Håvard Persson

Evaluation of 5G Non-Standalone Commercial Networks for Remote Control Operations

July 2021

NTNU
Norwegian University of
Science and Technology

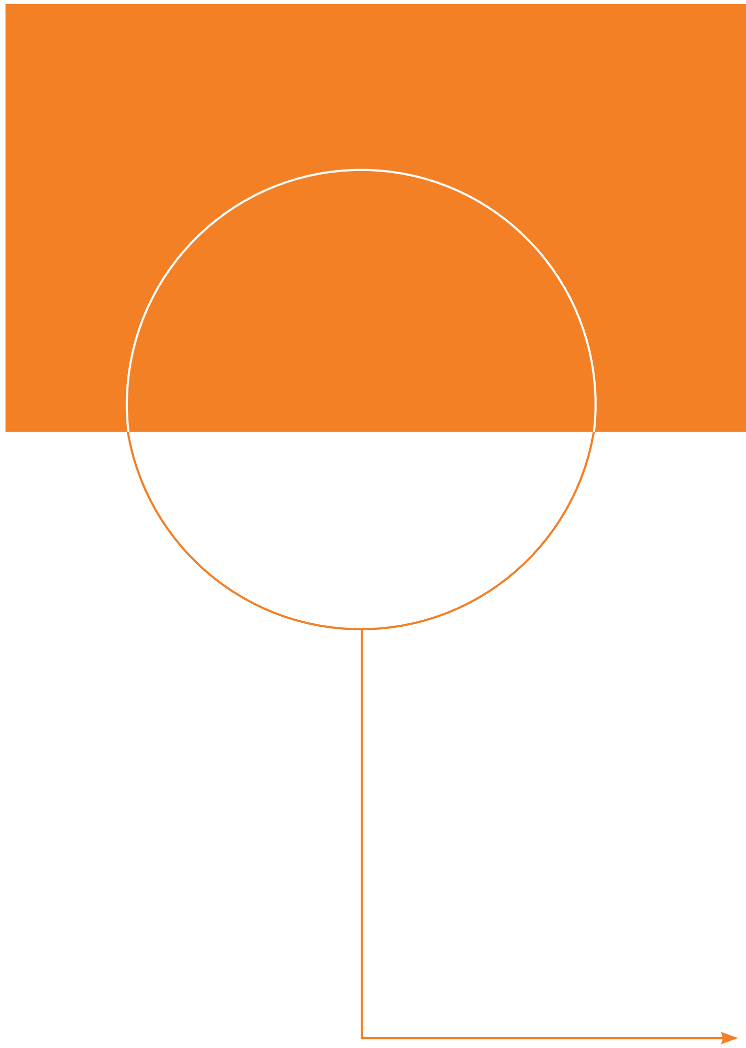
Master's thesis

2021

Håvard Persson

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering



NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Evaluation of 5G Non-Standalone Commercial Networks for Remote Control Operations

Håvard Persson

Subsea Technology	
Submission date:	July 2021
Supervisor:	Professor Amund Skavhaug NTNU
Co-supervisor:	Professor Stig Petersen NTNU Professor Lars Dittmann DTU

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

Preface

This thesis concludes my work at the Norwegian University of Science and Technology and my Master's Degree in Subsea Technology. The work was conducted as a part of the 5G Solutions project for the Department of Mechanical and Industrial Engineering. The project work was carried out in the spring semester 2021, which considers both work in Norway and in Denmark. In addition, this thesis also represents work from when the specialization started in the spring of 2020.

The Covid-19 situation had a significant impact on the project. The 5G Standalone network, which were intended for this project, was not installed at NTNU before the end of June 2021. Travel restrictions made it not possible for me to configure the NTNU setup, but collaboration with my colleagues at NTNU made this possible. Nevertheless, the project work was conducted in spite of these difficulties, and I am happy the way it turned out in the end.

I want to thank my supervisors Amund Skavhaug and Stig Peterson, for the guidance, motivation, and good conversations. I also want to thank Lars Dittmann and my colleagues at DTU and NTNU for the collaboration, and for helping me complete this project. I also want to thank my family and friends for their valuable support and discussions.

The thesis is intended for engineers without a telecommunication background, as well as those with one. It provides the basic understanding of telecommunication to support the people moving further with the thesis work. I hope that the work can lead up to discussions and benefit those who work with telecommunication and remote operations for industry 4.0.

Håvard Persson
July 7, 2021

Abstract

This thesis investigates the possibilities for remote control operations between two universities, DTU in Denmark and NTNU in Norway, using the 5G commercial networks available at the universities. The telecommunication is configured to control an AGV with a connected Raspberry Pi at the Manulab facilities at NTNU. The thesis also addresses a theoretical overview of the telecommunication system and of a systematized testing tool.

It was decided to use three Raspberry Pis (RPis) and one iPhone 12 as endpoints for the telecommunication setup. One Raspberry Pi connected to the AGV (RPR), one at NTNU acting as a hub for communication (RPT), and one at DTU (RPD). The iPhone 12 was used as an option for the RPD. The three Raspberry Pis enabled 5G New Radio technology by installing 5G HATs.

The two RPis at NTNU were connected to a 3500 MHz 5G Non-Standalone commercial network. Similarly, the RPD and the iPhone were connected to a 700 MHz 5G Non-Standalone commercial network. This thesis evaluated the system's performance with kernel evaluations and six network test scenarios. The RPT was the central unit of the telecommunication setup, meaning that all other endpoints wanted to send data to or receive from the RPT.

A networking analysis tool was used for measurements. This tool was used to measure KPIs between the endpoints of the telecommunication network. The Hawkeye tests simulated scenarios relevant for robotic control, and measured latency, reliability, and throughput during these test. Results showed that the kernel's performance of the RPis did not cause a significant latency throughout the system. Furthermore, the telecommunication system was not suited for time-sensitive robotic applications due to higher latency than required for the Industry 4.0 standards. The test scenarios' optimum solution was evaluated to be the scenario where the communication link between NTNU and DTU was wired, using the LAN of each university. Thus, suggestions of how to improve the system with 5G and further testing are presented.

5G is under massive development, meaning that this telecommunication system will be further developed when the 5G technology gets upgraded at the locations of the Raspberry Pis. The telecommunication test setup presents a setup that can be configured into a private 5G Standalone network, which provides the operators a better overview of the robotic systems' bandwidth usage.

This thesis developed a telecommunication testing setup that is relevant for further testing and evaluation. Measurements of a 5G private Standalone network could be compared to the findings in this thesis. The test setup, procedures and methodology can be reused in this private network, to isolate the qualities of the 5G standalone network.

Sammendrag

Denne masteroppgaven undersøker mulighetene for fjernstyring av en robot mellom to universiteter, DTU i Danmark og NTNU i Norge, ved hjelp av det kommersielle 5G nettverket som er tilgjengelig på universitetene. Telekommunikasjonen er konfigurert til å kontrollere en AGV med en tilkoblet Raspberry Pi ved Manulab på NTNU. Oppgaven håndterer også en teoretisk oversikt over telekommunikasjonssystemet og og over et systematisert testverktøy.

Det ble besluttet å bruke tre Raspberry Pis (RPis) og en iPhone 12 som Endpoints for telekommunikasjonen. Én Raspberry Pi koblet til AGVen (RPR), en på NTNU som fungerte som en hub (RPT), der robotsystemer skal kunne kobles til, og en på DTU (RPD). En iPhone 12 ble brukt som et alternativ for RPD. De tre Raspberry Piene implementerte 5G New Radio-teknologien ved å installere en 5G HAT.

De to RPiene på NTNU var koblet til et 3500 MHz 5G Non-Standalone kommersielt nettverk. På samme måte koblet RPDen og iPhonen seg til et 700 MHz 5G Non-Standalone kommersielt nettverk. Denne oppgaven evaluerte systemets ytelse med kjerneevalueringer og seks nettverks testscenarier. RPT var den sentrale enheten i telekommunikasjonsoppsettet, noe som betydde at alle andre endepunkter ønsket å sende data til eller motta fra RPT.

For å gjøre målinger ble et verktøy for nettverksanalyse brukt, kalt Hawkeye. Dette verktøyet ble brukt til å måle KPIer mellom endepunktene i telekommunikasjonsnettverket. Hawkeye-testene simulerte scenarioer som var relevant for robotkontroll, og målte latency, pålitelighet og throughput under testen. Resultatene viste at kjernens ytelse til RPiene ikke forårsaket betydelig latency i hele systemet. Telekommunikasjonssystemet ble vurdert til å ikke være egnet for tidssensitive robotapplikasjoner, på grunn av høyere latency-verdier enn det som kreves av industri 4.0-standardene. Den optimale løsningen av testscenariene ble vurdert til å være scenariet der kommunikasjonsforbindelsen mellom NTNU og DTU ble kablet. Dermed presenteres forslag til hvordan du kan forbedre systemet, og videre testing.

5G er under massiv utvikling, noe som betyr at dette telekommunikasjonssystemet vil bli videreutviklet når 5G-teknologien blir oppgradert der Raspberry Piene er. Systemoppsettet presenterer også et setup som kan konfigureres til et privat 5G Standalone nettverk, som gir operatørene bedre oversikt over bruken av båndbredde til robotsystemene.

Denne masteroppgaven utviklet et testoppsett for telekommunikasjon som er relevant for videre testing og evaluering. Et 5G privat Standalone nettverk kan sammenlignes med funnene i denne masteroppgaven. Testoppsettet, prosedyrene og metodikken kan brukes på nytt i dette private nettverket, for å isolere kvalitetene til Standalone 5G-nettverket.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	3rd Generation Partnership Project
AR	Augmented Reality
bps	bits per second
DDS	Data Distribution System
DL	Download
DOF	Degree of Freedom
DTU	Technical University of Denmark
FDI	Fast Data Interface
HAT	Hardware Attached on Top
HMI	Human-Machine-Interface
IEEE	Institute of Electrical and Electronics Engineers
KCP	KUKA Control PC
KMP	KUKA Mobile Platform
KMR iiwa	KUKA Mobile Robot Intelligent Industrial Work Assistant
KPI	Key Performance Indicator
KRC	KUKA Control PC
LAN	Local Area Network
LBR	Lightweight Robot
MAC	Media Access Control
mMTC	massive Machine-Type Communication
N-KPI	Hawkeye Network KPI Test
NIC	Network Interface Card
NTNU	The Norwegian University of Science and Technology
NTP	Network Time Protocol
OSI	Open System Interconnection
PLMN	Public Land Mobile Network
QoS	Quality of Service

ROS	Robot Operating System
RPD	Raspberry Pi at DTU Office
RPi	Raspberry Pi
RPR	Raspberry Pi Connected to Robot
RPT	Raspberry Pi at Trondheim Office
RSSI	Received Signal Strength Indicator
RTOS	Real-Time Operating System
RTPS	Real-Time Publish Subscribe
RTT	Round-Trip Time
S4B	Hawkeye Skype4B Test
SBC	Single Board Computer
SSH	Secure Shell
ST	Hawkeye Speedtest
TCP/IP	Transmission Control Protocol/Internet Protocol
TCP	Transmission Control Protocol
UC	Use Case
UDP	User Datagram Protocol
UL	Upload
URLLC	Ultra-Reliable Low Latency Communication
VR	Virtual Reality
WAP	Wifi Access Point

Preliminaries

PART 1

1 Introduction

The technological world is changing persistently, and adaption to the present solutions is crucial for being a part of this change. The Industrial Internet of Things (IIoT) is a goal and a longing for industries worldwide. To be able to control the industry across the globe and to perform crucial tasks constantly without interruption is something that is valued by companies and is the motivation of many technological developers. The profit comes from efficiency, and technology is a supporting factor of the efficiency for the industries.

1.1 Background and Motivation

5G presents the fifth generation of cellular networks, standardized by the 3rd Generation Partnership Project (3GPP). The 3GPP updates their standardization of 5G constantly and accelerates the development of 5G [1]. 5G promises lower latency, reliable connection, higher throughput, and more connected devices than the previous generations cellular networks. 5G brings new use cases to the IIoT, and demanding time-critical processes that usually do not rely on wireless technology, can find potential in adaptation of 5G [38]. For example, the robotic industries can benefit from the low latency for remote control. The distance from the operator to the robot is equivalent to the minimum distance packets have to travel, and which induces latency on the system. The scenario of remote control also relates to the current Covid-19 pandemic, where e-health and remote control from our homes can significantly impact both industries and people personally.

This project work evaluates a transmission network from the Technical University of Denmark (DTU) to the Norwegian University of Science and Technology (NTNU). The focus is around a remote control scenario where three Raspberry Pi's (RPI's) (two at NTNU and one at DTU) are used to control and observe a KMR iiwa robot. The three RPI's have a connection to a Non-Standalone 5G cellular band provided by Telenor in Norway and TDC in Denmark. This setup is monitored and evaluated in multiple test scenarios. This thesis is a continuation of the author's work conducted in the specialization project[32], and also a part of a 5G project, 5G Solutions for European Citizens. Some sections are included from or based upon sections provided in the specialization project.

1.2 Previous Work

Two former students at NTNU conducted work during the fall 2019 and spring 2020, and was about configuring and operating KMR iiwa in the Manulab at NTNU [7]. The KMR iiwas are the same AGVs used for the specialization project and this project. The goal was to integrate communication between a KMR iiwa with the middleware ROS. Conclusions made in these studies described high latency, and packet loss of up to 50% [7]. This study was conducted using the Wireless Access Point of the KMR iiwa as a direct link to the router. Figure 1.1 shows captured latency from the studies lead in 2019/2020. These graphs were obtained while heavy programs ran on the robotic system and showed an average latency of over 200ms. These latency results induced questions about the system's real-time capabilities, and latency in that range would not be accepted for time-sensitive tasks.

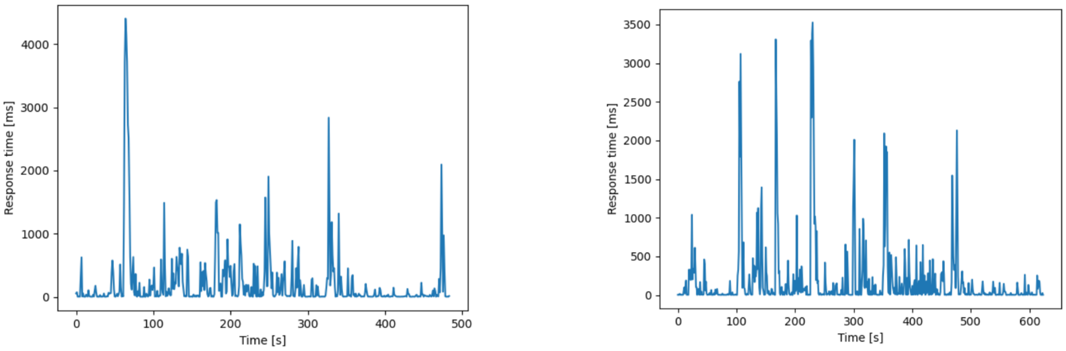


Figure 1.1: Latency experienced in the studies conducted in [7].

The previous work of the author was conducted in the fall of 2020. This project titled *Remote operations and testing of KMR iiwa using Raspberry Pi 4 with Wi-Fi* [32]. The project was about establishing a remote control setup for the KMR iiwa at the Manulab at NTNU. This setup included the KMR iiwa, a Raspberry Pi 4, an Wi-Fi access point, and a Work Station, see Figure 1.2. All test equipment were located at the Manulab, less than ten meters apart. The Raspberry Pi 4, referred to as RPi, was mounted on top and connected to the KMR iiwa via an Ethernet cable. The RPi controlled the robotic system via TCP/UDP socket communication between the Sunrise OS running on the robot and ROS2 on the RPi. The Wi-Fi connection between the Work Station and the Raspberry Pi was used to connect to the RPi via SSH, thus granting control over the KMR iiwa.

Four test scenarios were conducted to evaluate the latency, throughput, and reliability of the communication link between Work Station and KMR iiwa. The first scenario tested the communication links, Work Station to RPi and RPi to KMR iiwa, in idle states, which means no programs running in the background except the connection established between ROS2 and the Sunrise OS on the KMR iiwa. The second scenario was testing the indoor communication range of the Wi-Fi signal, also in idle states. The third scenario was conducted while the RPi was receiving sensor readings from the KMR iiwa. The fourth scenario was testing the whole system during heavy load on the robotic system.

The project concludes with a functional system and reliable connection between the KMR iiwa and RPi. As expected, there were significant benefits of having a cabled connection. The wireless communication between the RPi and the Work Station was experiencing latency spikes and inconsistency in the system. A concluding theory was that the communication between RPi and KMR iiwa was taking the priority of the CPU on the RPi, resulting in the measuring task needing to wait for the CPU.

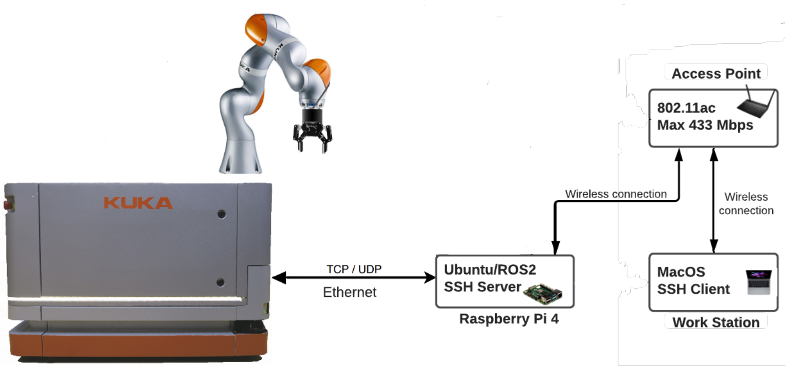


Figure 1.2: The system setup and communication channels of the specialization project.

1.2.1 Contribution to the 5G Solutions project

In June 2019, the EU started a project named 5G Solutions and aimed to prove and validate that the 5G capabilities provide prominent industry verticals with ubiquitous access to a wide range of forward-looking services with orders of magnitude of improvement over 4G [37]. The project focuses on targeting Vertical Industries applications, and these five are the most important ones; Industry 4.0, Smart Energy, Smart Cities, Smart Ports, and media. The project will conduct over 20 innovative use cases (UC) with 5G in Italy, Norway, Greece, Ireland, and Belgium, evaluating over 140 Key Performance Indicators (KPIs) for these industries. NTNU is contributing to this project and will, related to this, evaluate use cases at NTNU. UC 1.3 is about remotely controlling digital factories. The specialization project and this thesis are conducted based on UC 1.3. Initially, this thesis would provide the given KPI's that were originally stated in UC 1.3. Due to 5G deployment issues and time limitations, this work was based upon what initially was wanted by the UC. A problem description of UC 1.3 is giving in section 1.2.2.

1.2.2 UC 1.3: Remotely Controlling Digital Factories

[1]The simplest setup of this use case involves remote control applications running on tablets or smartphones, for example. However, given the trend of new AR devices, it is likely that new remote services may arise that facilitate the creation of virtual back office teams. Such remote teams may use the data coming from smart devices for preventive analytics and easy access to work instructions, whereby, e.g., they would be able to view the camera or iPad/Google Glass of a local worker. Additionally, the application of AR in the plant will facilitate:

- Augmented-reality support in production and assembly: Precisely positioned picture-in picture fade-ins, showing the operator the next step and helping to avoid misplacement and unnecessary scrap.
- Augmented-reality support in maintenance and repair: Repair machines without training due to augmented information and operational guidance.

Cross-functional communication, effective knowledge sharing and collaborative design platforms will be facilitated by solutions for communities of practice. In this use case family, there is a less stringent need for low-latency. Interaction times up to seconds are acceptable for remote servicing machines. However, high availability is key for allowing (emergency) maintenance actions to occur immediately. Bandwidth is important for video-controlled maintenance, with real-time augmented content mixed into the video signal. Moreover, latency is particularly important for real-time,

---

[1] This section is taken from the description of the UC 1.3 by the 5G Solutions project.

remote motion control of local robots. Edge computing within the network is required for fulfilling the low latency requirements. Security threats are introduced due to the opening up of the machines to allow remote reconfiguration. As such, the cyber-representation of a factory or supply chain needs to be protected, with mission-critical actions being shielded from non-authorised parties.

1.3 Problem description

The main objective of this thesis is to establish a telecommunication setup between DTU in Denmark and NTNU in Norway using 5G. This communication link is a complex combination of servers, routers, switches, base stations. A communication link should be established to utilize the 5G NR technology that is currently available and contribute to the local Internet Service Provider's (IPS) infrastructure in the 5G communication network. The framework of the setup should be configured with regard to the industrial applications of 5G.

