Global Communications Conference,

Kimogol, S. S. (2019). *Smart Farming For Emerging Economies A LoRaWAN Based Approach*

Knight, S., Nguyen, H. X., Falkner, N., Bowden, R., & Roughan, M. (2011). The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, *29*(9), 1765-1775.

Kobo, H. I., Abu-Mahfouz, A. M., & Hancke, G. P. (2017). A survey on software-defined wireless sensor networks: Challenges and design requirements. *IEEE Access*, *5*, 1872-1899.

Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., Zeng, P., Khan, M., & Das, S. K. (2022). Edge-computing-driven internet of things: A survey. *ACM Computing Surveys*, *55*(8), 1-41.

Kumar, H., Russell, C., Sivaraman, V., & Banerjee, S. (2016). A software-defined flexible inter-domain interconnect using ONOS. 2016 Fifth European Workshop on Software-Defined Networks (EWSDN),

Kumar, N. M., & Mallick, P. K. (2018). The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. *Procedia Computer Science*, *132*, 109-117.

Kundel, R., Meuser, T., Koppe, T., Hark, R., & Steinmetz, R. (2022). User Plane Hardware Acceleration in Access Networks: Experiences in Offloading Network Functions in Real 5G Deployments. HICSS,

Kunz, A., Nkenyereye, L., & Song, J. (2019). 5g evolution of cellular iot for v2x. 2019 IEEE conference on standards for communications and networking (CSCN),

Kwabena, A. (2018). Network Slicing System Supporting Ultra Reliable Low Latency Connectivity in 5G.

Laghari, A. A., Wu, K., Laghari, R. A., Ali, M., & Khan, A. A. (2021). A review and state of art of Internet of Things (IoT). *Archives of Computational Methods in Engineering*, 1-19.

Lando, G., Schierholt, L. A. F., Milesi, M. P., & Wickboldt, J. A. (2023). Evaluating the performance of open source software implementations of the 5G network core. NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium,

Leyva-Pupo, I., & Cervelló-Pastor, C. (2022). Efficient solutions to the placement and chaining problem of User Plane Functions in 5G networks. *Journal of Network and Computer Applications, 197*, 103269.

Leyva-Pupo, I., Cervelló-Pastor, C., Anagnostopoulos, C., & Pezaros, D. P. (2020). Dynamic scheduling and optimal reconfiguration of UPF placement in 5G networks. Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems,

Leyva-Pupo, I., Cervelló-Pastor, C., Anagnostopoulos, C., & Pezaros, D. P. (2022). Dynamic UPF placement and chaining reconfiguration in 5G networks. *Computer Networks*, *215*, 109200.

Leyva-Pupo, I., Cervelló-Pastor, C., & Llorens-Carrodeguas, A. (2019). Optimal placement of user plane functions in 5G Networks. Wired/Wireless Internet Communications: 17th IFIP WG 6.2 International Conference, WWIC 2019, Bologna, Italy, June 17–18, 2019, Proceedings 17,

Leyva-Pupo, I., Santoyo-González, A., & Cervelló-Pastor, C. (2019). A framework for the joint placement of edge service infrastructure and user plane functions for 5G. *Sensors*, *19*(18), 3975.

Li, X., Garcia-Saavedra, A., Costa-Perez, X., Bernardos, C. J., Guimarães, C., Antevski, K., Mangues-Bafalluy, J., Baranda, J., Zeydan, E., & Corujo, D. (2021). 5Growth: An end-to-end service platform for automated deployment and management of vertical services over 5G networks. *IEEE Communications Magazine*, *59*(3), 84-90.

Li, Y., Ma, X., Xu, M., Zhou, A., Sun, Q., Zhang, N., & Wang, S. (2021). Joint placement of UPF and edge server for 6G network. *IEEE Internet of Things Journal*, *8*(22), 16370-16378.

Li, Z., Wang, X., & Zhang, T. (2020). *5G+: How 5G change the society*. Springer Nature.

Lin, L., Liao, X., Jin, H., & Li, P. (2019). Computation offloading toward edge computing. *Proceedings of the IEEE*, *107*(8), 1584-1607.

Lin, X., & Lee, N. (2021). 5G and Beyond. *Cham, Switzerland: Springer Nature Switzerland AG*.

Lin, Y.-B., Huang, T.-J., & Tsai, S.-C. (2019). Enhancing 5G/IoT transport security through content permutation. *IEEE Access*, *7*, 94293-94299.