Research should investigate communication protocols and explore the different hardware and software required to make a valid link between the endpoints. Furthermore, using small endpoint devices for a scalable system, which can be further expanded. The system should be applicable in an industrial setting, and results should be gathered that support the system's usability in such a setting.

The project handles monitoring the end-to-end system setup and the local transmission line at NTNU compared to the specialization project's results. The monitoring technology should be comprehensive and capture the needed KPI measurements of the transmission network to the extent available.

1.3.1 Project Scope

The scope of the thesis can be summarized as an establishment of a 5G telecommunication architecture between DTU and NTNU. This architecture is used for different remote control scenarios of the robotic system mentioned in section 1.2, as of which the test scenarios for this thesis were developed. An overview of the communication architecture is illustrated in Figure 1.3. The Raspberry Pi, located in Denmark, will be switched with an iPhone 12 for supplemental test scenarios. It is impossible to pin point where delays occur between DTU and NTNU due to multiple hops between unavailable routers and switches, without private access. Nevertheless, this thesis tries to handle the following research questions:

1. Establishment of a remote control telecommunication setup for the KMR iiwa, between DTU and NTNU using 5G.
2. Measure and evaluate KPIs for the system setup.

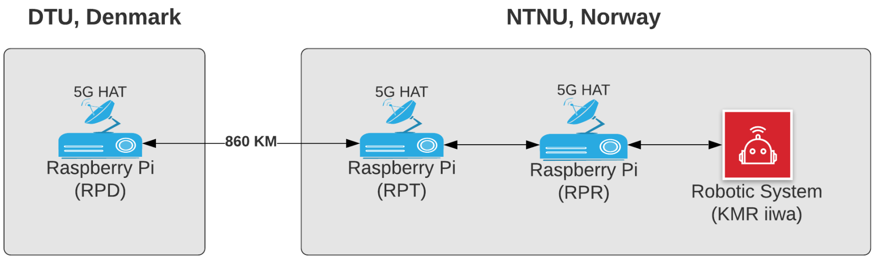


Figure 1.3: An illustration of the communication architecture investigated in this thesis.

1.4 Related Work

Related work concerning 5G technology for industrial robotic systems is available to an extent. Many papers discuss and evaluate different system setups. To contribute to this study in 5G robotic applications, researchers must develop unique systems and solutions. Three papers are discussed in this section, to provide an overview of how 5G communication is validated for industrial purposes. The importance of the data transmission networks discussed in these papers is their relation to the industries' scenarios and the use cases of such system setups. The following subsections discuss these papers briefly.

"Utilizing 5G in Industrial Robotic Applications"

This paper [5], suggests and describes 5G industrial use cases for automation, focusing on robotic systems and the benefits of 5G. Three robotic use cases are presented; Teleoperation/remote control, motion coordination/synchronization, and additive sensors.

Furthermore, the paper describes a scenario, which compares the 4G network to the 5G network. This scenario describes a wireless multi-axis force sensor that is installed on the end-effector of a manipulator. A controller controls the manipulator to exert a vertical force on a workpiece, see Figure 1.4. This scenario tries to map the difference in 4G and 5G communication regarding the communication stability needed for reliable force control in industrial applications.

Results acquired in this paper show that a robotic controller using 5G achieved higher stability when varying the applied force reference point over a 4G controller. In the scenarios, it is observed that 5G opens up possibilities for mission-critical applications that require very low latency and very high reliability for the communication links.

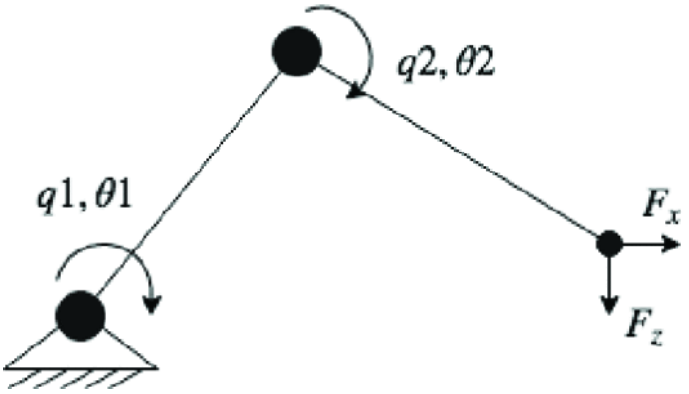


Figure 1.4: Structure of the multi-axes robotic system for exertion of force on a workpiece.

"5G for Robotics: Ultra-Low Latency Control of Distributed Robotic Systems"

This paper [42], considers the ultra-reliable low latency communication (URLLC) capabilities in 5G for a control system. The purpose is to detect the system's requirements and the possibilities to perform computational exhaustive remotely from the controller. Communication between the controller and the node is established via 5G.

A prototype is developed to test this outsourcing of demanding tasks to a controller in the cloud, which performs the calculations of a mobile robotic platform. These calculations include control calculations, path planning, and inverse kinematics. Nevertheless, control of the base and axis of the robotic arm are calculated by the robot's CPU and joint commands executed in the motors. The experiments evaluated time-critical tasks in ROS. They were analyzed by examining the transmitted messages for a robotic system with homogeneous data sources. The analysis showed promising results for time-critical tasks being remotely calculated.

This system analysis supports the scaling of remote 5G operations. A remote cloud (a server with great computational power) could support many robotic systems in calculations. This will lead to less demanding specifications for the robotic systems, given that even the time-critical tasks can be remotely calculated. A picture of the experimental setup is given in Figure 1.5.

Figure 1.5: Picture of the experimental setup given in article: "5G for Robotics: Ultra-Low Latency Control of Distributed Robotic Systems" [42].

"Teleoperation of an Industrial Robot using a Non-Standalone 5G Mobile Network"

This paper [45]proposes an architecture for a Non-standalone 5G Mobile network, as well as testing teleoperation with an ABB IRB120 industrial robot and a stylus pencil used by the operator.

The teleoperation performed in this paper was executed on a 5G network depicted in Figure 1.6. The 5G network operated on 28 GHz frequency bandwidth with the potential of delivering up to 10 Gbps. The operator's stylus represented the reference point for the robotic controller, which controls the robot's pencil. Reference points from the stylus were sent from the operator to the 5G CPE, to the AAU, and to Bangkok, which was over 100 km of fiber optic cables from the service provider. After the transmission reached Bangkok, data traveled back through the same link and then onto the robot controller, which set the robot's new position. This was achieved with just a few milliseconds delay. The update of the robot's position was then transmitted back through the same link to the GUI of the operator.

Experiments were conducted and aimed to mimic the precision of the robot. The robot was supposed to follow the trajectory of letters; in this scenario, "5G" was written with a star image. Results acquired in this paper showcase that teleoperation is possible with 5G. The robot did not follow a perfect trajectory but had minor errors, such as the pen of the robot was not touching the paper, or the operator was moving his hand too fast.

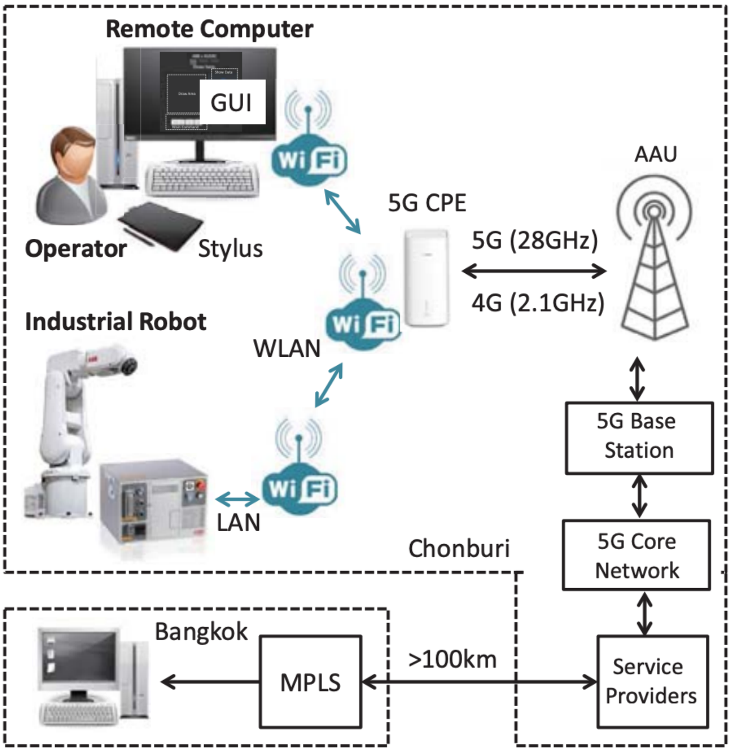


Figure 1.6: The experimental setup for teleoperation given in the paper: ”Teleoperation of an Industrial Robot using a Non-Standalone 5G Mobile Network” [45].

1.5 Outline

The thesis is intended for engineers without a telecommunication background and those with one. It provides the basic understanding of telecommunication to support those without a telecommunication background, whose interest is to move forward with this work.

The thesis structure is divided into four parts; Preliminaries, Experimental Setup and Methodology, Results and Evaluation, and Conclusion. These parts combine eight sections, representing different topics:

- Section 2 describes the fundamentals around networking theory, latency, and 5G technology.
- Section 3 presents the hardware and software used for the experimental work conducted in this thesis.
- Section 4 represents the setup and configuration of the experimental work, with test scenarios and communication setup.
- Section 5 presents the findings of the conducted experiments. This section handles the experiments with a presentation, explanation and discussion for each test scenario.
- Section 6 provides interpretations and evaluation of all the findings as a whole.
- Section 7 discusses the experimental approach and how it relates to the problem description.
- Section 8 concludes the thesis, with suggestions about researching opportunities associated with this thesis.

2 Theoretical Background

This section provides the theoretical background of this thesis. It contributes to the understanding of computer networking, latency, and the new technology of 5G. Computer Networking theory is introduced in section 2.1. An outline of the latency in connection to the Computer Networking section is provided in section 2.2, and details around the 5G networking technology are given in section 2.3.

2.1 Computer Networking

This section describes the basics of computer networking, terms, and equipment used for establishing connections throughout the Internet as is known of today. Computer networking deals with multiple interconnected devices, in scales of billions [9]. Networks are collections of computers and transmission channels that allow for communication between large and small distances [34]. This ranges from small-scaled networks, like a chipset, to a globally intercontinental transmission network.

According to Stallings, "Computer communications can generally be said to involve three agents; applications, computers, and networks." An application performs a task, e.g., sending a message, to another computer. The data transfer involves getting the message from the source computer to the receiving computer and transporting the message at the receiving computer to the intended application.

Local Area Networks (LANs) are networks covering small areas, such as houses, universities, or workplaces. LANs are the physical networks that provide the connection between computers in these places. When LANs connect to other LANs, they establish a more extensive network, connecting to other larger networks. This is the foundation of the Internet [11, p. 13].

2.1.1 Basics of Computer Networking

The procedure of communication is defined in a networking model or *protocol*. The key features of these protocols are that they take data, wraps it, and sends it. A protocol can send any data that needs to be transmitted, e.g., video stream, robot commands, and sensor readings. Typical data transmission has three stages:

- A source application passes down data through the *stack*. The data is wrapped in *packets*. Packets are then transmitted in the network.
- Transmission of data flows through the network until it reaches the destination computer, as described in the packet.
- Packets are received at the destination, which passes the packets up through the stack. This process extracts the data and passes it to the destination application [31].

Networking protocols or *stacks* handles these stages of data transmission. A more thorough description of some fundamentals of computer networking is presented in the upcoming paragraphs.

Layers and Stack A stack is a visual representation of the *layers* given in a protocol or model. These layers can correspond to programming interfaces or libraries which can only communicate directly to the layers above or below it [11, p. 13]. A representation of a data transmission structure is given in the *TCP/IP* model, which has four layers; Application Layer, Transport Layer, Internet Layer, and Link Layer. A LAN corresponds to the Link layer, and Internet Protocol (IP) corresponds to the Internet layer. Section 2.1.3 describes the TCP/IP model.

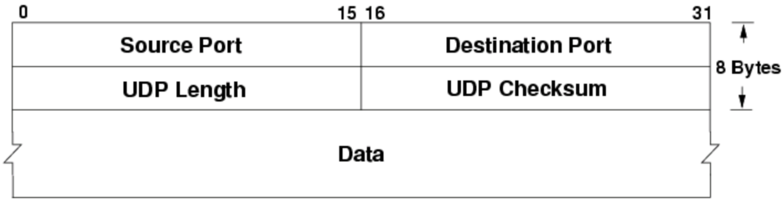


Figure 2.1: The header used for a UDP packet [44].

Packets Packets are defined-sized buffers of data transmitted through the physical link in a computer network. The maximum size a packet can have, depends on the interface and protocol, e.g., Ethernet allows up to 1500 Bytes of data per packet. These packets have a *header* which contains the delivery information of the packet [11, p. 14]. These headers are specified in the protocol or layer, e.g., the Transmission Control Protocol *(TCP)* uses 20 Bytes of the buffer size as its header. Another commonly used protocol is the User Datagram Protocol *(UDP)*, where the header uses 8 Bytes. A representation of how the UDP header takes place in the packet is illustrated in Figure 2.1.

Datagram Forwarding For *datagram forwarding* the headers contain the destination address such that forwarding processes can identify the connection for transmission. Routers and switches try to ensure this packet arrives at its intended destination. Nevertheless, packets can get lost in the transmission process, and this is further discussed in section 2.2. Switching devices that are forwarding packets on a LAN address are called switches, and such devices that are interacting with the IP layer of the network communication model, are called routers. Both switches and routers are responsible for forwarding datagrams, but the routers can connect to different, unified networks, opposed to the switches that can only forward to individual nodes.

Throughput and Bandwidth A connection provides a given data rate, and this data rate is regularly defined in bits transmitted each second (bps). *Throughput* refers to the effective transmission rate, meaning it considerate the protocol inefficiencies and traffic on the network [11, p. 14]. This makes throughput important for validating the overall connection between the source and destination. Bandwidth is commonly used as a synonym for data rate and related to the frequency spectrum for which radio transmissions and wireless communication originate. The width of the frequency band of a communication link is proportional to the data rate that can be achieved by undisturbed communication on that link.

2.1.2 Wireless Connectivity

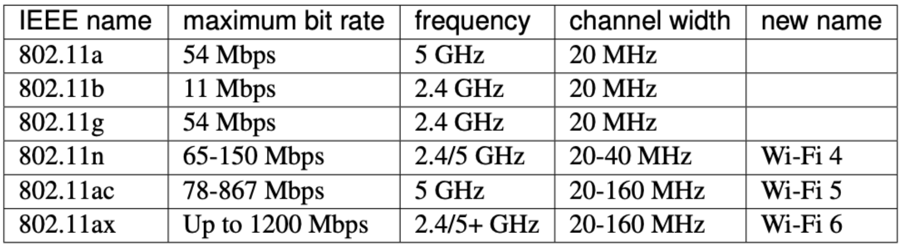
Wireless connectivity features are crucial for smartphones, tablets, and laptops in a society where Wi-Fi is the norm in homes and offices. The wireless connection can implement a slim design for equipment when removing the large Ethernet port, but more essentially, it provides mobility and versatility. Today, seeing a smartphone using Ethernet for internet access would be unimaginable.

To understand how wireless communication is established, it is vital to understand the principles of radio waves, frequencies, and wireless adapters. Radio waves are a part of the electromagnetic spectrum and travel close to the speed of light in the earth's atmosphere. In wireless communication, radio waves are generated by a *transmitter* and received at a *receiver*. Transmitters are typically routers or wireless access points (WAP) and a receiver, e.g., a network interface card (NIC) in a computer. In radio waves transmission, frequency describes the occurrence of a reference point on the waves propagating over a measurement point over a given period.

$$f = 1/T \tag{1}$$

Where f is the frequency in Hertz (Hz) and T is the period in seconds (s). Frequencies define the achievable data rate, which increases if the width of the frequency band increases. This is

Table 2.1: Specifications of a, b, g, n, ac and ax of the 802.11 standard [11].



IEEE name	maximum bit rate	frequency	channel width	new name
802.11a	54 Mbps	5 GHz	20 MHz	
802.11b	11 Mbps	2.4 GHz	20 MHz	
802.11g	54 Mbps	2.4 GHz	20 MHz	
802.11n	65-150 Mbps	2.4/5 GHz	20-40 MHz	Wi-Fi 4
802.11ac	78-867 Mbps	5 GHz	20-160 MHz	Wi-Fi 5
802.11ax	Up to 1200 Mbps	2.4/5+ GHz	20-160 MHz	Wi-Fi 6

displayed with $B = 2 \times f_m$, where B is the bandwidth, and f_m defines the frequency of a simple wave. By increasing the frequency of the sine waves, the bandwidth must increase proportionally. Regardless, if the bandwidth increases, the frequency must not increase, but the bandwidth can fit more sine waves of that frequency.

The efficiency of the data rate over a bandwidth is limited by the Shannon-Hatley theorem, which says that the upper limit for the channel's capacity is given by

$$C = Blog_2(1 + \frac{S}{N}) \tag{2}$$

Where C is the channel's capacity, in bits per second, B is the bandwidth of the channel, and $\frac{S}{N}$ denotes the signal-to-noise ratio, which is another term for the quality of the signal.

There are multiple different standards for wireless connectivity. In the following paragraph, two are explained; Wi-Fi and Cellular Networks.

Wi-Fi Wi-Fi or 802.11 is a protocol developed by the Institute of Electrical and Electronics Engineers (IEEE). In the earlier years, Wi-Fi only used the 2.4 GHz band, before the 5 GHz band was introduced [11, p. 94]. This 5 GHz band did provide more channels for communication and thus less interference. Nevertheless, the 5 GHz waves did not travel through walls with the same efficiency as the 2.4 GHz waves. The specifications of the variations of the 802.11 Wi-Fi standard are listed in Table 2.1. The table shows bit rates in optimal conditions, meaning no interference with the single stream of bits on the band. A maximum bit rate such as the one in the table is rarely achieved. Noise, errors, obstructions, and interference on the Wi-Fi channels are reasons for reduced bit rates.

Cellular Networks Cellular networks, like 4G and 5G, rely on the wireless connectivity of devices. The cellular network 4G or LTE (Long Term Evolution) were intended to use for mobile devices but are also available for some stationary devices [11]. 4G uses, for the most part, the licensed spectrum and follows the same principles as Wi-Fi when it comes to data rate, bandwidth, and frequency. Relay stations connect to a central station or base station that provides the area's network signal. A cellular network area can range between one to ten kilometers in radius, thus much larger than a typical home or office Wi-Fi. As distance increases, the data rate is reduced. Both the base station and relay stations can provide cellular network connections to the surrounding nodes. An illustration of a cellular network is given in Figure 2.2, with cell phones as representations for surrounding nodes.

Commonly used terms for describing the direction of transmission are *Downlink* and *Uplink*. Downlink is when transmissions go from the base station to a subscribing node, and uplink is when the transmission comes from the subscribing node. Uplink usually has a lower data rate than the downlink due to a node's wireless transmitting specifications compared to the base station.

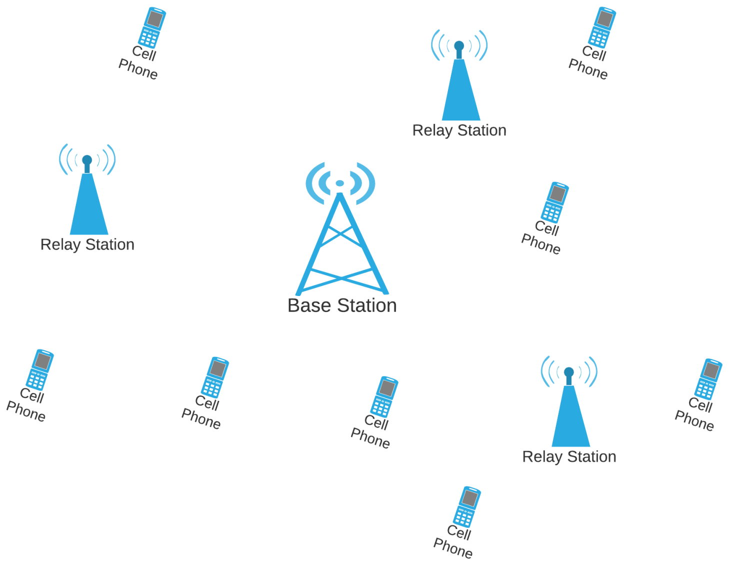


Figure 2.2: Illustration of a cellular network, with base station relay stations and cell phone nodes.

2.1.3 TCP/IP

Models describing communication are essential for the interconnecting devices of a network. Such models describe the protocols, layers, and structure of a communication network. They also define the pipeline for packet transmissions and indicates where delays can occur from node to node. This section describes the TCP/IP model and explains how the Open System Interconnection (OSI) Model compares to TCP/IP. TCP/IP is a communication model and was developed by the US Department of Defense. This model was first called Advanced Research Project Agency NETwork (ARPANET) and was later developed to TCP/IP. From 1983 the network was called the *Internet* [4].

The TCP/IP-communication model can be divided into four layers; Network Access Layer/Link Layer, Internet Layer, Transport Layer, and Application Layer. This is illustrated in Figure 2.3. The layers are described as following according to Alani;

Link Layer This layer covers the physical interface, transmission, and receiving of data between the physical component and the connected network. The layer uses the Media Access Control (MAC) address, a unique address given to a NIC of a computer system. The layer maps the connection between the IP addresses from the Internet Layer and forwards data to the hardware's addresses, such as the MAC address.

TCP/IP does not manipulate this layer much because TCP/IP only cares for delivering IP packets through this layer. Moreover, most of this layer is pre-configured software and drivers from the hardware producers. This layer uses the WAN and LAN interfaces to connect the packet stream from endpoint to endpoint physically.

Internet Layer This layer contains procedures to allow data to be transported across interconnected networks. The Internet Protocol (IP) defines a packet and addressing scheme, includes transport data between the Link Layer and Transport Layer. The layer uses the IP to route data, meaning the protocol chooses the best path from endpoint to endpoint.

The IP protocol relies on higher-level layers, such as the transport layer, to assure acknowledgment of delivered packets and error detection and corrections. This means that it is a connectionless

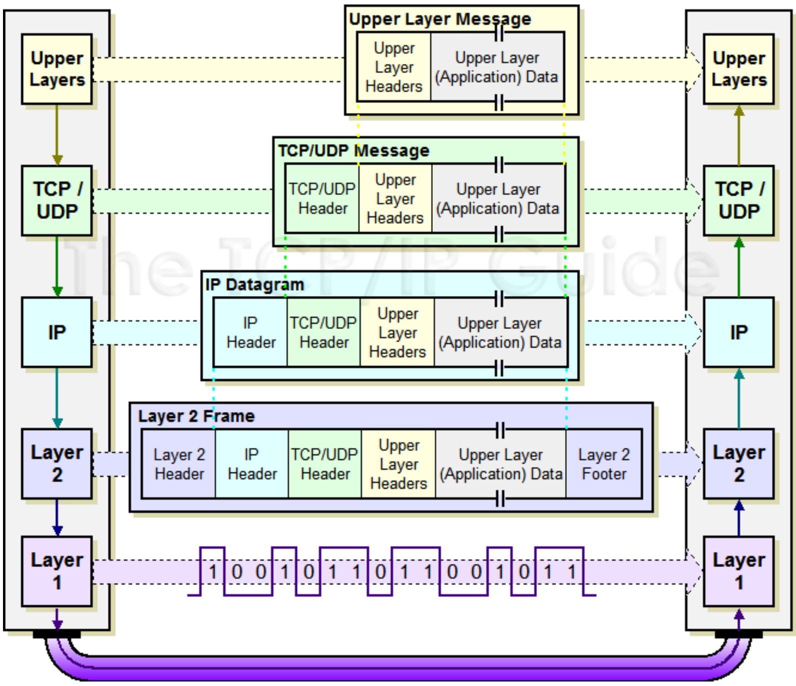


Figure 2.3: The TCP/IP model with layers, headers and packets [31].

protocol. The internet layer is often implemented in end systems and routers, e.g., computers and databases.

Transport Layer The Transport layer provides end-to-end communication. The transport layer delivers the data to the proper process, given each unique address within the host. The Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) are commonly used protocols to communicate between endpoints. This communication is often called socket or client-server communication. Socket communication is where one of the sockets is listening to a port, whereas the other one is trying to connect to that port [16, 39].

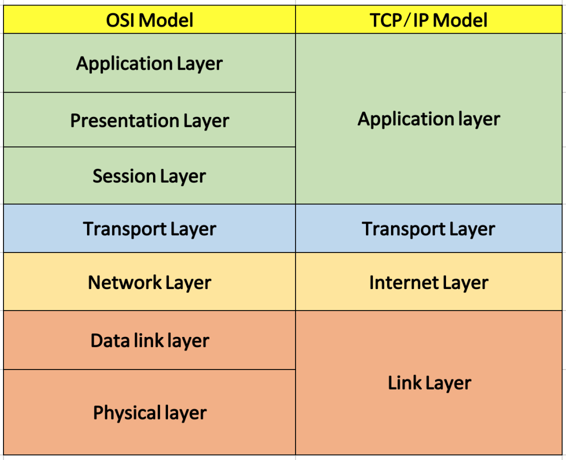
TCP utilizes a three-way handshake, where the client sends a request to the server, which is listening to a given port. The server then responds to the client with connection information, of which the client confirms the reception of the connection information [16]. This ensures reliable transmission of data throughout the connection. TCP uses numbers to identify each package, such that the data can be reconstructed if the packet is lost. Confirmations are sent with a sequence number, which gives the sender the possibility to detect when data is lost and then re-transmit the same data again [39].

UDP is a connectionless protocol like IP, meaning that it does not connect with the other socket. It does not guarantee delivery, protection against duplicates, or the correct sequence of the data. The protocol can provide checksums for data integrity but is optional, and UDP is often used for time-sensitive applications, e.g., with video and audio synchronization. Allowing bits to get dropped for the benefit of the fast round-trip time (RTT) [39].

Application Layer This is the logical connection that supports the user interface and application. This layer makes up the fundamentals in communication in the application for the user, such as representation, encoding, and dialogue control. The application layer contains multiple protocols for the user, such as web browsing (HTTP, DNS) or mail (SMTP).

For the most part, the Application layer protocols are simple text conversations established between user and destination. A goal is to make the Application layer protocols universal so that that protocol data can travel through as many devices as possible.

The OSI Model The OSI model is built from ISO Standardization ISO/IEC 7498-1. The model is a basis upon which independent developers can contribute and still maintain consistency within the communication frame [39]. The OSI model differs from the TCP/IP model to separate the application layer into an application, presentation, and session layer. Similarly, it defines the Network Access/Data link layer as one Data Link Layer and one Physical Layer. Due to the OSI model being seven-layered, it can be easier to customize protocols to fit different needs. A visualization of the TCP/IP model and OSI model is given in Figure 2.4.



OSI Model	TCP/IP Model
Application Layer	Application layer
Presentation Layer	
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data link layer	Link Layer
Physical layer	

Figure 2.4: Comparison of the OSI model and the TCP/IP model [35].

2.2 Latency in Computer Networks

From [18], one-way latency is defined as ”the time, which elapsed since a message is sent by the source node to it is received a message on the destination node.” Networking latency is usually given in milliseconds (ms). It is essential to distinguish between one-way latency and Round-Trip time (RTT), which includes the time it takes for the destination node to process and send a message back to the source node. It is also crucial to state that the latency does not affect the quality of the data being transmitted but can still affect to workload a transmission line can process [18]. The latency is unstable and can be time-variant, which can cause variable load on the network. For real-time data transmission, such instability can cause bits to disappear, e.g., frames from a live video.

2.2.1 Encapsulation and Routing

The encapsulation process of packets transmitted throughout the layers in the TCP/IP model is commonly known as the TCP/IP Stack [31]. Data from the application layer is passed down through the stack like shown in Figure 2.3. In this process, the data is wrapped in packets, which are then transmitted to the network. Packets are then physically passed through the network as bits, displayed as the purple line in Figure 2.3, until they reach their destination computer. At the destination, the packets move up throughout the stack until they are unwrapped in the application layer [31].

2.2.2 Delays in networking

When exchanging packets over the network, latency occurs, regardless of how good the hardware, bandwidth, and connection are. Some of the delays, another name for latency, are more significant than others. A listing of different delay types is mentioned, also discussed in [31].

Application Layer Delay: This latency occurs in the application layer in the TCP/IP model or layer 5-6-7 in the OSI model. This latency can be due to the processor's speed or the efficiency of the software running in this layer.

Serialization Delay: The encapsulation process demands a finite time to run. This is highly dependent on the packet size and the transmission rate. This is represented in 3

$$\tau_{SD} = \frac{PacketSize[bits]}{TransmissionRate[bits/s]} \tag{3}$$

Where τ_{SD} is the Serialisation Delay, typically given in $[ms]$ or $[\mu s]$.

Routing Delay: Forwarding IP packets from the transport layer through the physical layer demands routing and switching. In an extensive network, this delay can be of significant value. Rerouting packets when a connection breaks can make the packet travel much further than needed initially.

Queuing Delay: After the packet's routing is processed at the router, the package must wait in queue if the outgoing link is overused, e.g., occupied bandwidth.

Propagation Delay: Propagation delay arises when the physical properties of the transporting-medium slow down the data flow. The slowing is due to a *velocity factor* (VF). The velocity factor is a measurement of how fast the data flows through the medium, compared to the speed of light. For example, a fiber optic cable transmits at 70% of the speed of light. Propagation delay, τ_{PD}, given in $[ms]$, is specified by Equation 4.

$$\tau_{PD} = \frac{Distance[m]}{Speed[m/s]} \tag{4}$$

Transmission Delay: Transmission rate, also called bandwidth, describes the number of bits that can be extracted from the transporting medium. The destination computer endures transmission delay due to the extraction of the packets. This delay is highly dependent on the performance of the physical layer, which determines the transmission rate.

2.2.3 Real Time Synchronisation

In a real-time system, clock synchronization is essential for developing a low-latency wireless network system. Problem with time synchronization can often result in a delay of data transmission, and reception [3]. Synchronization of two nodes is vital to guarantee the correct real-time behavior of the system [14]. An industrial plant may consist of various nodes for controllers, actuators, sensors, and communication, which induces clock synchronization between the nodes.

A clock is composed of two parts; an oscillating device that decides the clock's frequency, determining how long a second is. The second part is an accumulator that counts the seconds or the oscillator's cycles [14].

A clock can not be perfect since no oscillator is perfect, i.e., exposed to variations from environmental measures. This means the frequency accuracy and stability can dynamically change, and thus the time accuracy and stability [14]. The frequency accuracy is a measurement of how well the oscillator realizes the predefined length of a second. Frequency stability represents how well a clock maintains the same frequencies between two points in time.

Networking Time Protocol

Since all clocks are non-ideal (drifts), extensive systems can accumulate large amounts of delays due to asynchronous oscillators. One protocol for handling the synchronization of clocks over the network is the Networking Time Protocol (NTP).

NTP is a protocol used for synchronizing network clocks using a set of distributed clients, and servers [27]. NTP provides a protocol to synchronize time with the precision of nanoseconds and is a part of the application layer of the TCP/IP model. The protocol contains provisions to specify the precision of synchronization, which also estimates the error of the local clock regarding the clock it should be synchronized with [27].

The NTP protocol is organized into a hierarchy. Levels in the hierarchy are defined as stratums. These stratums are labeled stratum [0,1,2...,n], where stratum 0 is the server for stratum 1 and defines the synchronizing clock for stratum 1. This is typically a server connected to the GPS network. The NTP supports a given number of stratums, and typically the NTP time represents 64 bit, where 32 bit describes seconds and 32 bit is the fractional parts of the seconds [14].

The functionality of synchronization with NTP is based upon the client asks a time server for the time in a packet. The client stores a timestamp and the packet. The server receives the packet and sends the timestamp in a packet back to the client. When the client receives the timestamp again, the client logs it and estimates the traveling time. Further, the client must validate the estimation with a "sanity check," which is a comparison with other clients of that server [14]. The expected accuracy of synchronization over NTP is 10-20 ms for WAN and < 1 ms for LAN [2].

Precision Time Protocol
Another protocol used for synchronizing clocks is the Precision Time Protocol (PTP). Similar to the NTP protocol, PTP may also utilize the GPS satellite network as its highly accurate time source [24]. The PTP is a newer standard than the old NTP protocol and is more commonly used in the industry for control and more time-sensitive tasks. This is because it can achieve synchronization down in microseconds, compared to NTP's millisecond accuracy. PTP is also designed for local networks, with a single Grand Master Clock that synchronizes to the GPS satellite network.

PTP also consists of stratums, but the reference clock (typically the GPS satellite) is called stratum 1 instead of stratum 0 for NTP. The grandmaster clock is stratum 2, and connects to several master clocks. Indifference to NTP, the hierarchy is not multiple paths. The master clocks only have *one* slave clock, thus no negotiation between the clocks. The master clock calls the time to the slave. The hierarchy is illustrated in Figure 2.5.

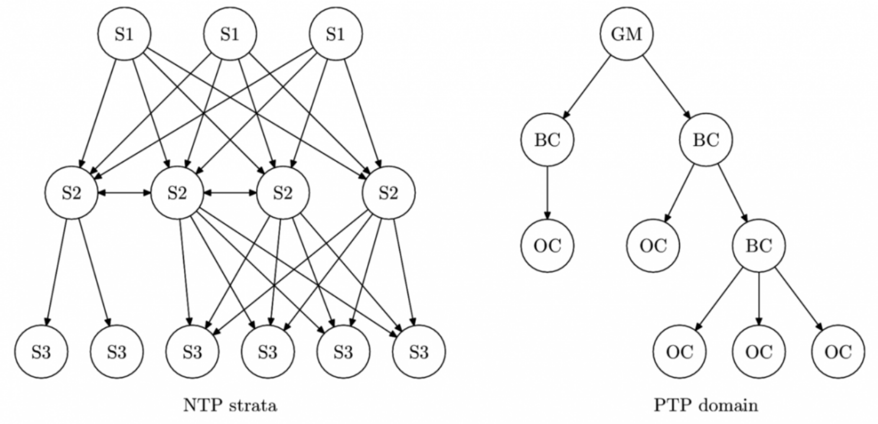


Figure 2.5: The hierarchy of NTP and PTP [24].

2.3 5G

The fifth-generation (5G) wireless communication technology was expected to be deployed in 2020 and is still being deployed worldwide. The 5G network is inducing expectations of a more reliable network for the user. Nevertheless, the network is just another cellular network that employs cells (base stations and relay stations) and data transmissions over radio waves. This section provides an overview of the 5G network and the serviceability it provides.

2.3.1 Introduction to 5G

The 5G interface provides users with multiple features. Some fundamental properties are high throughput, improved spectrum efficiency, reduced latency, better mobility support, and high connection density [13]. The increased throughput demands a higher frequency and 5G support *mmWave*, which is located in the spectrum band between 30GHz - 300 GHz.

The growth of the Internet of Things (IoT) and Industrial IoT (IIoT) requires higher throughput and more reliable services, which 5G can provide [13]. For mission-critical communication, latency requirement and reliability is crucial. For services like industrial safety systems in process plants, this can mean life or death. 5G can provide the needs for these industrial applications. Other new use cases for networks are developed from the qualities of 5G [38]:

- Latency requirement - Industry, Surveillance, real-time control, Virtual and Augmented Reality.
- Reliability requirement - electrical grids, e-health, and other services which are highly sensitive to downtime (low reliability).
- Throughput requirement - Video Streaming, Cloud-based services.
- Network Slicing - Dynamically allocating bandwidth to needed services
- Low battery consumption - Provides a possibility for wireless sensors and small devices.
- Scalability and Mobility - Provides the full potential for all things to be connected, anywhere, e.g. massive machine-type communication (mMTC).

The 5G retailers provide the expected Key Performance Index(KPIs) for customers. These KPIs define the Quality of Service (QoS) of the 5G network. The 5G network is presumed to be 10-100 times faster than 4G, achieving latency < 1 ms and a data rate of 10 Gbs. It is also going to support hundreds of billions of machines [38].

2.3.2 5G-Technology

The cells of the 5G network are connected to the network backbone, just like an ordinary cellular network. The 5G network uses the 4G network infrastructure, such as fiber optic cables, cellular towers, and interconnections. This means the 5G network is depending upon the 4G network to function correctly. Thus, a 5G Standalone (SA) network is an end-to-end 5G network, meaning it is not reliant upon the 4G core structure to function. This is currently under development and testing [37]. This means that the 5G connections currently established with a Non-Standalone (NSA) might not provide an end-to-end 5G connection. The difference between NSA and SA 5G is illustrated in Figure 2.6. The Evolved Packet Core (EPC) is the framework for the LTE network. A SA 5G network is independent of the current LTE coverage in a given area, and according to [40] it is suspected to have less latency and realizes all use cases for 5G. Compared to this, the NSA only supports eMBB. A SA 5G network can provide the industry with private networks used for high throughput, low latency, and good coverage with multiple devices, where they can distribute IP addresses locally for secure 5G communication.

The 3rd Generation Partnership Project (3GPP) is an organization that provides communication standards for cellular networks, and they are currently defining the standards for 5G networks. This network is based upon a system architecture that uses services as functionalities. One of the primary services that 5G provides is network slicing, which makes a whole Public Land Mobile Network (PLMN) configurable [26]. This means, e.g., a network provider can allocate bandwidth such that users are not interfering with the necessary equipment that requires bandwidth to function. 5G opens up opportunities for IoT development and digitization of the industries. URLLC and mMTC can make large crucial industries join the IoT, and also, the low battery consumption strengthens the capabilities of small electronic equipment in the industries.

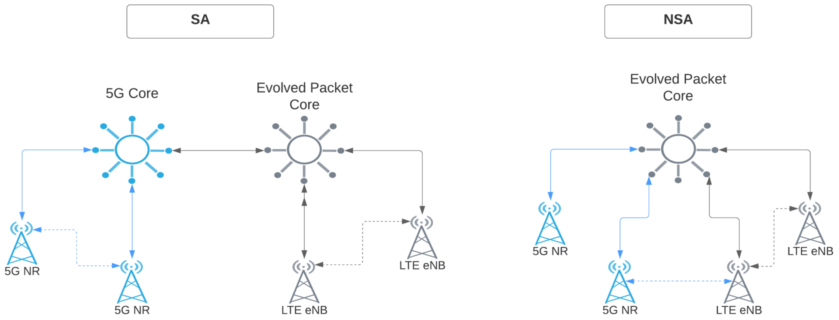


Figure 2.6: Comparison between the Standalone and Non-Standalone 5G Networks.

The 5G NR (New Radio) is the way 5G technology was standardized by 3GPP and is a Radio Access Technology (RAT), which the radio waves travel. This standardization is separated into two frequency bands. The Frequency Range 1 (FR1) sub-6 GHz frequency band provides connectivity for 5G networks operating beneath the 6 GHz mark. The other Frequency Range (FR2) includes electromagnetic waves in the millimeter spectrum (mmWave). This incorporates frequencies between 24-100 GHz and provides. The 5G frequency bands are commonly referred to with an "n-number," e.g., n1, n2, n3,.., nx.

2.3.3 5G mMTC

5G is a game-changer for the IoT, and the concept of mMTC is an important factor. The lower frequency bands of FR1 supply a large coverage area, along with a reliable wireless connection to small devices. Narrowband IoT (NB-IoT) is a standard issued by the 3GPP, which provides low energy consumption, low cost, wide coverage requirements for the IoT, and thus utilizing the low-frequency bands [6]. With 1 million interconnected devices/km^2, security is important for ensuring the authenticity of the network users [30].

The importance of low power consumption for the technology is coming from the power capabilities of small devices. These devices are vital for achieving the mMTC solution, and from here, the term Low Power Wide Area Networks (LPWAN) emerges. NB-IoT is meant for low data rate and can tolerate delay, for the advantages that low power consumption gives [30]. Frequency influences the power consumption of devices, along with the size of packets that are being transmitted. The frequency channel bandwidth distributed to each device must be small enough to fit as many as possible devices for the given bandwidth. Therefore, essential optimizations for protocols of small devices are crucial for providing the considerable coverage and low power consumption promised by the 5G technology.

The low 5G frequency spectrum is typically located between 600-850 MHz. These frequencies provide more extensive coverage areas because of better propagation of the electromagnetic waves with longer wavelength, previously discussed in section 2.1. This promotes lower-cost deployment, given that each base station can be placed farther away from each other, lowering the number of base stations needed. Also, transmission at higher frequencies requires better specifications, meaning the base stations will be cheaper for production. With the 5G protocols and interconnectivity of mMTC, these low-frequency bands can cover whole countries, with billions of devices, in one extensive wide area network.