Liu, D., Liang, H., Zeng, X., Zhang, Q., Zhang, Z., & Li, M. (2022). Edge computing application, architecture, and challenges in ubiquitous power internet of things. *Frontiers in Energy Research*, *10*, 850252.

Liu, G., Huang, Y., Chen, Z., Liu, L., Wang, Q., & Li, N. (2020). 5G deployment: Standalone vs. non-standalone from the operator perspective. *IEEE Communications Magazine*, *58*(11), 83-89.

Liu, J., Shu, L., Lu, X., & Liu, Y. (2023). Survey of Intelligent Agricultural IoT Based on 5G. *Electronics*, *12*(10), 2336.

Liu, Y., Peng, M., Shou, G., Chen, Y., & Chen, S. (2020). Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things. *IEEE Internet of Things Journal*, *7*(8), 6722-6747.

Ma, Z., Xiao, M., Xiao, Y., Pang, Z., Poor, H. V., & Vucetic, B. (2019). High-reliability and low-latency wireless communication for internet of things: Challenges, fundamentals, and enabling technologies. *IEEE Internet of Things Journal*, *6*(5), 7946-7970.

Makondo, N., Kobo, H. I., & Mathonsi, T. E. (2023). The latest developments in Software Defined Networking: Adoption rate and challenges. 2023 IEEE AFRICON,

Makondo, N., Kobo, H. I., Mathonsi, T. E., & Mamushiane, L. (2023). A review on edge computing in 5G-enabled IoT for agricultural applications: Opportunities and challenges. 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET),

Mamushiane, L. (2019a). Towards the development of an optimal SDN controller placement framework to expedite SDN deployment in emerging markets.

Mamushiane, L. (2019b). *Towards the development of an optimal SDN controller placement framework to expedite SDN deployment in emerging markets* Faculty of Engineering and the Built Environment].

Mamushiane, L., & Dlamini, S. (2017). Leveraging SDN/NFV as key stepping stones to the 5G era in emerging markets. 2017 Global Wireless Summit (GWS),

Mamushiane, L., Lysko, A., Kobo, H., & Mwangama, J. (2023). Deploying a stable 5G SA testbed using srsRAN and Open5GS: UE integration and troubleshooting towards network slicing. 2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD),

Mohamed, E. S., Belal, A., Abd-Elmabod, S. K., El-Shirbeny, M. A., Gad, A., & Zahran, M. B. (2021). Smart farming for improving agricultural management. *The Egyptian Journal of Remote Sensing and Space Science*, *24*(3), 971-981.

Mohamed, R., Zemouri, S., & Verikoukis, C. (2021). Performance evaluation and comparison between SA and NSA 5G networks in indoor environment. 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom),

Monamodi, S. (2020). SMART FARMING: IS CREATING INNOVATION HUBS A WAY OF INTRODUCING EMERGING FARMERS TO THIS SCIENCE? https://www.namc.co.za/wp-content/uploads/2021/03/Smart-Farming-Innovation-Hubs.pdf

Mrabet, H., Belguith, S., Alhomoud, A., & Jemai, A. (2020). A survey of IoT security based on a layered architecture of sensing and data analysis. *Sensors*, *20*(13), 3625.

Navarro, E., Costa, N., & Pereira, A. (2020). A systematic review of IoT solutions for smart farming. *Sensors*, *20*(15), 4231.

Nguyen, V.-G., Brunstrom, A., Grinnemo, K.-J., & Taheri, J. (2017). SDN/NFV-based mobile packet core network architectures: A survey. *IEEE Communications Surveys & Tutorials*, *19*(3), 1567-1602.

Nsiah, K. A., Amjad, Z., Sikora, A., Hilt, B., & Lauffenburger, J.-P. (2019). Latency reduction techniques for NB-IoT networks. 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS),

Ojha, T., Misra, S., & Raghuwanshi, N. S. (2015). Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Computers and Electronics in Agriculture*, *118*, 66-84.

Open5GS. *Open5gs documentation*. Retrieved 02/04/2024 from https://open5gs.org/open5gs/docs/guide/01-quickstart/

opensignal. Retrieved 03/04/2024 from https://www.opensignal.com/apps

Paolini, M., & Fili, S. (2018). Latency matters.