Experimental Setup and Methodology

PART 2

3 Hardware and Software for the Telecommunication Setup

A telecommunication setup is defined to establish the fundamental architecture of the thesis project. An illustration of the architecture, with no regard to additional information around configuration, is shown in Figure 3.1. This figure shows different transmission paths further shown in section 4 Table 4.1. This section describes the hardware and software that were used for this system setup.[2] How these hardware and software were configured for the experimental setup is further discussed in section 4. Section 3.1 provides information about the Raspberry Pis and the 5G HAT that provides the 5G connectivity for the RPi. Section 3.2 describes the robotic system, and section 3.4 describes the analytic software that handles measurements and analysis. Information around phones and 5G technology, with a focus on the iPhone 12, is given in section 3.3.

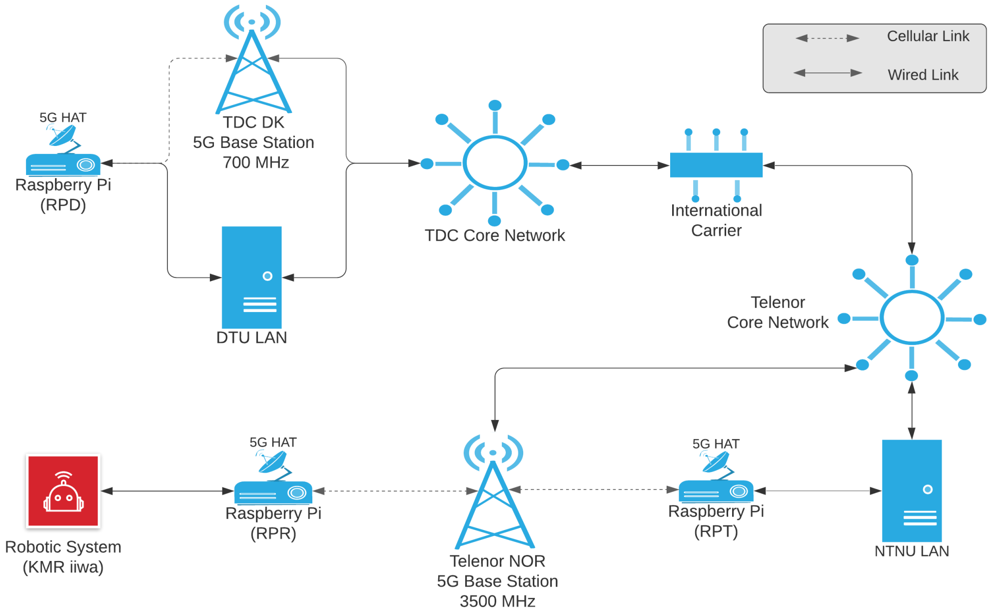


Figure 3.1: The Data Transmission Network used for experimental setup, and visualizes of how the robot can be controlled remotely.

3.1 Raspberry Pi with 5G HAT

This section describes the Raspberry Pis used in the experimental setup. An introduction to Raspberry Pi as a computer and the usability of the small computer with regards to IoT is presented in section 3.1.1. ROS2 is used for control of the robotic system, a description of ROS2 is given in section 3.1.2. For a Raspberry Pi to function with 5G, a 5G HAT (Hardware Attached on Top) is mounted on top. How this is set up and functions are described in section 3.1.3.

3.1.1 Introduction to Raspberry Pi

Raspberry Pis are small, low-cost Single Board Computers (SBC), which were developed to advance computer systems and industrial applications [8]. Raspberry Pis usually runs Linux with either Ubuntu or Raspberry Pi OS as its operating system. It can run Real-Time Operating Systems (RTOS) or tweak the operating systems into an RTOS, e.g., patching the kernel. The newer models of the Raspberry Pis are delivering high performance with an ARM 64-bit quad-core processor delivered by the company Broadcom. The performance of ARM processors has vastly increased over the past years, and the company Apple made a huge breakthrough with their release of the

---

[2] This thesis is a continuation of the author's previous works, parts of this section are assembled from the author's report "Remote operations and testing of KMRiiwa using Raspberry Pi 4 with Wi-Fi"[32].

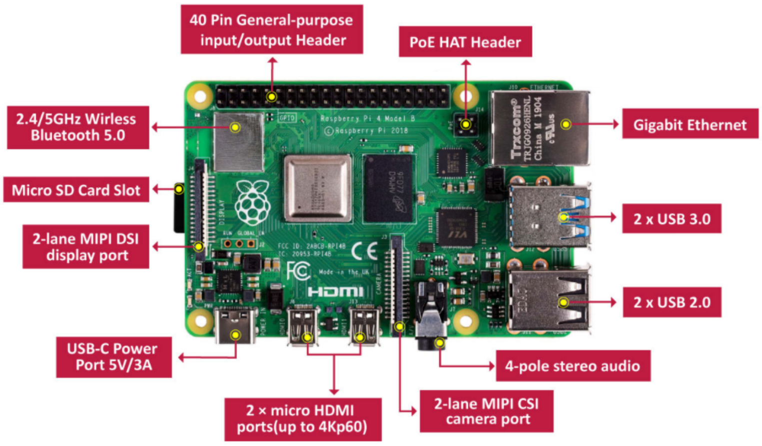


Figure 3.2: Overview of the Raspberry Pi 4B's components [36].

M1 ARM chip for their Macbooks in 2020. This development reflects the need for ARM chips and their compatibility in the future.

The Raspberry Pi 4B (RPi) is the newest version of the standard Raspberry Pi boards. An overview of the Raspberry Pi 4B model is shown in Figure 3.2. The USB-C Power Port (5V/3A) enables the RPi to use power banks instead of a wall outlet and can therefore be utterly mobile if needed. The portability of the RPi is an advantage when it comes to industrial capabilities; hence mobile robotic systems can profit from this.

The RPi uses a 40 GPIO Header connector that can be used for attaching HATs. This port allows third-party companies to make their add-on boards that fit the protocol of the RPi 40 GPIO. Also, configurations of third parties hardware are accessible with the 40 GPIO Header, and multiple RPi's can be set up in the same way, and can therefore e.g., achieve an mMTC 5G setup with the use of multiple 5G HATs and RPi's.

Since the RPi does not come with any display for Graphical User Interface (GUI) purposes. The Raspberry Pi comes with 2 HDMI ports and the ability to use Remote Desktop or Virtual Network Computing (VNC) for enabling GUI. In cases where the GUI is not needed, the RPi can be set up using the SSH (Secure Shell) protocol to connect to the command line of the RPi.

3.1.2 ROS2

ROS is an abbreviation for Robot Operating System and is an open-source operating system built upon a higher level of programming that lets the user easier manipulate the robot without dealing with the hardware.

ROS2, built on ROS (first version), is a collection of tools and libraries that helps robotic software development [12]. Structure of ROS2 is depicted in Figure 3.3.

ROS2 is programmed in many languages, but the ROS-team typically maintains the rclpy (Python), and rclcpp (C++) libraries [7]. Nodes are written in these program languages and communicates via

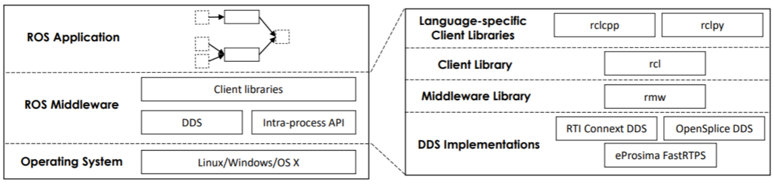


Figure 3.3: The layered structure of ROS [10].

a ROS master node. ROS2 exchanges messages using the transport layer called TCPROS based on TCP/IP. The master defines paths for nodes to find each other and communicate. Communication between nodes uses *topics* as a bus, based on the publish-subscribe pattern [7]. A node specifies which topic it is interested in and then receives the information from the publisher node the topic is listening to. A node can both be a subscriber and a publisher.

The development of ROS2 was focused on improving the real-time capabilities of ROS. ROS2 is built on top of the Data Distribution Service (DDS), which allows any two DDS programs to make contact. DDS enables implementation of ROS2 in industrial use cases but also enables possibilities for real-time publish-subscribe (RTPS) wire transfer protocol, which is a protocol that does not need to connect to the master node [12]. One goal during the development of ROS2 was to safely implement time-critical control paths inside the software [10]. These control paths require prediction of the end-to-end latency (which often is called response time) of time censorious processes, such as the process from receiving odometry data to manipulate them. ROS2's end-to-end latency in a local ROS2 system (no transmission delay) is predicted to be $< 1ms$, i.e., good real-time capabilities[46].

3.1.3 SIM8200EA-M2 5G HAT

The SIM8200EA-M2 is a wireless communication module produced by SIMCOM, equipped with the Qualcomm Snapdragon x55 chip, focusing on the 5G market [25]. A 5G HAT produced by Waveshare utilizes this module for a 5G connection, and it is interconnected with an M2 connector. The 5G HAT board also features a 3.1 USB port, audio jack (for phone calls), SIM card slot, and cooling fan. The SIM8200EA-M2 module supports 3G/4G/5G (sub-6 GHz). Six antennas are connected to the module, five used for cellular networking and one used for GPS signal. The 5G HAT can achieve 4 Gbps for downlink (DL) and 500 Mbps for uplink (UL) with 5G NR, according to Waveshare. Likewise, for 4G, it offers 2 Gbps DL and 200 Mbps UL [43]. A fully assembled Waveshare SIM8200EA-M2 5G HAT with a Raspberry Pi is displayed in Figure 3.4.

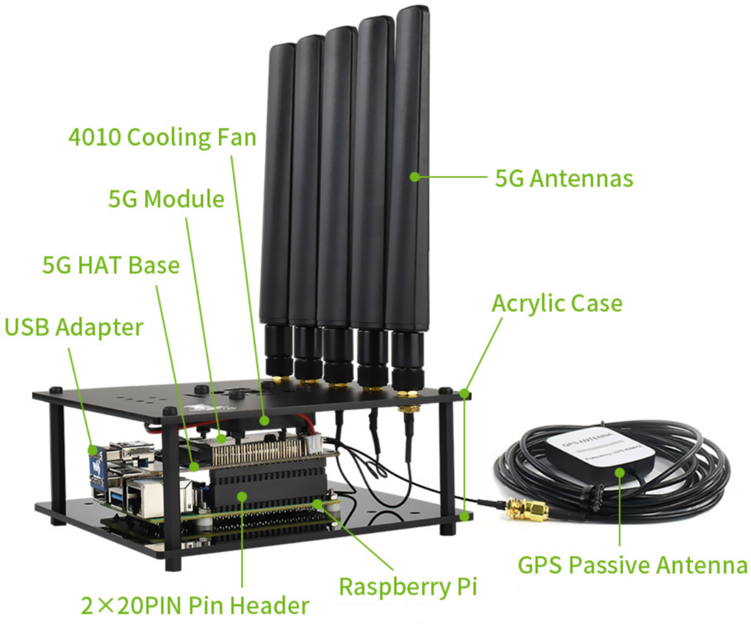


Figure 3.4: A fully assembled Raspberry Pi and Waveshare SIM8200EA-M2 5G HAT setup with components [43].

The operating frequencies of the 5G HAT's antennas are between 617 MHz - 5000 MHz. It can also act as both the receiving and the transmitting unit, making it also capable of providing 5G network connections for other devices. The HAT also provides functionalities such as phone calling and sending SMS.

Figure 3.5: The Communication Protocol between the Raspberry Pi, 5G HAT and the internet [43].

The 5G HAT connects to the Raspberry Pi via the 40 GPIO of the Raspberry and the 40 Pin Header of the HAT. The USB port is used as a communication link between the HAT and the RPi. This USB adapter provides a 5G/4G/3G connection from the 5G HAT as shown in Figure 3.5. AT commands are commands sent from the RPi to the SIM8200 modem on the 5G HAT, which are instructions that control the modem, these AT commands can configure the modem to fit a specific network wanted for connection. Minicom is a command-line program for Linux that sends AT commands to the module. The configurations made with AT commands from Minicom specifies how the internet connection is established. The AT commands are used to unlock the SIM card provided by the Internet Service Provider (ISP) and specify which frequency band to use for connectivity.

3.2 KMR iiwa

This section describes the fundamentals of the physical robotic setup (KMR iiwa) for the test bench at Manulab NTNU. It is divided into an introduction to the AGV in section 3.2.1, which explains the two robots that define the KMR iiwa. Furthermore, a description of the Sunrise Cabinet in section 3.2.2.

3.2.1 Introduction to KMR iiwa

The KMR iiwa is short for KUKA Mobile Platform intelligent industrial work assistant. The KMR iiwa consists of one mobile platform and one robotic manipulator. The platform is a KMP 200 omniMove which an LBR iiwa robot is mounted on top, both produced by KUKA, hereby referred to as KMP and LBR, respectively.

The primary purpose of the KMR iiwa is to be used in industrial settings, where an automated manufacturing unit can increase the efficiency of the company more than a traditional conveyor belt. The joint movement of the LBR is entitled to a stationary platform, which means that the KMP can not move for the manipulator to work. Equivalently the KMP can not move if the LBR is not parked in a specific position.

The hardware responsible for the control of the KMR iiwa is the Sunrise Cabinet. This cabinet is located inside the mobile platform and relies on two PCs for motion and control; the KUKA control PC (KCP) and the Navigational PC. The cabinet also has an X19 smartPAD interface which allows connection for a smartPAD that can jog the KMR iiwa manually or launch applications for it to run.

KMP 200 omniMove The KMP 200 omniMove is a mobile platform used for loading and mounting the LBR. The platform can manipulate movement in two dimensions simultaneously due to the four Mecanum wheels, which gives the platform its name omniMove, which means omnidirectional movement. A picture of the KMP 200 omniMove is shown in Figure 3.6.

Figure 3.6: KMP 200 omniMove [22].

The KMP is equipped with two SICK s300 sensors used for safety and navigation. These scanners operate with a 270°scanning area each, for up to 30 m. The scanners map a safety field that is dynamically allocated as the velocity of the KMP changes. This area consists of a Warning Area (outer zone) and a Protected Area (inner zone). In the warning area, the vehicle with reducing the velocity, and in the protected area, the vehicle will stop. These areas are shown in Figure 3.7 with corresponding sensors B1 and B4.

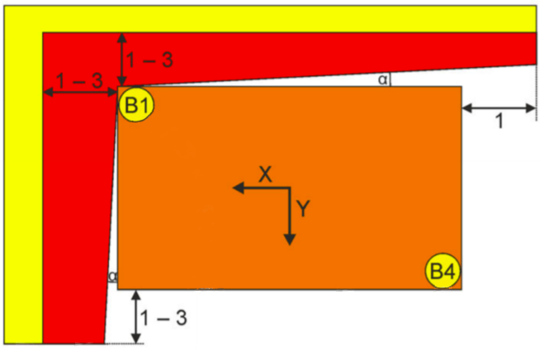


Figure 3.7: Warning and Protected area of B1 SICK sensors scan [20].

LBR iiwa 14 R820 The LBR iiwa (Lightweight robot intelligent industrial work assistant) is a lightweight robot intended for handling tools or lifting components or products. The LBR obtains seven axes (7 DOF) controlled by the Sunrise Cabinet inside the KMP. Different sensors monitor every axis; axis range sensor, axis load senor and axis temperature sensor [19]. A representation of the LBR manipulator is shown in Figure 3.8.

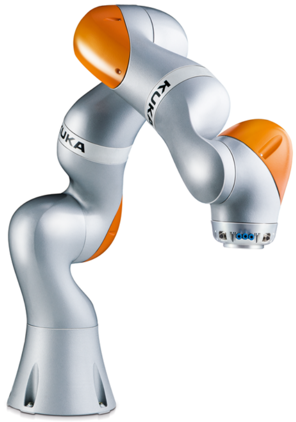


Figure 3.8: LBR iiwa 14 R820 [19].

The Sunrise Cabinet controls the LBR and can control it with a ± 0.1 mm precision. Max load is 14 kg, and the length of reach is 820mm, represented in the title of the robot (LBR iiwa 14 R820). The LBR does not feature any inbuilt security features when it comes to the area of movement. The LBR can move in a linear motion, circular motion, and point-to-point motion.

3.2.2 Sunrise Cabinet

The KUKA Sunrise Cabinet is installed inside the KMP. Its purpose is to operate a KUKA ax, which is, for example, industrial robots as the LBR and mobile platforms as the KMP [21]. The Sunrise Cabinet consists of a Control PC, Navigation PC, and a connected KUKA smartPAD. The cabinet and the smartPAD are shown in Figure 3.9.

Figure 3.9: KUKA Sunrise Cabinet with the KUKA smartPAD [21].

KUKA Control PC The KUKA Control PC (KRC) is a computer that runs Windows Embedded Standard. The KRC also runs robotic applications that interpret the data and transmit commands throughout the busses to the robot's actuators. The KRC is the main interface for external communication of the KMR iiwa. The KRC's runs a Quad-Core Intel CPU and has 2 GB of ram. The KPC contains both a real-time Ethernet connection and a Wireless Access Point (WAP) for connecting to Wi-Fi connections. The Ethernet interface (KLI) of the KRC is located on the rear of the KMR iiwa and administrates together with the WAP, external communication to the KRC. The KRC is also responsible for handling the Human-Machine Interface (HMI) of the

KUKA smartPAD.

KUKA smartPAD The KUKA smartPAD is connected to the Sunrise Cabinet through an interface on the rear of the KMR iiwa. The smartPAD has all the operator control and displays the different functions required for the operation [21], and the smartPAD is shown in Figure 3.9. The smartPAD is a touch panel that can be operated with either finger or stylus and is the display for the KRC. It allows jogging of the KMP and the primary HMI between the operator and the KMR iiwa. The smartPAD can launch applications made in Sunrise Workbench. These applications realizes remote operations, automated operations, and more configurations of the system. The smartPAD can also be used the access the windows embedded system running on the KRC when closing the HMI application running on top of it.

Additional hardware On top of the KMR iiwa's out-of-box modules, an onboard computer, a robotic gripper, and three cameras are installed. These components have been installed to increase the robot's usability and analyze different perspectives of utilization in different software.

The onboard computer is an Intel NUC installed in a predefined space on the KMR for additional modules for the KMR. The NUC runs Ubuntu with ROS and communicates with RTPS, which is further described in section 3.1.2. This onboard computer is responsible for power and communication between the three Intel Realsense D435 cameras and communication for the Robotiq 2F-85 Gripper. This is further discussed in section 4.

3.3 Telecommunication in Phones

ISPs worldwide focus on marketing 5G technology in phones, which is because today almost everyone owns a phone. 5G is currently being implemented in phones and is still under development. In particular, phones have been criticized for their cellular capabilities due to insufficient antenna specifications. Mobile technology is one of the significant market contributors and developing industries. This section provides a correlation between the mobile technology and the 5G capabilities and the iPhone.

Release 15, delivered by 3GPP, provided the first complete set of 5G standards and initiated the upcoming work for release 16. Release 15 was defined as a standardization of the 5G system - Phase 1 and included the NR standardization.

Qualcomm is a worldwide distributor of mobile network modems to phones. Qualcomm adopted the 5G technology of NR, and their first-generation 5G chip called Snapdragon x50 was seen in multiple phones later on. This chip was the first 5G modem and was built on a ten-nanometer process. This was the start of the 5G integration in phones [28].

Today chip-sets have moved forward, and the new chip from Qualcomm, x65, will provide extensive 5G performance. This chip is promised to deliver speeds over 10 Gbps on both NSA and SA. It will support release 16, which includes standardization of URLLC, and for an industrial IoT. This modem will also support wider bandwidth channels at 200 MHz [33]. Nevertheless, x65 has not yet been released, and the x60 is the newest modem by Qualcomm (upgrade from x55). Currently, only phones released in early 2021 have had access to this modem [41].

3.3.1 The iPhone 12

The iPhone 12 is a phone designed and developed by the company Apple. It is the fourteenth generation of the iPhone lineup. The CPU of the iPhone is a 64-bit Apple A14 Bionic ARM chip. The iPhone 12 brought 5G to the iPhone's and managed to do so by the integrated x55 chip from Qualcomm. This chip is communicating with Apple's antenna framework, delivering 5G NR connectivity in both Sub-6GHz and mmWave frequencies [41].