Parvez, I. (2018). Spectrum Sharing, Latency, and Security in 5G Networks with Application to IoT and Smart Grid.

Parvez, I., Rahmati, A., Guvenc, I., Sarwat, A. I., & Dai, H. (2018). A survey on low latency towards 5G: RAN, core network and caching solutions. *IEEE Communications Surveys & Tutorials*, *20*(4), 3098-3130.

Perakis, K., Lampathaki, F., Nikas, K., Georgiou, Y., Marko, O., & Maselyne, J. (2020). CYBELE–Fostering precision agriculture & livestock farming through secure access to large-scale HPC enabled virtual industrial experimentation environments fostering scalable big data analytics. *Computer Networks*, *168*, 107035.

Pereira, S., & Karia, D. (2018). AI Use Cases in Operational Support System and Business Support System. 2018 3rd International Conference on Communication and Electronics Systems (ICCES),

Peters, S., & Khan, M. A. (2019). Anticipatory session management and user plane function placement for ai-driven beyond 5g networks. *Procedia Computer Science*, *160*, 214-223.

Prasad, A., & Chawda, P. (2018). Power management factors and techniques for IoT design devices. 2018 19th International Symposium on Quality Electronic Design (ISQED),

Qin, J., Li, Z., Wang, R., Li, L., Yu, Z., He, X., & Liu, Y. (2021). Industrial Internet of Learning (IIoL): IIoT based pervasive knowledge network for LPWAN—concept, framework and case studies. *CCF Transactions on Pervasive Computing and Interaction*, *3*(1), 25-39.

Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., & Wu, D. O. (2020). Edge computing in industrial internet of things: Architecture, advances and challenges. *IEEE Communications Surveys & Tutorials*, *22*(4), 2462-2488.

Quy, V. K., Hau, N. V., Anh, D. V., Quy, N. M., Ban, N. T., Lanza, S., Randazzo, G., & Muzirafuti, A. (2022). IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges. *Applied Sciences*, *12*(7), 3396. https://www.mdpi.com/2076-3417/12/7/3396

Rafea, W. (2021). The effect of edge computing on Iot (5G Perspective).

Raghunandan, K. (2022a). Backhaul Network. In *Introduction to Wireless Communications and Networks: A Practical Perspective* (pp. 59-73). Springer.

Raghunandan, K. (2022b). Cellular Systems. In *Introduction to Wireless Communications and Networks: A Practical Perspective* (pp. 177-221). Springer.

Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, *19*(2), 855-873.

Ritola, V., Lamminmäki, J., Sainio, P., & Isoaho, J. (2022). 5G standalone network's reliability, one-way latency and packet loss rate analysis for URLLC implementation.

Romanov, O., Nesterenko, M., Marinov, A., Skolets, S., & Burlaka, H. (2022). SDN network modeling using the GUI MiniEdit. 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET),

Sadeeq, M. M., Abdulkareem, N. M., Zeebaree, S. R., Ahmed, D. M., Sami, A. S., & Zebari, R. R. (2021). IoT and Cloud computing issues, challenges and opportunities: A review. *Qubahan Academic Journal*, *1*(2), 1-7.

Saiz-Rubio, V., & Rovira-Más, F. (2020). From smart farming towards agriculture 5.0: A review on crop data management. *Agronomy*, *10*(2), 207.

SANREN. *South African NREN Backbone*. Retrieved 13/05/2024 from https://www.sanren.ac.za/backbone/

Saqib, M., Almohamad, T. A., & Mehmood, R. M. (2020). A low-cost information monitoring system for smart farming applications. *Sensors*, *20*(8), 2367.

Shah, S. D. A., Gregory, M. A., & Li, S. (2021). Cloud-native network slicing using software defined networking based multi-access edge computing: A survey. *IEEE Access*, *9*, 10903-10924.

Sheham, M. N. H., Hassan, M. T., Imran, A., Ahmed, T. F., Parvez, M. S., & Ahmed, A. (2022). Design of an IoT-based Smart Farming System: IoT-based Smart Farming System. Proceedings of the 2nd International Conference on Computing Advancements,

Shetty, R. S. (2021). 5G Mobile Core Network. *Apress, Bangalore*.

Singh, R. K., Bisht, D., & Prasad, R. (2017). Development of 5G mobile network technology and its architecture. *International Journal of Recent Trends in Engineering & Research (IJRTER)*, *3*(10), 196-201.