3.4 Analytic Software

This section describes the analytic software used for the experimental approaches' retrieval of valuable data to be used for further analysis and evaluation in section 5 and section 6. For measurements of real-time capabilities of kernels for the RPi, a description of Cyclictest is giving in section 3.4.1. Hawkeye is a software, sending packets over between endpoints, and the software is described in section 3.4.2. Wireshark is used for control of the packets sent by Hawkeye, and presented in section 3.4.3.

3.4.1 Cyclictest

Cyclictest is a commando line tool commonly used to test Real-Time Operating Systems and measures the kernel's latency. Cyclictest is a scheduling latency benchmarking tool that can help compare different kernel versions. The approach of using Cyclictest has been extended to the industry in the way it monitors the scheduling latency in production environments [23].

Cyclictest starts up a given amount of threads (the user can configure that) that are real-time tasks for the kernel. Each of these threads starts an execution process of a measurement loop that measures the scheduling latency [23]. The threads have timers that start when the thread passes through the loop. The measurement loop then measures the difference between the current time and when the task should have been executed. The thread is then detached, and a new thread enters the loop. Cyclictest repeats this sampling process and can log the data as a CSV file, which then can give output as the one depicted in Figure 3.10. Each sample is one cycle through the measurement loop. Usually, the most critical value is the max, which in this figure is 15.13μs.

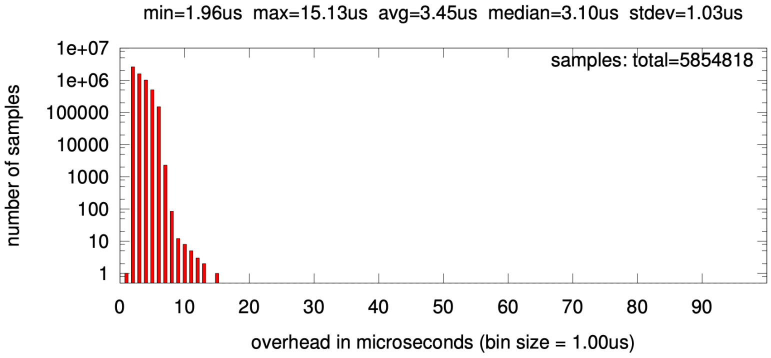


Figure 3.10: A plot of a Cyclictest Result [23].

3.4.2 Hawkeye

Hawkeye is a networking analysis tool from Keysight that wants to offer the possibility of verifying the performance of networks [17].[3] The analytical benefits of Hawkeye, according to Keysight are "quickly and effectively validate network performance. And to isolate problems and proactively detect issues by running scheduled verification tests on any site, by using wireline or wireless connections". Hawkeye simulates application traffic between hardware and software, called performance endpoints. These endpoints represent nodes of a network, and talks with each other through the Hawkeye server, represented in Figure 3.11 for a representation of a Hawkeye Server with associated nodes. The endpoints simplify monitoring, diagnosing, and fixing network issues. This is done by performing tests from the GUI of Hawkeye.

---

[3] This section is inspired by the Hawkeye User Guide [17].

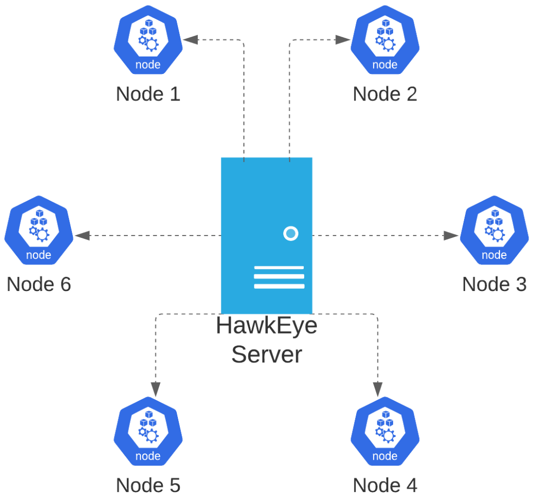


Figure 3.11: A model of a Hawkeye Server with six endpoints represented as Node 1-6.

The Hawkeye Server (HS) is the server for Hawkeye and includes the test engine server and the Registration Server (RS). The RS sets up communication between the Hawkeye Server and the endpoints. The IP address of the Registration Server is initializing the endpoint in the network by configuring the endpoints to connect to the given RS. The RS can usually detect if an endpoint is located behind a firewall and performing address translation and port translation accordingly. The connection for endpoints to the ports of the HS is established with TCP and UDP protocols. The UDP is used for Node to Node and Mesh testing, where tests require synchronization (Network KPI test and any real-time traffic). The ports used for communication to the HS are TCP ports 25025-25050 and 27000-27009 range, with the Registration Server at TCP port 10117. For endpoints, the ports are 10115 and 10116 for TCP, where 10115 is also for UDP traffic. An example of a test scenario is shown in Figure 3.12. Depending on the test, required ports must be open in the firewall. If endpoints have private IP, the access point must open up the ports to the corresponding endpoint, or bidirectional communication is impossible.

Figure 3.12: Test scenario with source and destination endpoint in same Enterprise network, and HS and RS in the Cloud [17].

In the GUI of Hawkeye, one can select Node to Node, Mesh, and Real Services testing to start the test executions. Node to Node allows tests between one source endpoint and one destination endpoint. Mesh is for testing a whole network of endpoints. Real Services allows tests between endpoints and servers located on the internet. Multiple different test scenarios are available in Hawkeye, and more on this can be found in the User Guide [17]. For the tests, one can specify thresholds that the connection must pass for the test to not return a ”failed” message due to threshold exceeding. Otherwise, if the connection passes the test, a ”passed” message is displayed. The Network KPI tests for Node to Node and Mesh testing are suited to understand critical transport metrics such as loss, jitter, and delay. It generates a unidirectional data traffic flow for measuring key transport metrics. The tests send 100 kbps of data, 50 packets/sec, which indicates a negligible impact on the network’s bandwidth.

Hawkeye sends data either with TCP or UDP traffic. The DL/UL throughput is configured in the test tool, and the specifications of the data transmission network limit how large the throughput can be. An overview of the metrics data given by Hawkeye when running Network KPI tests is shown in Table 3.1. These metrics, and many more, are described in the Hawkeye User Guide [17, p. 240-247].

Table 3.1: Overview of Hawkeye metrics from KPI tests.

Network KPI
One way delay (ms)
jitter (ms)
max jitter(ms)
packet loss(%)
voice MOS score
packet loss burst
Max delay variation(ms)

3.4.3 Wireshark

Wireshark is a network protocol analyzer tool that intercepts packets sent over a given channel and is available for Windows, Linux, Solaris, and macOS [15]. An output from a capture of a wireless channel is depicted in Figure 3.13. This output captures; Packet number, time, source address where the packet was sent from, destination address of the packet, Protocol of the packet, length (bits), the time between TCP packets in the same TCP stream, and info about the packet. Wireshark features a shell for investigating packets and a shell for representing the packets as hexadecimal. Wireshark can calculate throughput and latency between packets and illustrate this in a human-readable manner using the I/O graphs option. Wireshark is essential for monitoring and validating that packets are received as intended.

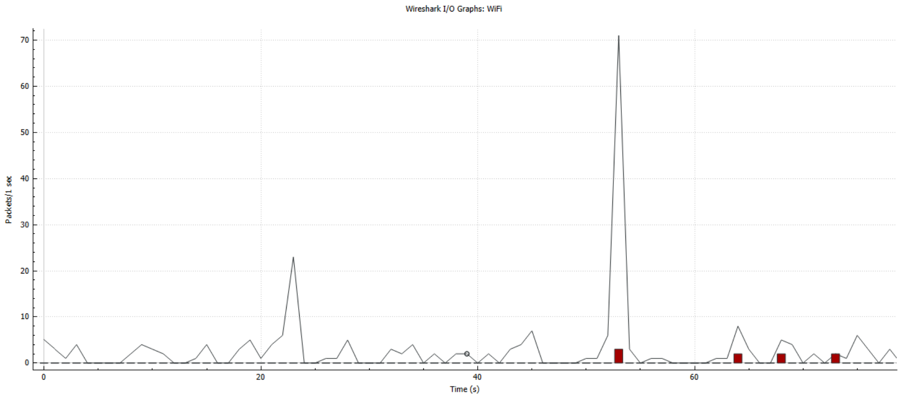


Figure 3.13: Packet capturing output from Wireshark.

4 Experimental Approach

This section describes the essentials for conducting the experimental approach and the procedure of the scientific experiments. The end-to-end approach for the Data Transmission Network (DTN) is illustrated in Figure 4.1 as previously shown in section 3. The RPT operated on the NTNU LAN for the Hawkeye test executions. Nevertheless, the RPT was configured with a 5G HAT for future work. All transmission network scenarios are listed in Table 4.1.[4] The Table is built upon the model in Figure 4.1, but neglects the core networking structures between Denmark and Norway, because this regards inaccessible routers and switches. The procedure of installations, configurations, and controlling of the system setup is presented in section 4.1. The monitoring and test scenarios of the DTN are described in section 4.2. This section also outlines the procedure of the monitoring tools, which captures packets over the DTN, and their reported results are represented in section 5.

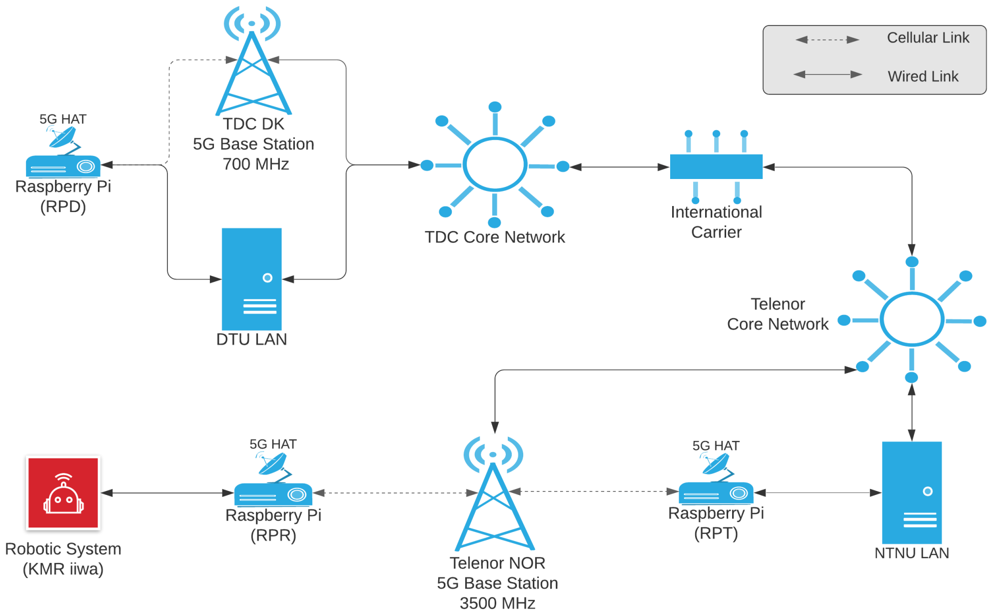


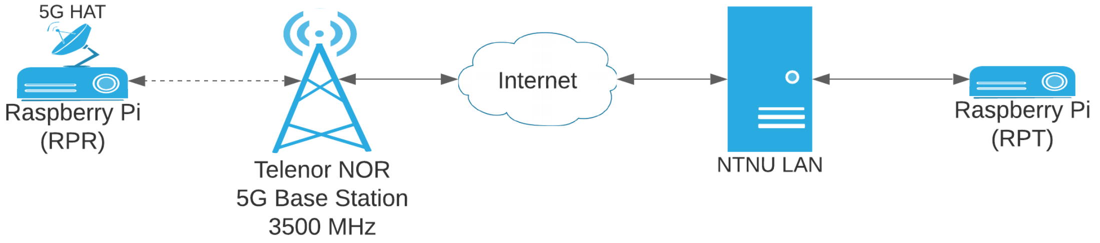
Figure 4.1: The Data Transmission Network used for experimental setup, and visualizes of how the robot can be controlled remotely.

[4] Scenario 4/5 represents two scenarios (the iPhone located at two different locations), see section 4.2.4.

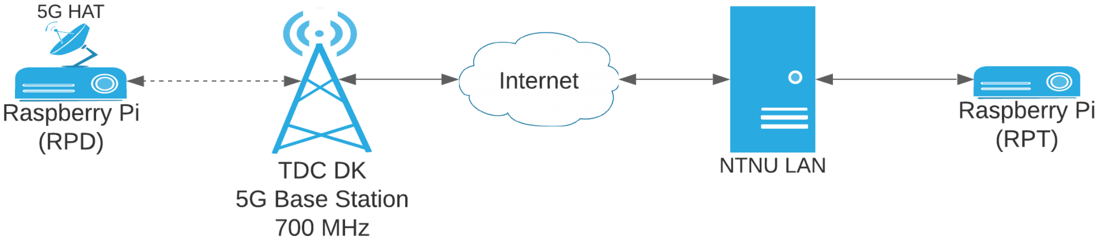
Table 4.1: Different transmission networks used for data traffic for the test scenarios. The network setups show end-to-end bidirectional communication, without the core network structures.

Telecommunication Test Setup Scenarios

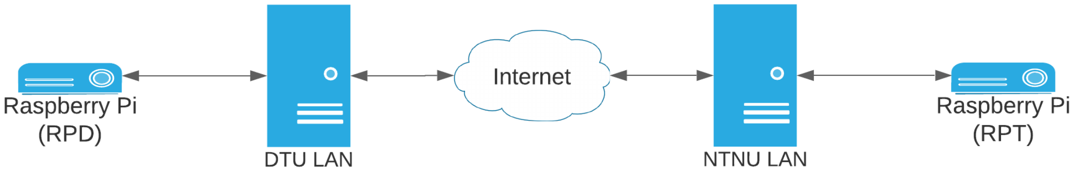
1:



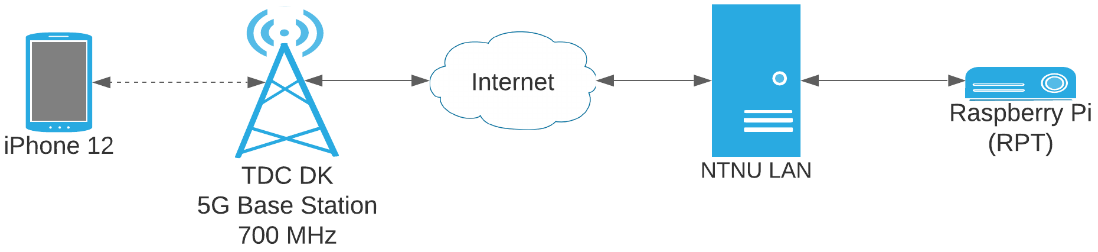
2:



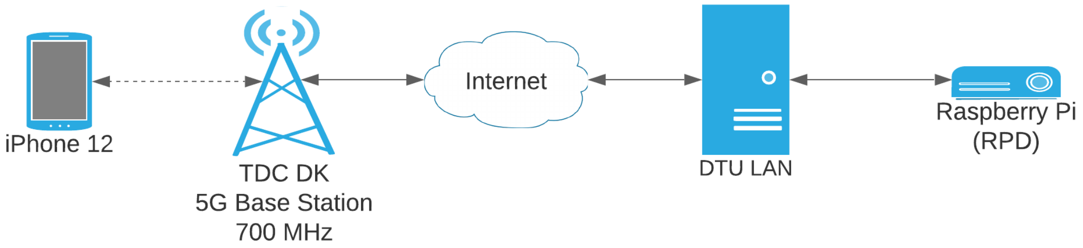
3:



4/5:



6:



4.1 Setup and Preparations

The experimental setup is comprehensively explained in this section. The goal is to give a thorough presentation, such that this project can be repeated with different test scenarios or use cases to compare performance or networking development. How the different hardware in the network is configured and located is described in section 4.1.1. The network of endpoints and its connectivity to the Hawkeye Server is explained in section 4.1.2.

4.1.1 Hardware Setup

The experimental setup consists of three Raspberry Pi's, one iPhone 12, a robotic system (KMR iiwa), and multiple connection interfaces. The Raspberry Pi's are located at three different locations. One in Denmark, one in Trondheim at NTNU, and one connected to the robotic system in a lab also at NTNU. These Raspberry Pi's are hereby referred to as RPD (Raspberry Pi Denmark), RPT (Raspberry Pi Trondheim), and RPR (Raspberry Pi Robot), respectively. Given the communication architecture shown in Figure 4.1, the robotic system can be controlled with different scenarios. [5]

RPR and Robotic Setup In order to command the robotic system as intended, a communication link must be provided for data transmissions. The main element of the system is the RPR, sending ROS2 commands through an Ethernet connection to the KMR iiwa. The wireless connection is made through a SIM8200EA-M2 5G HAT connected to a 5G Base Station from Telenor at NTNU. An Ethernet connection between the KMR iiwa's KUKA Control PC (KPC) and the RPR establishes the connection for ROS2 commands. An illustration of the RPR's communication links is depicted in Figure 4.2. The RPR and the Robot are located in an industrial lab. In this lab, wielding and manipulation of large objects can occur and therefore interfere with the transmissions.

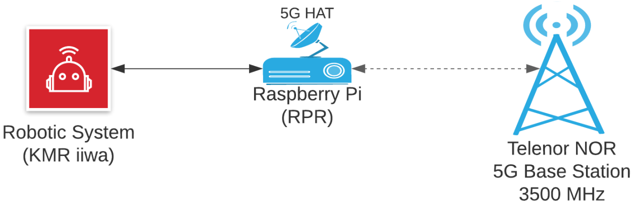


Figure 4.2: An illustration of the RPR's communication links.

A guide for setting up the RPR with regards to this laboratory setup is described in Appendix A. The choice of kernel derives from the results of the Cyclictests, shown in section 5.1. The robotic system is similar to the one used in [32], and the only difference is the cellular communication coming from the 5G HAT. The robotic setup and communication protocols between the robot and its equipment is represented in Figure 4.3 and built upon the structure in [7].

---

[5] Parts of the paragraphs regarding the RPR setup is gathered from the author's report "Remote operations and testing of KMRiiwa using Raspberry Pi 4 with Wi-Fi".

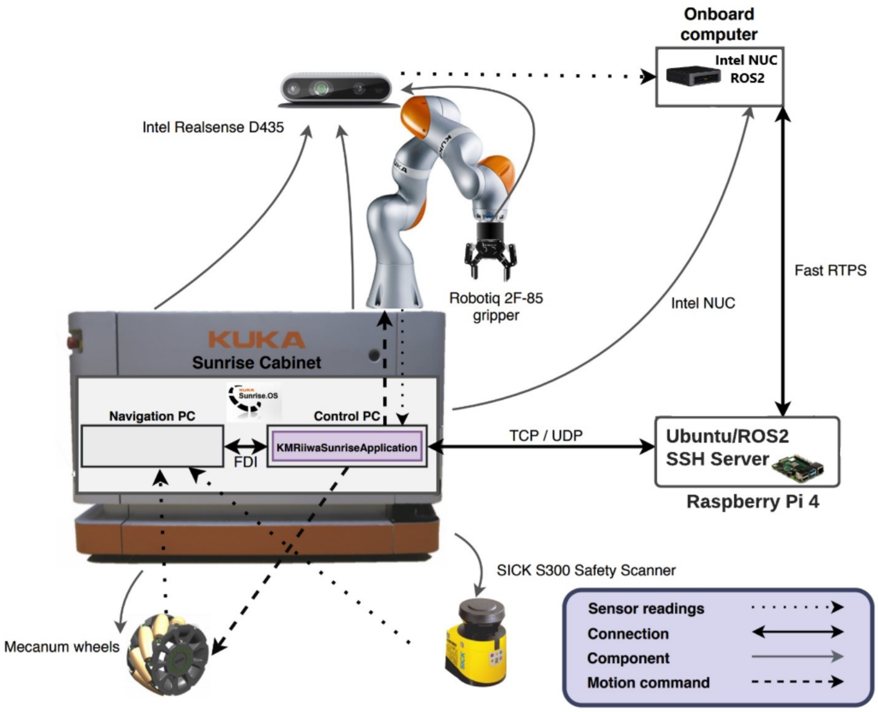


Figure 4.3: Robotic setup with components, connections and applications.

The connection between ROS2 on the RPR and KMRiiwaSunriseApplication running on the KCP is established via the KLI port on the robot. A TCP or UDP connection for the transport layer can either be established, and this is created with programmable socket nodes defining the IP addresses and ports, created in ROS2 with corresponding nodes in the Sunrise application. ROS2 nodes publish and subscribe to the corresponding Java nodes over the TCP/UDP connection. This is represented in Figure 4.4. This way, the RPR can control the robot by sending ROS2 commands over TCP/UDP, either moving the robot or receiving sensor readings. This is done by calling the seven nodes; kmp_odometry, kmp_laser, kmp_status, kmp_command, lbr_sensordata, lbr_status, and lbr_command in ROS2. This means that TCP packets are being sent back and forth between ROS2 and the KPC when the system is running.

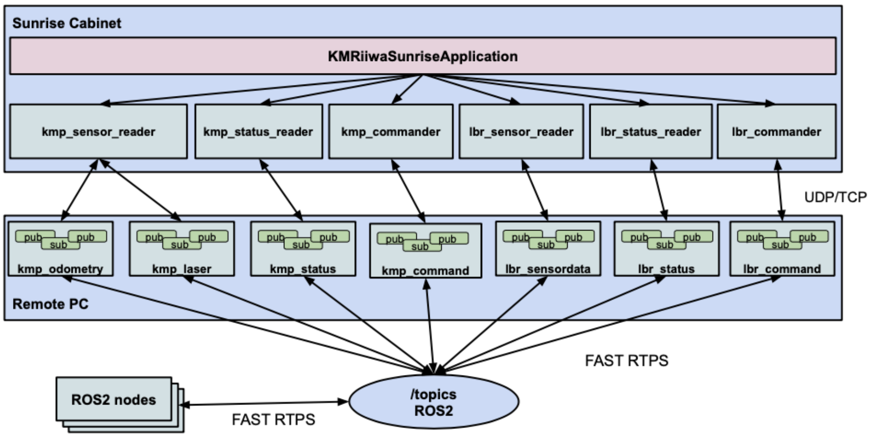


Figure 4.4: Overall architecture of the communication between ROS2 and KMRiiwaSunriseApplication [7].

RPT Setup For local utilization of the 5G network from a base station at NTNU, a Raspberry Pi is set up for communication, both between DTU and NTNU, and also to be able to operate the robotic system remotely. The RPT is set up accordingly to Appendix A, but it is given a public IP for the Ethernet port on the NTNU LAN for enabling Hawkeye endpoints to set up communication with RPT. This gives the RPT networking interfaces of both Ethernet and cellular, meaning the system's connectivity is diverse and promotes test scenario configurations. An illustration of the RPT with its coherent connection link is displayed in Figure 4.5. The RPT is located in an office at NTNU $< 100m$ away from RPR. It is connected to the same 5G NR frequency band as the RPR.[6]

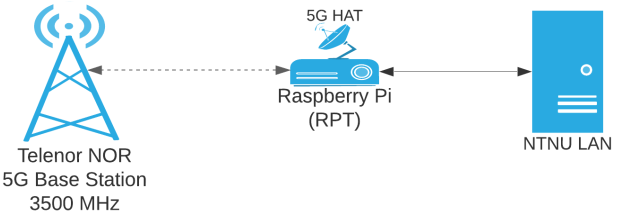


Figure 4.5: An illustration of the RPT's connection links.

RPD Setup Transmissions from NTNU in Trondheim propagates through the core networks to DTU in Lyngby, north of Copenhagen. The receiving Raspberry Pi, the RPD, is set up similarly to RPT, given in Appendix A, with a public IP on the DTU LAN. The RPD can use the TDC 700 MHz 5G network, or the Ethernet connection to DTU LAN. This is represented by Figure 4.6.

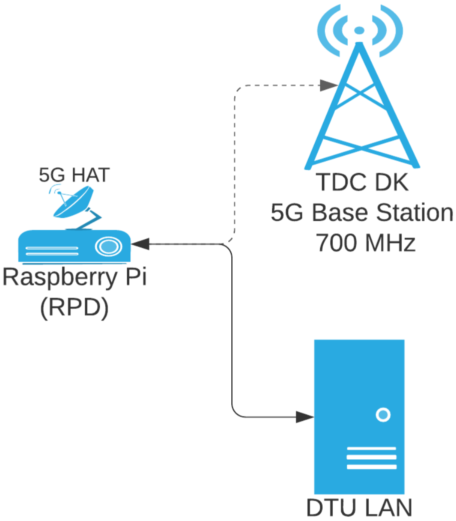


Figure 4.6: An illustration of the RPD's connection links.

iPhone 12 Setup The versatility of a handheld device, which requires no cables or physically connected interfaces, is advantageous for frequency and RSSI evaluation. The iPhone 12 is running iOS 14.5 OS and is connected to the TDCs 5G network similarly to RPD; thus, the iPhone has the mobility to move to better locations for better signal. The iPhone can connect to DTU's WLAN, Eduroam. The connection links is depicted in Figure 4.7.

---

[6] This connection is not used for this thesis but is set up for future work.

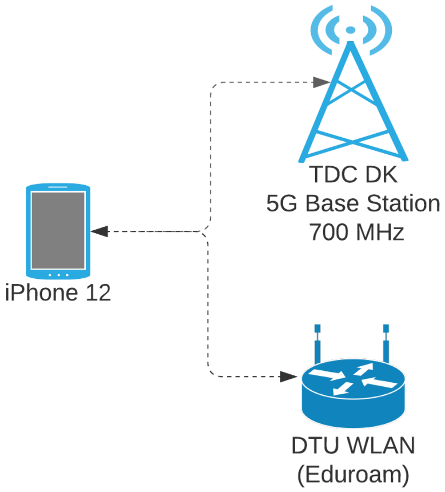


Figure 4.7: An illustration of the iPhone 12's connection links.

4.1.2 Hawkeye Setup

To utilize Hawkeye, the Hawkeye Server must be set up. This server is where all endpoints are connected to and manages test schedules, results, and all the UIs available on the server's web page. Four nodes are connected to this server; RPR, RPT, RPD, and iPhone 12, represented in Figure 4.8. For registration of the nodes on the Hawkeye server, an installation guide for setting up endpoints for Raspberry Pi's and iPhone 12 is given in Appendix B.

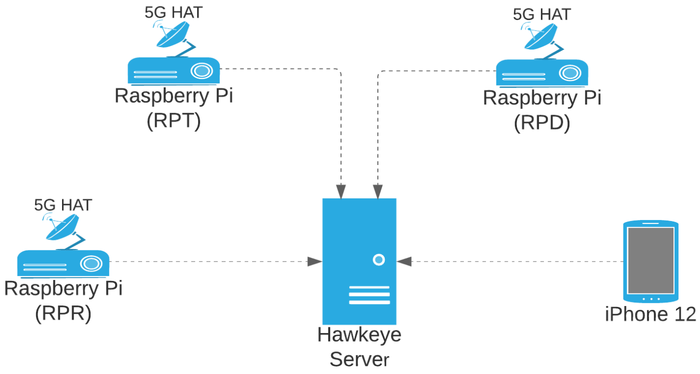


Figure 4.8: An overview of the Hawkeye server, with the connected endpoints.

When Endpoints is properly registered on the Hawkeye Server (HS), the Endpoints is kept active on the HS by sending "keep-alive" messages every 2-4 seconds to the server on TCP port 10117. For each message, the server updates the Endpoint's IP automatically if it changes [17]. Hawkeye uses UDP packets to synchronize all Endpoints to the same clock, and according to the user guide, it can achieve anything from a couple of milliseconds to tens of milliseconds synchronization accuracy. The PMUI displays *Public/Private* probes, which indicate which probes have a public IP address and which have a private IP address. As long as both probes are not private, tests can be generated in the Test Execution User Interface (TEUI). Both 5G networks used for testing are commercial Non-Standalone (NSA) networks. Because ISPs are not interested in handing out private IPs on a commercial cellular network, a public IP address is set on the NTNU LAN on the

RPT for Hawkeye communication, *RPD <->RPT, RPT <->RPR and iPhone 12 <->RPT*. This puts RPT in a central role for the Hawkeye test executions, which means that all tests are *to* or *from* the RPT via the NTNU LAN.

During test executions, the Wireshark software monitors the interface that is sending or receiving packets. When the packets received or sent, correlates with the packets that should have been received or sent for that specified test, the data transmission is validated.

4.2 Test Scenarios

The primary purpose of the test scenarios is to utilize Hawkeye to test the capabilities of the Data Transmission Network (DTN) shown in Figure 4.1. To evaluate this network and its QoS, different Key Performance Index (KPI) is used. These KPIs are explained in section 4.2.1.

As discussed in section 2.2 multiple parameters introduce delays in the system, and the kernel could be an influencing factor. For minimizing the latency of the kernel of the Raspberry Pis, the RT patch was applied as discussed in Appendix A. The latency impact of this RT patch, measured with Cyclictest as explained in section 3.4.1, is given in section 5.

Section 4.2.2 deal with test scenarios conducted between the RPR and RPT. Similarly, section 4.2.3 and section 4.2.4 describe the conducted test cases between RPT and RPD, and RPT and iPhone 12, respectively. The different transmission networks for the test scenarios are also shown in Table 4.1.

4.2.1 Key Performance Indices

Network latency is essential for remote operations because a time-critical command to the robot must reach the robot on time. In this system setup, DTU and NTNU are more than 800 kilometers apart. Thus latency can be a factor that influences the performance of the system. The KMR iiwa should be able to transmit sensor data and video streams to DTU. Therefore, it is beneficial to measure how much available throughput the robot can achieve, uplink and downlink. Also, when a system is sending or receiving a stream of data, it behaves differently, and latency can occur if the endpoint cannot receive or send the given streaming service. The KPIs represent the aims of the system testing scenarios and are given in Table 4.2. These KPIs are used for evaluating *each* communication link.

Table 4.2: KPI's for evaluating communication links.

KPI	**Unit**	**Description**
One Way Delay (OWD)	ms	The time it takes for one packet to transmit from source to destination, similar to propagation delay, but with regards to only the network itself.
Throughput Downlink	Mbps	The amount of data available to pass through from *destination* to *source* over time.
Throughput Uplink	Mbps	The amount of data available to pass through from *source* to *destination* over time.
Reliability	%	The chance for successful packet transmission, without corruption, from source to destination.

The test scenarios in this project use the Network KPI test mentioned in section 3.4.2 with metrics in Table 3.1. In addition to this test, a Hawkeye Skype4B test is executed. This test provides a unidirectional stream of audio and video, where the two streaming services' throughput can be configured in the test execution user interface. This test uses the Real-time Transport Protocol (RTP) for transmission, which is built upon the UDP protocol. For a remote control operation, this test can simulate a streaming service between the robot and the operator and then evaluate the reliability delay of the stream. The difference between the two is that the KPI test has a low impact on the network, while the Skype4B can be configured to require a given throughput between the endpoints to deliver the streaming service. To evaluate the maximum available throughput between two endpoints, a Hawkeye Speedtest is used. This test measures the available throughput between the two endpoints over time.

4.2.2 RPR - RPT

This section focuses on the transmission line between the RPR and the RPT. In a remote control operation of the robot, commands can be transmitted from the RPT to the RPR, and then given to the Robot via ROS2 and the Ethernet cable, see Figure 4.1 for communication links.[7] The packets can be transmitted bidirectionally from the RPT to the NTNU LAN, to the cellular base station, and then to the RPR, see Figure 4.9.

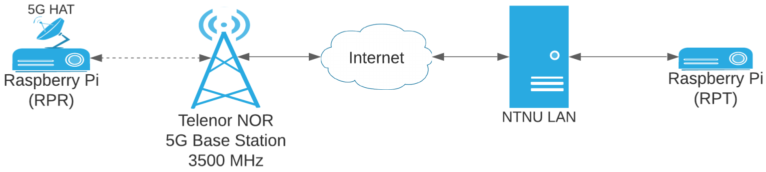


Figure 4.9: The bidirectional test setup between the RPR and RPT.

The Hawkeye Tests used for monitoring this transmission network are Network KPI, the Skype4B, and the Speedtest. Each KPI and Skype test runs 3x10 minutes, where one test takes ten minutes and is executed three times over a time period. The speedtests runs related to these tests to check the available throughput. The Skype4B test is executed with a 10 Mbps stream, resulting in 6 Gb of data being transmitted per test. The Speedtest is executed with seven TCP streams, the same number as TCP connections between ROS2 and the sunrise application on the KMR iiwa. All test results are gathered from the receiving endpoint in the test.

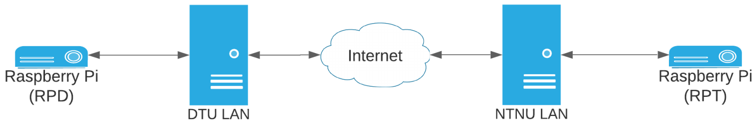
4.2.3 RPD - RPT

This section is about the transmission network between NTNU and DTU. From a control perspective, the RPT is a communication hub for the incoming and outgoing traffic from and to DTU. RPT receives data traffic from or sends it to RPR, which communicates with the KMR iiwa. Both the RPT and RPD can use LAN as the first step of communication, previously shown in Figure 4.5 and Figure 4.6 respectively. However, the RPD can also communicate via TDC's 5G commercial network. Figure 4.10a and Figure 4.10b shows these two test links for LAN and 5G communication. The LAN is monitored for comparison to the 5G network, which composes two test scenarios, LAN to LAN and 5G to LAN.

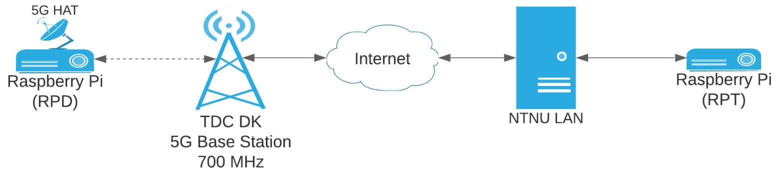
This transmission network is a simulated extension of the RPR-RPT setup. Thus the scenarios are the same for testing the 5G TDC network and the LAN. The Hawkeye tests are conducted similarly to RPR-RPD for both communication links, as presented in section 4.2.2.

---

[7] Results from the RPR-RPT setup are also considered for comparing to the findings in the author's previous work in [32].



(a) LAN Link



(b) 5G Link

Figure 4.10: The two bidirectional test scenarios between RPD and RPT.

4.2.4 iPhone 12 - RPT

This communication setup is configured to see any difference between the iPhone and the RPD in 5G performance. The iPhone and the 5G HAT both use the same Qualcomm chip as a 5G modem, but other hardware can still impact the performance. The link is shown in Figure 4.11. The iPhone is a replacement for the RPD in the network but is a portable device. This link is tested in two locations for evaluating different coverage areas of the 5G commercial network. The first location being next to the RPD, and one outside, which should be, according to TDC, a better coverage area. This will result in two test scenarios. The Hawkeye tests for both locations are executed as described in section 4.2.3.

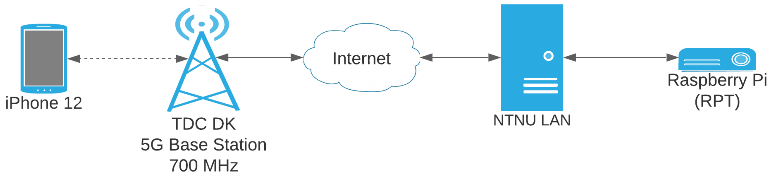


Figure 4.11: The bidirectional test setup between the iPhone 12 and RPT.

4.2.5 iPhone 12 - RPD

This scenario is about testing the TDC 700 MHz network at DTU, similar to the Telenor 3500 MHz network tests in Trondheim. Both devices are located in an office such as the system in Trondheim, and the RPD will be representing the RPT, and the iPhone 12 will represent the RPR, like the scenario discussed in Figure 4.9. An illustration of the test communication link is presented in Figure 4.12.

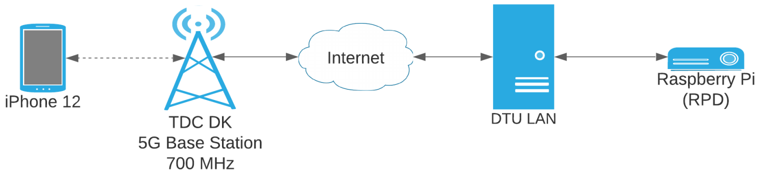


Figure 4.12: The bidirectional test setup between the iPhone 12 and RPD.

4.2.6 Test Scenarios

All the endpoints and transmission networks results in the test scenarios listed in Table 4.3.

Table 4.3: The telecommunication test scenarios.

Nr.	**Label**	**Test Link Description**
1	*RPR 5G - RPT LAN*	RPR is connected to Telenor's 3500 MHz 5G at NTNU and RPT connected to NTNU LAN.
2	*RPD 5G - RPT LAN*	RPD is connected to TDC's 700 MHz network, and RPT connected to NTNU LAN.
3	*RPD LAN - RPT LAN*	RPD is connected to DTU LAN and RPT connected to NTNU LAN.
4	*1st: i12 5G - RPT LAN*	iPhone 12 located at DTU office, connected to TDC's 700 MHz 5G and RPT connected to NTNU LAN.
5	*2nd: i12 5G - RPT LAN*	iPhone 12 located outdoor for better signal on the TDC 700 MHz 5G network. RPT connected to the NTNU LAN.
6	*i12 5G - RPD LAN*	iPhone 12 located at DTU office, connected to TDC's 700 MHz 5G network, and RPD connected to DTU LAN.

Results and Evaluation

PART 3

5 Experimental Findings

This section presents and explains the essential findings for this thesis. First, a presentation of the results from the Cyclictests, and comparison of the different kernels in section 5.1. The following sections are structured such that each section represents the given data transmission link. Three different Hawkeye tests were used for measuring mainly the KPIs latency, throughput, and reliability of the transmission networks. These three tests, Network KPI, Skype4B and Speedtest, is hereby referred to as N-KPI, S4B and ST respectively. Each test were conducted over three different time periods, represented as T1, T2 and T3 for each test type. Each test was evaluated both for Uplink (UL) and Downlink (DL).

The experimental findings from the test scenarios are presented, explained and discussed individually for each section. This is to provide readability and to make a clearer overview of each scenario. Furthermore, all the test scenarios' findings and the significance of the findings, will be discussed as a whole in section 6.

Findings from tests between RPR and RPT is presented in section 5.2. Results from tests between the RPD and RPT, using 5G to LAN are presented in section 5.3, and using LAN to LAN in section 5.4. The results from the iPhone 12 on 5G to RPT on LAN, with regards to the two test scenarios, are presented in section 5.5 and section 5.6. The sixth test scenario, between the iPhone 12 and RPD is given in section 5.7.

5.1 Cyclictest

This sections presents the Cyclictest findings. These findings is based on three different kernels. Three preemptive kernels were tested, two real-time and one non real-time, and the overhead latency of the four cores of the Raspberry Pi 4B were measured. The Cyclictests ran for 200 million cycles. The three versions of the kernel were 64 bit Linux for Raspberry Pi OS, with patch versions 4.19.59-rt24, 5.4.93-rt51 and 5.10.10, and the results from the three tests is given in Figure 5.1, Figure 5.3 and Figure 5.4 respectively. The tests are shown in a logarithmic y-axis, because the small samples are hard to see when the y-axis is not logarithmic, see Figure 5.2 for the 4.19.59-rt24 results without logarithmic y-axis.

The maximum overhead latency and for the 4.19 real-time patch was measured to be 72 μs. The results show that core4 measured higher latency, as the overhead profile in Figure 5.1 indicates. The maximum overhead latency of the 5.4 real-time patch was measured to be 46 μs. It also has a more left-shifted gathering of the latency measurements in Figure 5.3, compared to the 4.19 patch. The 5.10 patch was not a real-time patch, but an upgrade from a 5.4 preemptive patch. This means that it did have a preemptive setting, but not a fully preemptive patch indicating that the OS is has real-time qualities. The maximum overhead latency of 5.10 was measured to be 91 μs. Both core 1 and 2 presented a higher average of overhead latency, compared to the two real-time patches.