Sinha, B. B., & Dhanalakshmi, R. (2022). Recent advancements and challenges of Internet of Things in smart agriculture: A survey. *Future Generation Computer Systems*, *126*, 169-184.

Slalmi, A., Saadane, R., Chehri, A., & Kharraz, H. (2021). How will 5G transform industrial IoT: Latency and reliability analysis. Human Centred Intelligent Systems: Proceedings of KES-HCIS 2020 Conference,

Soepeno, R. A. A. P. (2023). Wireshark: An Effective Tool for Network Analysis.

srsRAN. *srsRAN 4G 23.11 Documentation*. Retrieved December from https://docs.srsran.com/en/latest/

srsRAN. *srsRAN Project*. Retrieved 02/04/2024 from https://www.srslte.com/5g

Sunyaev, A., & Sunyaev, A. (2020). Cloud computing. *Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies*, 195-236.

Tang, Y., Dananjayan, S., Hou, C., Guo, Q., Luo, S., & He, Y. (2021). A survey on the 5G network and its impact on agriculture: Challenges and opportunities. *Computers and Electronics in Agriculture*, *180*, 105895.

Team, M. (2017). Mininet: An Instant Virtual Network on your Laptop (or other PC)-Mininet.[online] Mininet. org. In.

Triantafyllou, A., Tsouros, D. C., Sarigiannidis, P., & Bibi, S. (2019). An architecture model for smart farming. 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS),

Uchenna, F., Aibinu, A. M., Olayiwola, R., Abba, E., Adedigba, A. P., & Folorunso, T. A. (2019). Step by Step Method of Programming Multiple Profile in a SIM Card.

Van Huynh, D., Nguyen, V.-D., Khosravirad, S. R., Sharma, V., Dobre, O. A., Shin, H., & Duong, T. Q. (2022). URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach. *IEEE Transactions on Communications*, *70*(11), 7669-7682.

Veissier, I., Kling-Eveillard, F., Mialon, M.-M., Silberberg, M., Roches, A. d. B., Terlouw, C., Ledoux, D., Meunier, B., & Hostiou, N. (2019). Precision Livestock Farming and animal welfare: is the numerical revolution of agriculture able to take into account animals' and farmers' needs? *INRA Productions Animales*, *32*(2), 281-290.

Wang, M., Simon, G., Neto, L. A., Amigo, I., Nuaymi, L., & Chanclou, P. (2022). A MEC and UPF compatible OLT for time-critical mobile services. 2022 International Conference on Optical Network Design and Modeling (ONDM),

Wang, S.-Y., Chang, J.-E., Fan, H., & Sun, Y.-H. (2020). Performance comparisons of NB-IoT, LTE Cat-M1, Sigfox, and LoRa moving at high speeds in the air. 2020 IEEE Symposium on Computers and Communications (ISCC),

Wang, T.-H., Hu, M.-C., Yen, L.-H., & Tseng, C.-C. (2023). Heterogeneous UPF Integration Framework and 5G User Plane Acceleration. 2023 24st Asia-Pacific Network Operations and Management Symposium (APNOMS),

Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2014). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, *17*(1), 27-51.

Yan, L., Cao, S., Gong, Y., Han, H., Wei, J., Zhao, Y., & Yang, S. (2019). SatEC: A 5G satellite edge computing framework based on microservice architecture. *Sensors*, *19*(4), 831.

Yazıcı, V., Kozat, U. C., & Sunay, M. O. (2014). A new control plane for 5G network architecture with a case study on unified handoff, mobility, and routing management. *IEEE Communications Magazine*, *52*(11), 76-85.

Yousaf, F. Z., Bredel, M., Schaller, S., & Schneider, F. (2017). NFV and SDN—Key technology enablers for 5G networks. *IEEE Journal on Selected Areas in Communications*, *35*(11), 2468-2478.

Yusuf, B. (2023). Minimizing the clock drift in partially synchronized heterogeneous TSN networks. In.

Zacarias, I., Carpio, F., Drummond, A. C., & Jukan, A. (2023). Optimum Network Slicing for Ultra-reliable Low Latency Communication (URLLC) Services in Campus Networks. 2023 19th International Conference on the Design of Reliable Communication Networks (DRCN),

Zeyu, H., Geming, X., Zhaohang, W., & Sen, Y. (2020). Survey on edge computing security. 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE),

Zhang, S. (2019). An overview of network slicing for 5G. *IEEE Wireless Communications*, *26*(3), 111-117.