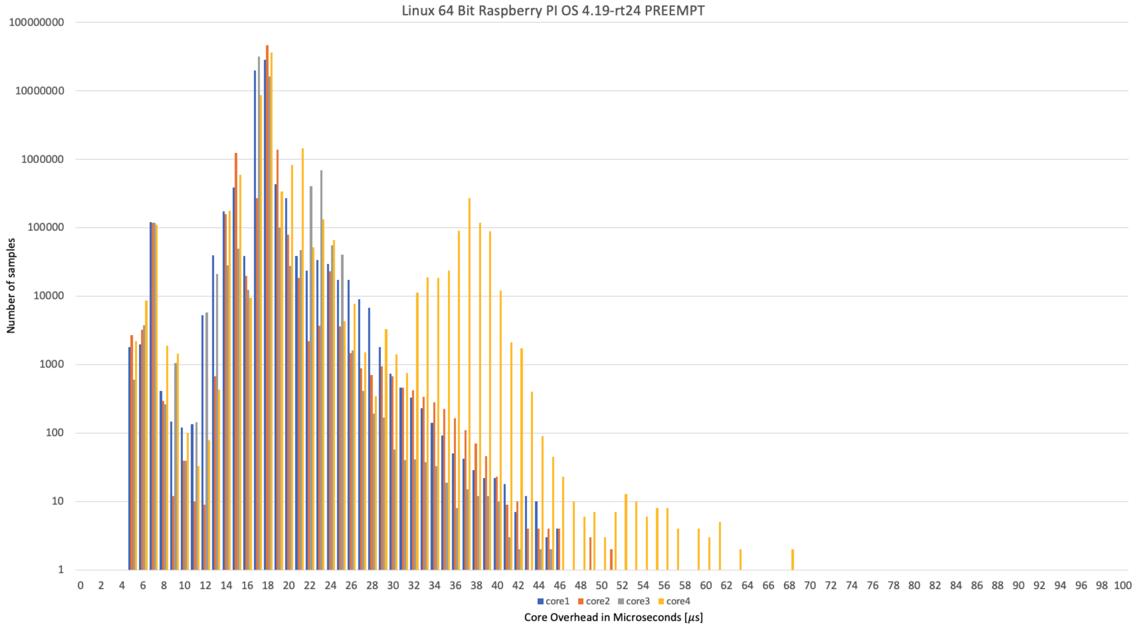


Figure 5.1: Core overhead from 200 million cycles in a Cyclictest for Raspberry Pi OS 4.19-rt24 PREEMPT.

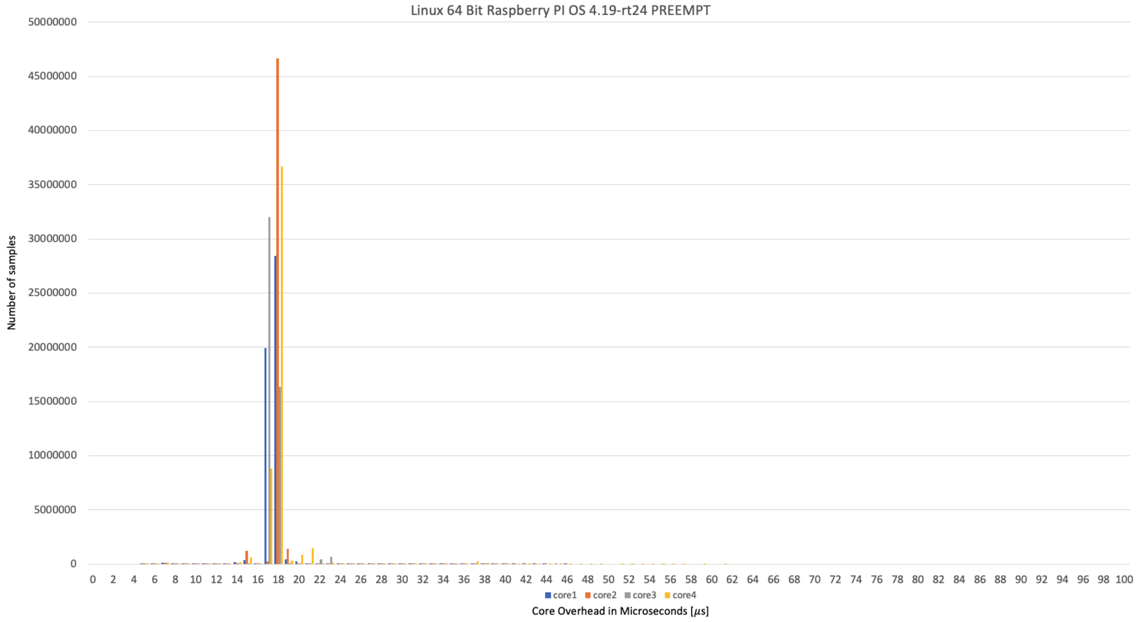


Figure 5.2: Core overhead from 200 million cycles in a Cyclictest for Raspberry Pi OS 4.19-rt24 PREEMPT. This is without a logarithmic y-axis.

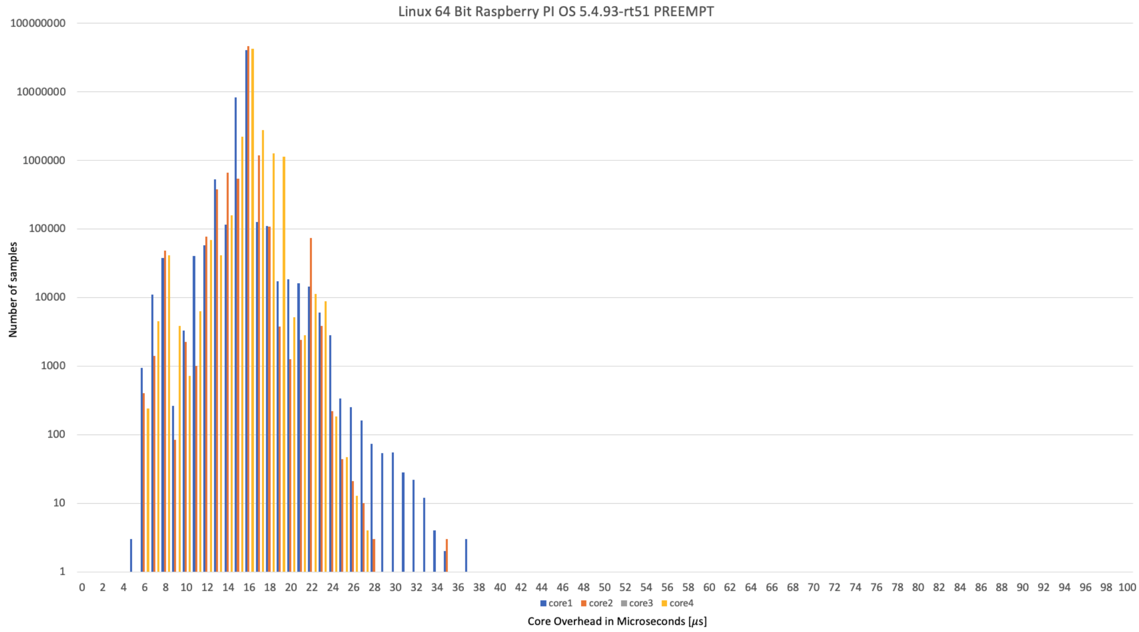


Figure 5.3: Core overhead from 200 million cycles in a Cyclictest for Raspberry Pi OS 5.4.93-rt51 PREEMPT.

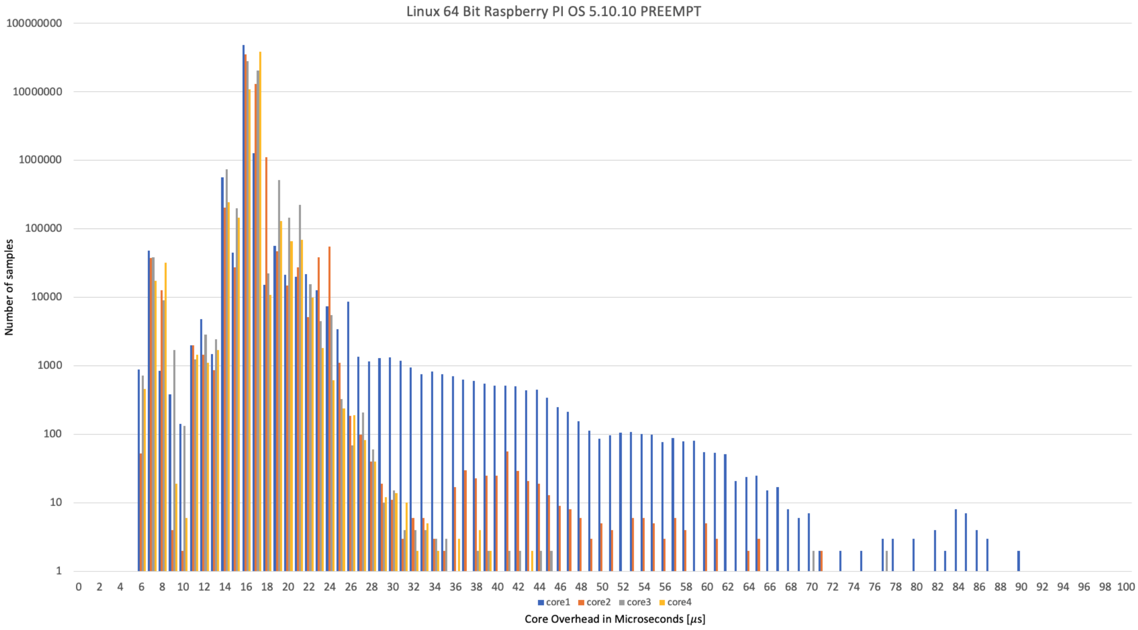


Figure 5.4: Core overhead from 200 million cycles in a Cyclictest for Raspberry Pi OS 5.10.10 PREEMPT.

5.2 Scenario 1: RPR - RPT

The findings were acquired from monitoring the transmission network between the Raspberry Pi in the lab connected to the robot and the Raspberry Pi at the NTNU office. This scenario is represented in section 4 Table 4.1, number 1. Sections present the latency and the other essential observations that were made during these tests. Section 5.2.1, section 5.2.2 and section 5.2.3 presents findings from the N-KPI, S4B and ST tests respectively. Section 5.2.4 summarizes the test scenario by presenting the resulting KPIs. Section 5.2.5 discusses and evaluates the results of these findings.

5.2.1 N-KPI Test

The latency findings for the three N-KPI tests for this scenario are given in Figure 5.5 for T1, Figure 5.6 for T2 and Figure 5.7 for T3. These figures show the One Way Delay (OWD) for UL (RPR to RPT) and DL (RPT to RPR). For all tests for this scenario, RPR is represented as the "from" probe, and RPT is represented as the "to" probe. Essential latency metrics from these three tests are represented in Table 5.1. This Table shows maximum (Max) and average (Avg) latency for the UL, DL, and Round-Trip Time (RTT).

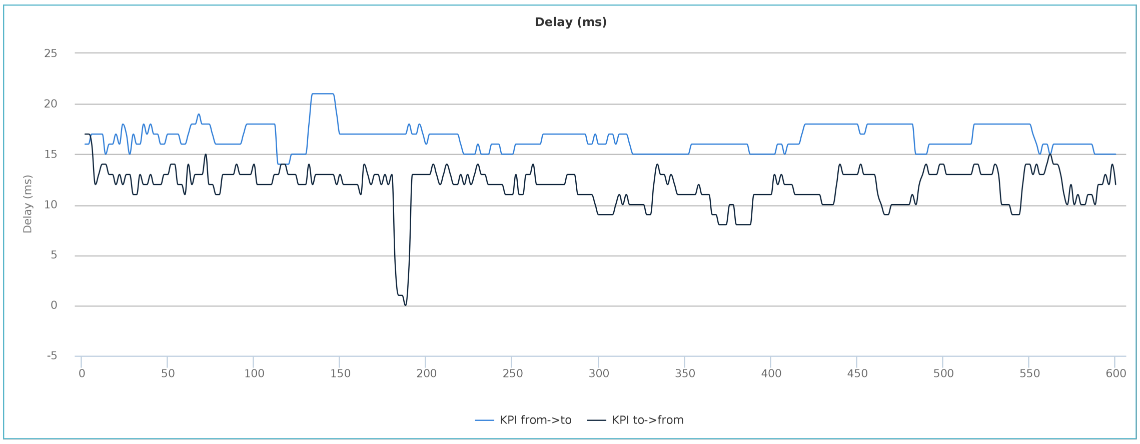


Figure 5.5: Latency findings for 10 minutes of the N-KPI T1 between the RPR and RPT. Blue line represents UL and black line is the DL.

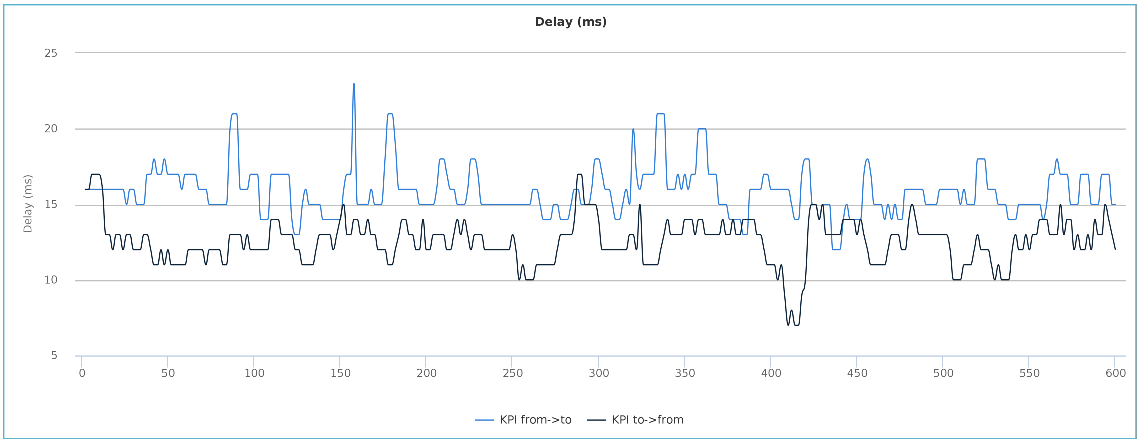


Figure 5.6: Latency findings for 10 minutes of the N-KPI T2 between the RPR and RPT. Blue line represents UL and black line is the DL.

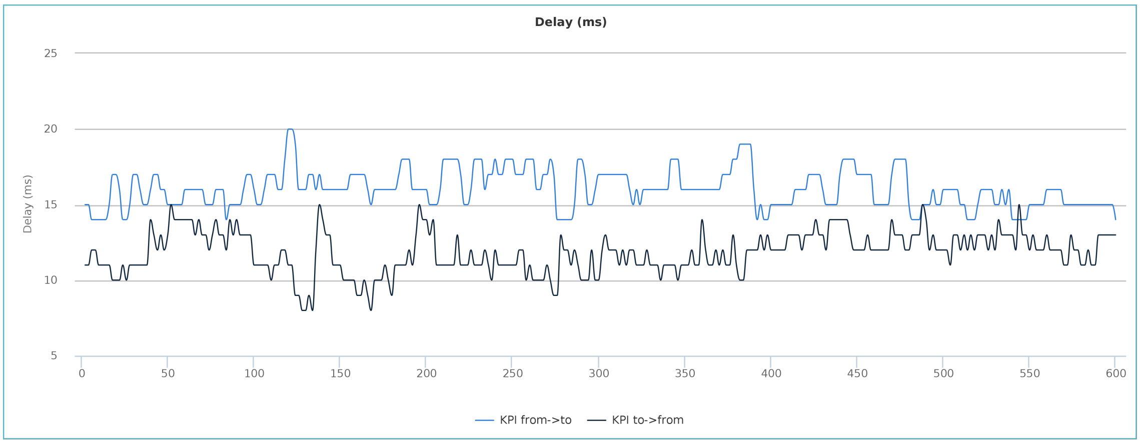


Figure 5.7: Latency findings for 10 minutes of the N-KPI T3 between the RPR and RPT. Blue line represents UL and black line is the DL.

Table 5.1: N-KPI latency observations from the three tests.

N-KPI Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	21	16.61	17	11.82	38	28.43
T2	23	15.85	17	12.54	40	28.39
T3	20	16.04	15	11.82	35	27.86

Packet Loss during N-KPI
No packet loss was observed for the three N-KPI tests between the RPR and the RPT.

5.2.2 S4B Test

The S4B tests represented a streaming scenario for which each endpoint sent a stream of 10 Mbps to the other. The streaming services from S4B T1 are represented in Figure 5.8.

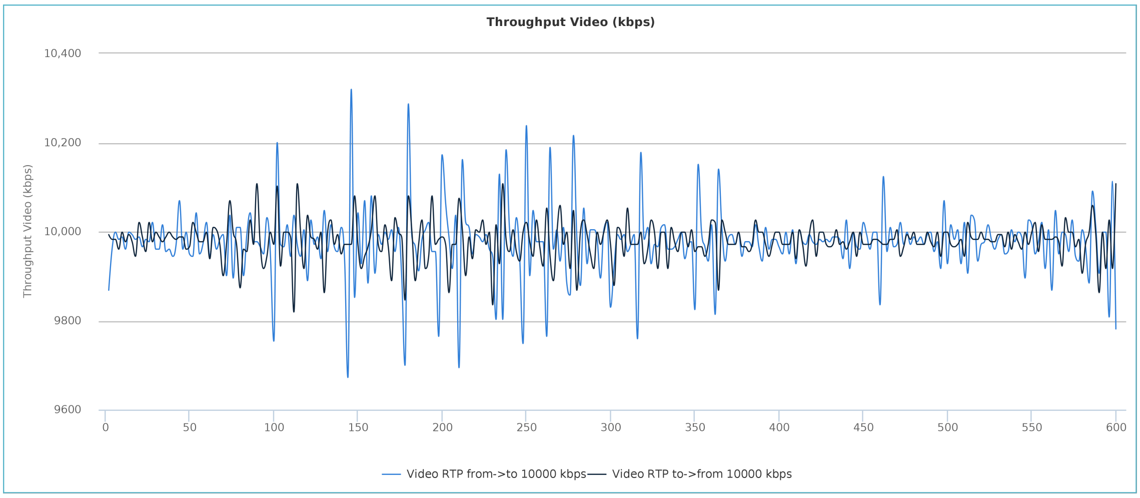


Figure 5.8: The bidirectional streaming service between RPD and RPT.

The OWD of S4B T1, T2 and T3 are presented in Figure 5.9, Figure 5.10 and Figure 5.11 respectively. Essential latency metrics from these three tests are represented in Table 5.2. The packet loss observed during S4B T1 is shown in Figure 5.12.

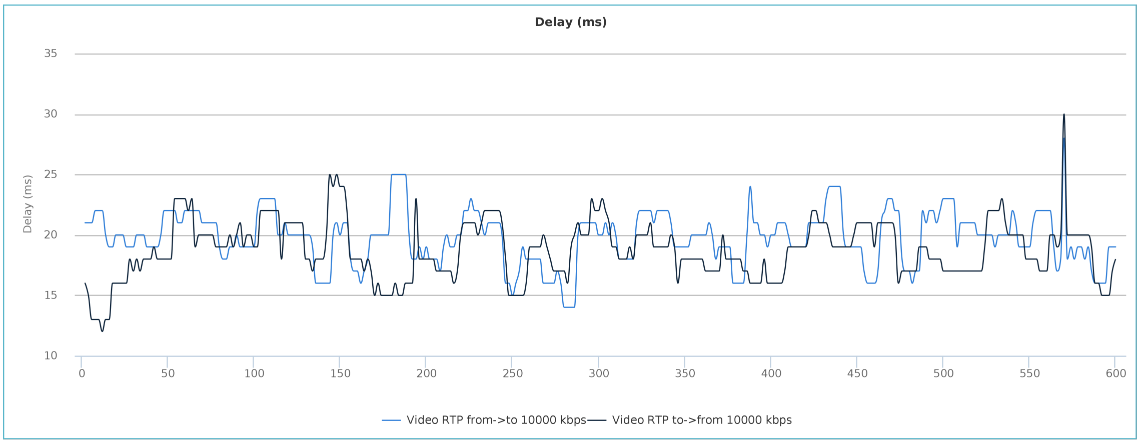


Figure 5.9: Latency findings for 10 minutes of the S4B T1 between the RPR and RPT. Blue line represents UL and black line is the DL.