Zhu, R., Liu, L., Song, H., & Ma, M. (2020). Multi-access edge computing enabled internet of things: advances and novel applications. In (Vol. 32, pp. 15313-15316): Springer.

Zieliński, B. (2023). Assessment of iPerf as a Tool for LAN Throughput Prediction. *International Journal of Electronics and Telecommunications*, 523-528-523-528.

Appendix A: SANReN Network Topology Python Source Code

```
#!/usr/bin/python
# This code does the following:
# Reads a topology gml file using igraph and networkx;
# Computes link latencies;
# Emulates the topology;
# Partitions the network into two domains and instantiate controllers;
# Finds optimal locations to place controllers using a ping procedure
import os
from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.node import OVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
import igraph as ig
import networkx as nx
import matplotlib.pyplot as plt
import haversine
import numpy as np
import haversine as hv
from mininet.nodelib import NAT
from mininet.nodelib import NAT
from mininet.topolib import TreeNet

def myNetwork():
```

```
net = Mininet(topo=None, build=False, ipBase='10.0.0.0/24')
info('*** Adding controller\n')
c0 = net.addController(name='c0', controller=RemoteController, ip='146.64.28.254', port=6633)

g = ig.Graph.Read_GML("Sanren.gml")

info('*** Adding switches and hosts\n')
nodes = g.vcount()
print(nodes)

for i in range(0, nodes):
    s = "s" + str(i)
    h = "h" + str(i)
    IP = "10.0.0." + str(i + 1)

    s = net.addSwitch(s, cls=OVSSwitch, protocols='OpenFlow14')
    h = net.addHost(h, cls=Host, ip=IP, defaultRoute=None)
    net.addLink(h, s, cls=TCLink, bw=1000, delay= '0.5ms') #delay='0ms'
   # net.addLink('s1', 's6', cls=TCLink, delay='200ms')  # Increase delay to 200ms


    print(s)

info('*** Connecting switches\n')
edges = g.get_edgelist()
print(edges)
s = [None] * 0

for i in edges:
    s = [None] * 0
    for j in i:
        r = "s" + str(j)
        s.insert(0, r)
    print(s)
    source = s[1]
    target = s[0]
    source_target = "*" + "*" + source + target
    print(source)
    print(target)
    print(source_target)
    net.addLink(source, target, cls=TCLink)

#############configure link delays########
info('*** Configuring link delays\n')
latlon = []
lats = list(g.vs['Latitude'])
lons = list(g.vs['Longitude'])
lat = np.array([lats])
```

```
lon = np.array([lons])

latlon = np.concatenate((lat.T, lon.T), axis=1)
lat_lon1 = latlon.tolist()
print(lat_lon1)
m, n = 2, 2;
lonlat2 = [[0 for x in range(m)] for y in range(n)]
print(lonlat2)
edges = g.get_edgelist()
print(edges)
s = [None] * 0
for i in edges:
    print(i)
    for j in range(0, 2):
        print(i[j])
        k = i[j]
        lonlat2[j][:] = lat_lon1[k]
    print(lonlat2)
    link_delays = hv.distance(lonlat2)
    print(link_delays)
    for z in i:
        q = "s" + str(z)
        s.insert(0, q)
    p = s[1] + s[0]
    link_delays2 = str(link_delays)
    p = {'delay': link_delays2}
    print(p)

info('*** Starting network\n')
net.build()
info('*** Starting controllers\n')
for controller in net.controllers:
    controller.start()

info('*** Starting switches\n')
for i in range(0, nodes):
    s = "s" + str(i)
    print(s)
    net.get(s).start([c0])

info('Optimizing controller placement through ping\n')
hosts = net.hosts

# Remove hosts h0 and h6
for host in ['h1', 'h6']:
#for host in ['h0', 'h1', 'h2', 'h3', 'h4', 'h5', 'h6']:
```

```
        net.delNode(net.get(host))

    # Attach s0-eth1 and s6-eth1 to switches s0 and s6
    s1 = net.get('s1')
    s1.attach('s1-eth1')

    #UPF port
    s1 = net.get('s1')
    s1.attach('s1-eth2')


    s6 = net.get('s6')
    s6.attach('s6-eth1')

    # Bridge s1-eth1 to enp4s0
    os.system('ovs-vsctl add-port s1 enp4s0')
    os.system('ip link set dev enp4s0 up')

    #UPF PORT
    os.system('ovs-vsctl add-port s1 enx0c37965d00b8')
    os.system('ip link set dev enx0c37965d00b8 up')

    # Bridge s6-eth1 to enp3s0
    os.system('ovs-vsctl add-port s6 enp3s0')
    os.system('ip link set dev enp3s0 up')

    outfiles, errfiles = {}, {}
    hosts_count = len(hosts)
    for k in range(0, hosts_count):
        for h in hosts:
            server = hosts[k - 1]
            outfiles[h] = '/tmp/%s.out' % h.name
            errfiles[h] = '/tmp/%s.err' % h.name
            h.cmd('echo >', outfiles[h])
            h.cmd('echo >', errfiles[h])

    for hpair in zip(hosts, hosts[1:]):
        net.ping(hpair, timeout='1')
    net.ping((hosts[-1], hosts[0]), timeout='1')

    info('*** Post configure switches and hosts\n')
    CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    myNetwork()
```

Appendix B: Open5GS Configurations on the Cloud Core

```
db_uri: mongodb://localhost/open5gs

logger:

sbi:
  server:
    no_tls: true
    cacert: @build_configs_dir@/open5gs/tls/ca.crt
    key: @build_configs_dir@/open5gs/tls/testserver.key
    cert: @build_configs_dir@/open5gs/tls/testserver.crt
  client:
    no_tls: true
    cacert: @build_configs_dir@/open5gs/tls/ca.crt
    key: @build_configs_dir@/open5gs/tls/testclient.key
    cert: @build_configs_dir@/open5gs/tls/testclient.crt

parameter:
#   no_nrf: true
#   no_scp: true
#   no_amf: true
#   no_smf: true
  no_upf: true                    #Commented out as per CUPS
#   no_ausf: true
#   no_udm: true
#   no_pcf: true
#   no_nssf: true
#   no_bsf: true
#   no_udr: true
#   no_mme: true
#   no_sgwc: true
#   no_sgwu: true
#   no_pcrf: true
#   no_hss: true
#   use_mongodb_change_stream: true

mme:
  freeDiameter:
    identity: mme.localdomain
    realm: localdomain
    listen_on: 127.0.0.2
    no_fwd: true
    load_extension:
      - module: @build_subprojects_freeDiameter_extensions_dir@/dbg_msg_dumps.fdx
        conf: 0x8888
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_rfc5777.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_mip6i.fdx
```

```
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_nasreq.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_nas_mipv6.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_dcca.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_dcca_3gpp/dict_dcca_3gpp.fdx
    connect:
      - identity: hss.localdomain
        addr: 127.0.0.8

  s1ap:
    - addr: 127.0.0.2
  gtpc:
    - addr: 127.0.0.2
  metrics:
    addr: 127.0.0.2
    port: 9090
  gummei:
    plmn_id:
      mcc: 999
      mnc: 70
    mme_gid: 2
    mme_code: 1
  tai:
    plmn_id:
      mcc: 999
      mnc: 70
    tac: 1
  security:
      integrity_order : [ EIA2, EIA1, EIA0 ]
      ciphering_order : [ EEA0, EEA1, EEA2 ]

  network_name:
      full: Open5GS

sgwc:
  gtpc:
    - addr: 127.0.0.3
  pfcp:
    - addr: 127.0.0.3

smf:
  sbi:
    - addr: 127.0.0.4
      port: 7777
  pfcp:
   - addr: 10.0.0.40                          #IP ADDRESS OF EDGE UPF

  gtpc:
```

```
    - addr: 127.0.0.4
    - addr: ::1
  gtpu:
    - addr: 127.0.0.4
    - addr: ::1
  metrics:
    addr: 127.0.0.4
    port: 9090
  subnet:
    - addr: 10.45.0.1/16
    - addr: 2001:db8:cafe::1/48
  dns:
    - 8.8.8.8
    - 8.8.4.4
    - 2001:4860:4860::8888
    - 2001:4860:4860::8844
  mtu: 1400
  freeDiameter:
    identity: smf.localdomain
    realm: localdomain
    listen_on: 127.0.0.4
    no_fwd: true
    load_extension:
      - module: @build_subprojects_freeDiameter_extensions_dir@/dbg_msg_dumps.fdx
        conf: 0x8888
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_rfc5777.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_mip6i.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_nasreq.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_nas_mipv6.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_dcca.fdx
      - module: @build_subprojects_freeDiameter_extensions_dir@/dict_dcca_3gpp/dict_dcca_3gpp.fdx
    connect:
      - identity: pcrf.localdomain
        addr: 127.0.0.9

amf:
  sbi:
    - addr: 127.0.0.5
      port: 7777
  ngap:
    - addr: 127.0.0.5
  metrics:
    addr: 127.0.0.5
    port: 9090
  guami:
    - plmn_id:
        mcc: 999
```

```
        mnc: 70
      amf_id:
        region: 2
        set: 1
  tai:
    - plmn_id:
        mcc: 999
        mnc: 70
      tac: 1
  plmn_support:
    - plmn_id:
        mcc: 999
        mnc: 70
      s_nssai:
        - sst: 1
  security:
    integrity_order : [ NIA2, NIA1, NIA0 ]
    ciphering_order : [ NEA0, NEA1, NEA2 ]
  network_name:
    full: Open5GS
  amf_name: open5gs-amf0

sgwu:
  pfcp:
    - addr: 127.0.0.6
  gtpu:
    - addr: 127.0.0.6

upf:
  pfcp:
    - addr: 10.0.0.40                   #IP ADDRESS OF EDGE UPF
  gtpu:
    - addr: 10.0.0.40                    #IP ADDRESS OF EDGE UPF
  subnet:
    - addr: 10.45.0.1/16
    - addr: 2001:db8:cafe::1/48
  metrics:
    - addr: 127.0.0.4
      port: 9090
```

Appendix C: UPF node configurations

```
logger:
  file: /var/log/open5gs/upf.log
#  level: info   # fatal|error|warn|info(default)|debug|trace

global:
  max:
    ue: 1024  # The number of UE can be increased depending on memory size.
#    peer: 64

upf:
  pfcp:
    server:
      - address: 10.0.0.40
    client:
      smf:    #  UPF PFCP Client try to associate SMF PFCP Server
        - address: 10.0.0.30                    #IP ADDRESS OF CLOUD CORE
  gtpu:
    server:
      - address: 10.0.0.40
  session:
    - subnet: 10.45.0.1/16
    - subnet: 2001:db8:cafe::1/48
  metrics:
    server:
      - address: 10.0.0.40
        port: 9090
#
```

Appendix D: srsRAN Configuration

```
amf:
  addr: 10.0.0.30                                 # The address or hostname of the AMF.
  bind_addr: 10.0.0.20                            # A local IP that the gNB binds to for traffic from the
AMF.

ru_sdr:
  device_driver: uhd                              # The RF driver name.
  device_args: type=b200,num_recv_frames=64,num_send_frames=64    # Optionally pass arguments to
the selected RF driver.
  clock: gpsdo                  # Specify the clock source used by the RF.
#  sync:  external                # Specify the sync source used by the RF.
  srate: 23.04                                    # RF sample rate might need to be adjusted according to
selected bandwidth.
  otw_format: sc12
  tx_gain: 85                                     # Transmit gain of the RF might need to adjusted to the given
situation.
  rx_gain: 35                                     # Receive gain of the RF might need to adjusted to the given
situation.

cell_cfg:
  dl_arfcn: 632628                                # ARFCN of the downlink carrier (center frequency).
  band: 78                                        # The NR band.
  channel_bandwidth_MHz: 20                       # Bandwith in MHz. Number of PRBs will be
automatically derived.
  common_scs: 30                                  # Subcarrier spacing in kHz used for data.
  plmn: "00101"                                   # PLMN broadcasted by the gNB.
  tac: 1                                          # Tracking area code (needs to match the core configuration).
  pci: 1                                          # Physical cell ID.
  pucch:
    sr_period_ms: 10
log:
  filename: /tmp/gnb.log1                         # Path of the log file.
  all_level:
 # phy_level:                                     # Logging level applied to all layers.

pcap:
  mac_enable: True                                # Set to true to enable MAC-layer PCAPs.
  mac_filename: /tmp/gnb_mac.pcap                 # Path where the MAC PCAP is stored.
  ngap_enable: True                               # Set to true to enable NGAP PCAPs.
  ngap_filename: /tmp/gnb_ngap.pcap               # Path where the NGAP PCAP is stored.
```