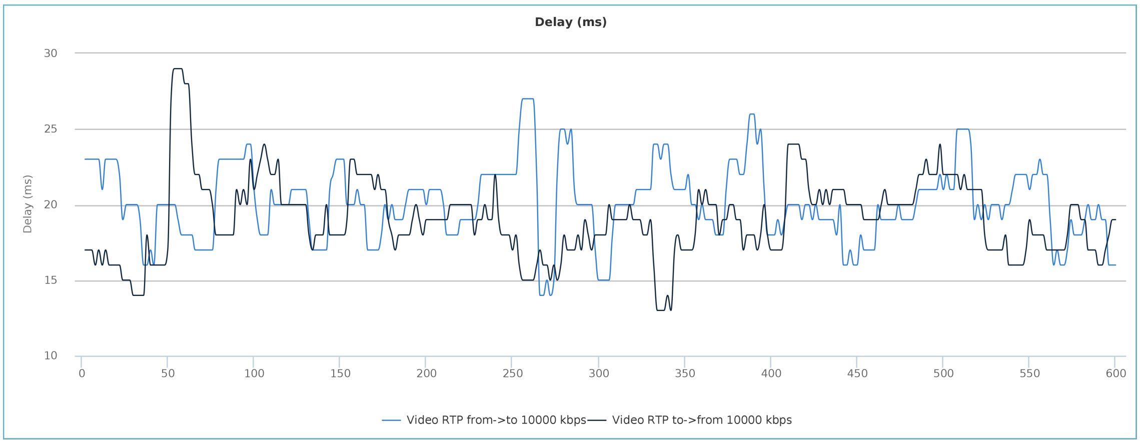


Figure 5.10: Latency findings for 10 minutes of the S4B T2 between the RPR and RPT. Blue line represents UL and black line is the DL.

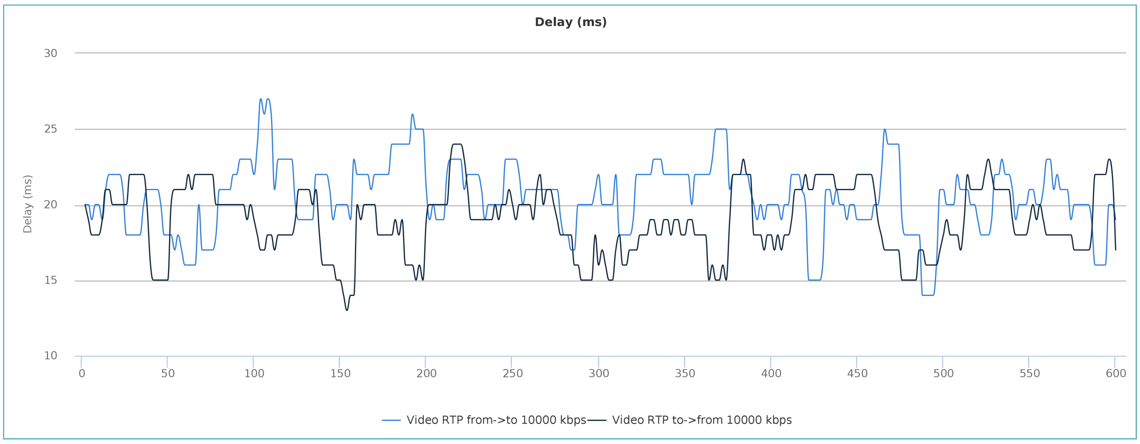


Figure 5.11: Latency findings for 10 minutes of the S4B T3 between the RPR and RPT. Blue line represents UL and black line is the DL.

Table 5.2: S4B latency observations from the three tests.

S4B Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	28	19.78	30	18.73	58	38.51
T2	27	20.11	29	19.01	49.11	20.08
T3	27	20.57	24	18.91	51	39.48

Packet Loss during S4B

The packet loss observed during S4B T1 is shown in Figure 5.12. The figure shows two spikes at the 570-second mark, one for UL (blue line) and one for DL (black line). The blue spikes at 1.73% packet loss and black spikes at 5.84% packet loss. No loss was observed for S4B, T2, and T3. This is summarized in Table 5.3.

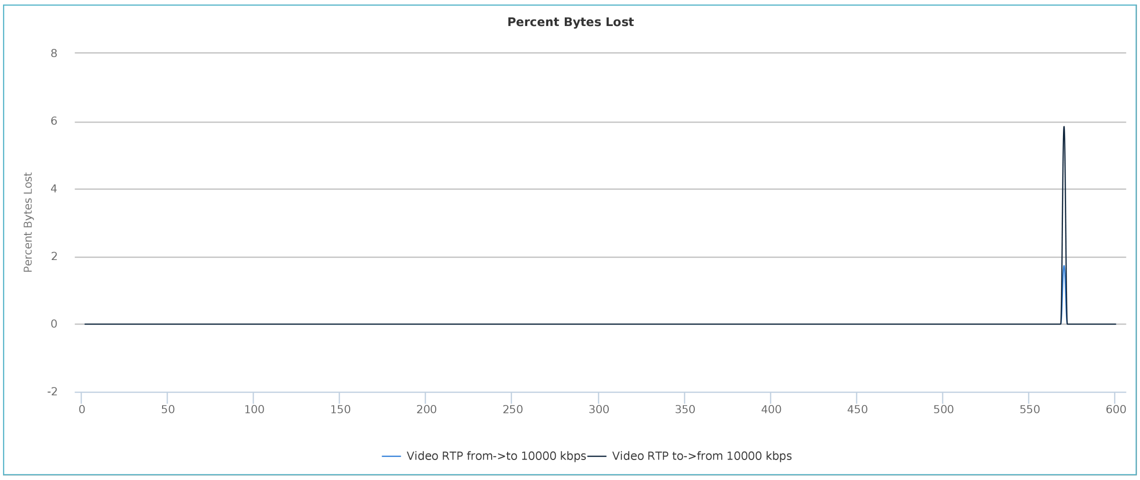


Figure 5.12: Packet loss in percentage during S4B T1.

Table 5.3: Percentage packet loss from three N-KPI tests.

Packet Loss S4B		
Test	UL (%)	DL (%)
T1	0.01	0.02
T2	0	0
T3	0	0

5.2.3 ST test

Findings of the three STs between RPR and RPT are presented in Table 5.4. This Table shows the Average throughput available between the two endpoints over three different time periods.

Table 5.4: Average throughput from three STs.

Throughput ST		
Test	UL (kbps)	DL (kbps)
T1	79165	90755
T2	78322	89856
T3	70383	89800

5.2.4 KPI Findings

The communication link between RPR and RPT was monitored, and measurements resulted in the KPIs presented in Table 5.5.

During the three (T1, T2, T3) Network KPI (N-KPI) tests, the one-way-delay (OWD) averaged at 16.17 ms for the Uplink (UL) and 12.06 ms for Downlink (DL). There was observed a drop in latency for DL for the first test, seen in Figure 5.5. This drop was from about 12 ms and to around 0 ms. The maximum OWD measured for all three tests was 23 ms for UL and 17 ms for DL. The average round-trip time (RTT) was 28.23 ms, and the maximum RTT was 40 ms. No packet loss was observed for these N-KPI tests.

The Skype4B test showed an increase in average RTT of 38% compared to the N-KPI test. For the most part, the contribution to this increase comes from a rise in the DL OWD, which increases by around 70% for the S4B test. During the S4B T1, there was a peak at around the 570-second mark, which resulted in a packet loss of almost 6% for DL and almost 2% for UL. This resulted in the total reliability of all tests to be 99.997% for UL and 99.993% for DL.

The Speedtests showed an average throughput of 75.95 Mbps for UL and 90.14 Mbps for DL. The UL speed is expected, but the DL is slower than what the 5G network at NTNU should achieve. Regardless, this speedtest is between the RPR and RPT, meaning the connection also goes through the NTNU LAN, indicating that multiple reasons can cause this.

Table 5.5: Summary of the KPI's collected for the RPR-RPT test scenario.

KPIs										
	OWD N-KPI (ms)		**OWD S4B (ms)**		**Throughput ST (kbps)**		**Reliability N-KPI (%)**		**Reliability S4B (%)**	
Test	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL
T1	16.61	11.82	19.78	18.73	79165	90755	100	100	99.99	99.98
T2	15.85	12.54	20.11	19.01	78322	89856	100	100	100	100
T3	16.04	11.82	20.57	18.91	70383	89800	100	100	100	100
Avg	**16.17**	**12.06**	**20.15**	**18.88**	**75957**	**90137**	**100**	**100**	**99.997**	**99.993**

5.2.5 Discussion

The findings did not correspond to what was achieved in the paper "Utilizing 5G in Industrial Robotic Applications", described in section 1.4, where it was seen a 5 ms for UL and DL for an LTE connection and 0.5 ms for 5G. The OWD from this thesis was about three times higher for UL and two times higher for DL than LTE for the N-KPI test. Nevertheless, in this paper, multiple cells were used in the simulation, which was located 20m apart, with 5-10 robots connected to each cell with throughput for DL/UL <100 bytes.

The RPR - RPT scenario was meant as a replacement for the Wi-Fi setup discussed in the specialization project (SP) [32]. Compared to the findings in the SP, the KPI test sent data at a rate of 100 kbps, which was more than the observed throughput from the SP. The throughput of this test scenario from the SP originated from receiving sensor readings while sending robot commands. This peaked at around 80 kbps. Nevertheless, no video stream from the robot was sent during the SP.

The findings from the SP show that multiple high peaks in latency were observed for the Wi-Fi link between the Raspberry Pi on the robot and a Work Station. Throughout the project, no reason for this was discovered. Regardless, the latency of the system setup between RPR and RPT in this thesis did not experience similar behavior. The highest RTT for all tests was 58 ms, and only 0.01% and 0.02% packets were lost for UL and DL, respectively. In contrast, scenario 1 from SP showed a max RTT of 1153 ms for UDP and 3021 ms for TCP and 3% packet loss.

5.3 Scenario 2: RPD 5G - RPT LAN

This section represents the fundamental findings from the executed tests between RPD and RPT, while RPD was connected to the TDC 700 MHz 5G network. Illustration and description of the scenario were discussed in section 4.2.3. Section 5.3.1, section 5.3.2 and section 5.3.3 presents findings from the N-KPI, S4B and ST tests respectively. Section 5.3.4 summarizes the test scenario by presenting the resulting KPIs. Section 5.3.5 discusses and evaluates the results of these findings.

5.3.1 N-KPI Test

The latency findings for the three N-KPI tests for this scenario are given in Figure 5.13 for T1, Figure 5.14 for T2 and Figure 5.15 for T3. These figures show the One Way Delay (OWD) for UL (RPD to RPT) and DL (RPT to RPD). For all tests for this scenario, RPD is represented as the ”from” probe, and RPT is represented as the ”to” probe. Essential latency metrics from these three tests are represented in Table 5.6. This Table shows maximum (Max) and average (Avg) latency for the UL, DL, and Round-Trip Time (RTT).

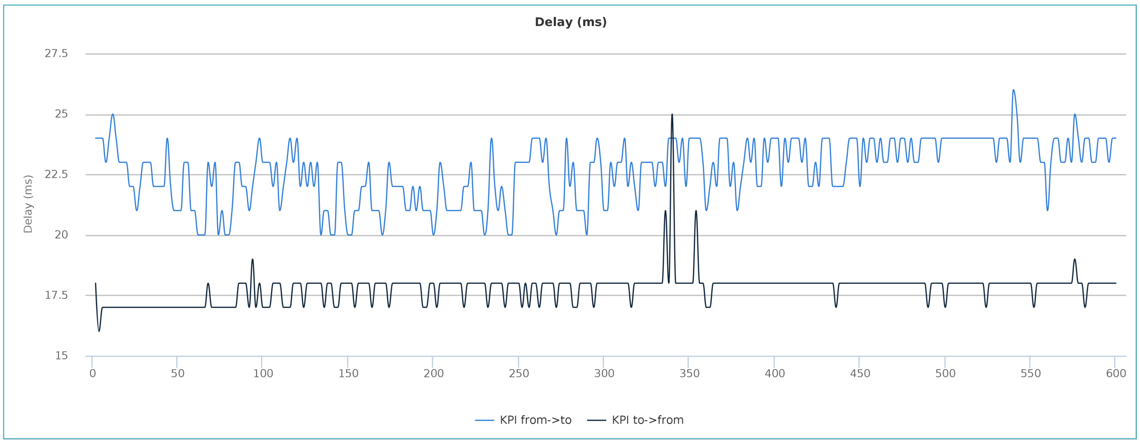


Figure 5.13: Latency findings for 10 minutes of the N-KPI T1 between the RPD and RPT. Blue line represents UL and black line is the DL.

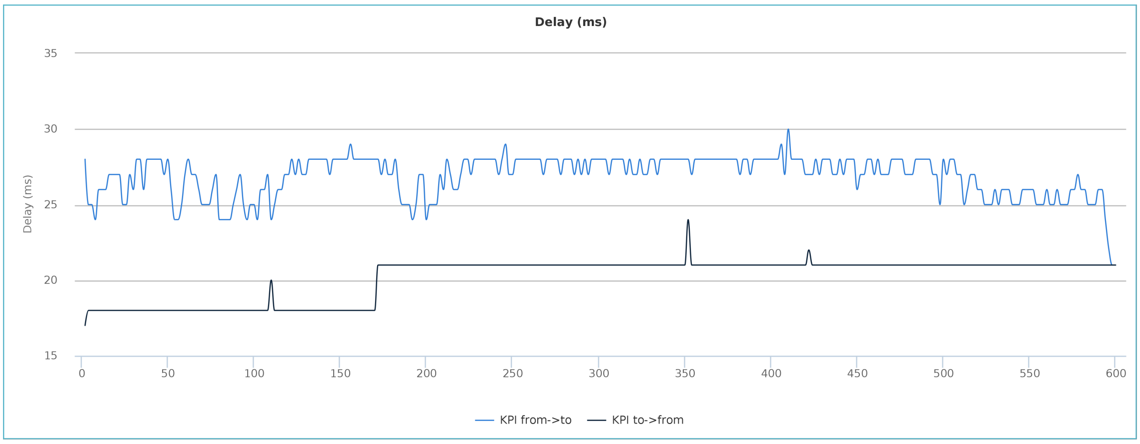


Figure 5.14: Latency findings for 10 minutes of the N-KPI T2 between the RPD and RPT. Blue line represents UL and black line is the DL.

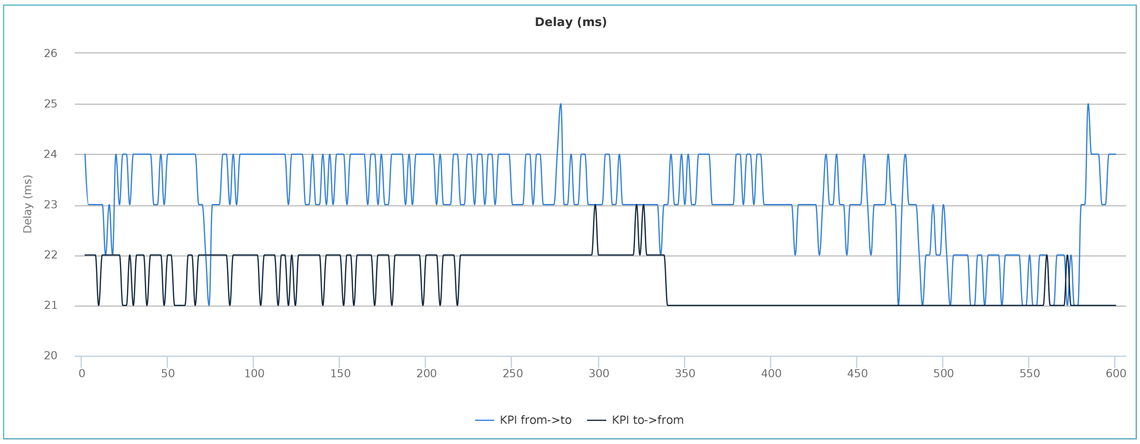


Figure 5.15: Latency findings for 10 minutes of the N-KPI T3 between the RPD and RPT. Blue line represents UL and black line is the DL.

Table 5.6: N-KPI latency observations from the three tests.

N-KPI Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	26	22.66	25	17.79	51	40.45
T2	30	26.9	24	20.17	54	47.07
T3	25	23.15	23	21.5	48	44.65

Packet Loss during N-KPI
There were detected packet loss during the 1st. This percentage of the loss is shown in Figure 5.16, peaking twice at 1.00% and 0.99% packets lost, for DL only. Results from all three tests show a packet loss percentage through each 10-minute test, shown in Table 5.7

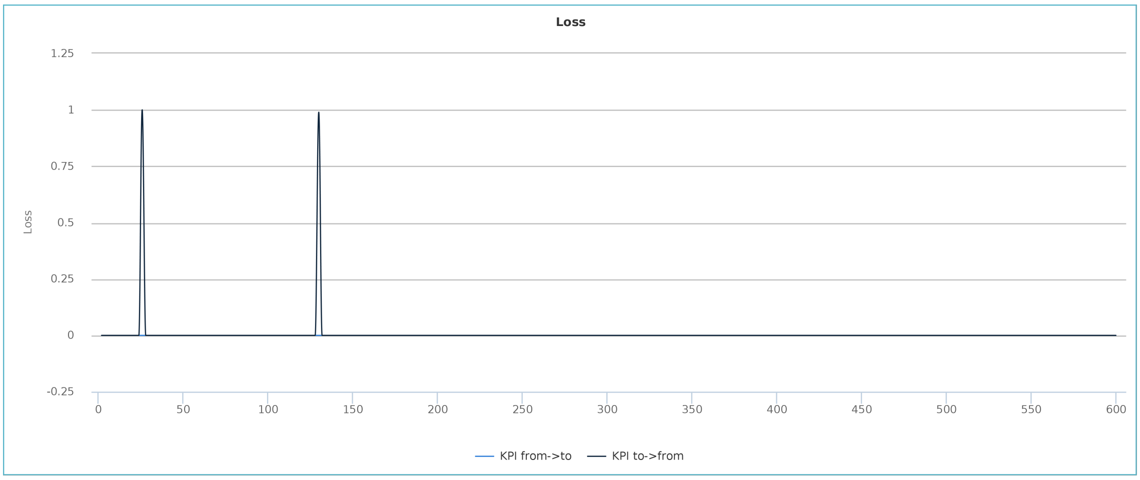


Figure 5.16: Loss during the KPI T1.

Table 5.7: Percentage packet loss from three N-KPI tests.

Packet Loss N-KPI		
Test	UL (%)	DL (%)
T1	0	0.01
T2	0	0
T3	0	0

5.3.2 S4B Test

A 10 Mbps video stream was sent both ways between two endpoints. The streaming services from S4B T1 are represented in Figure 5.17.

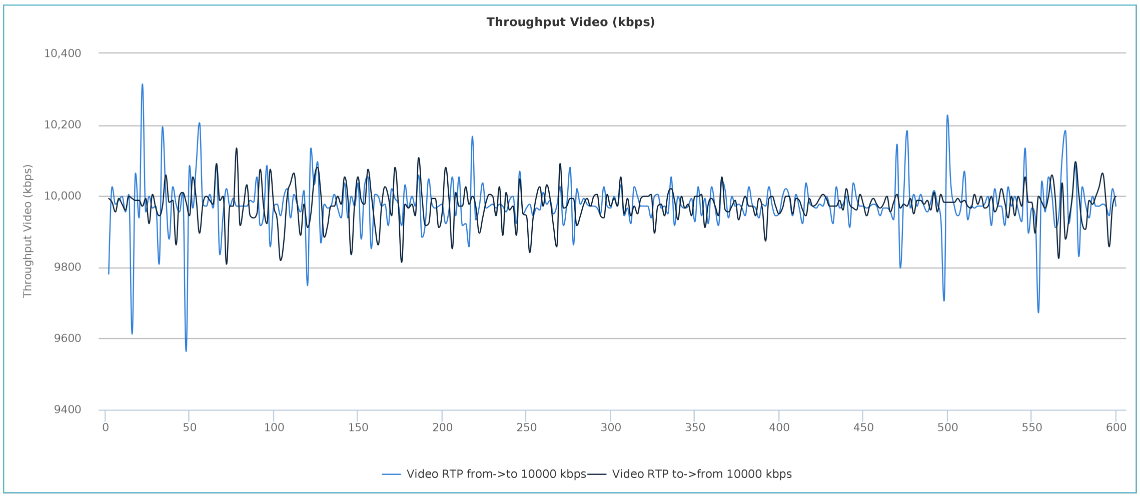


Figure 5.17: The bidirectional streaming service between RPD and RPT.

The OWD of S4B T1, T2 and T3 are presented in Figure 5.18, Figure 5.19 and Figure 5.20 respectively. Essential latency metrics from these three tests are represented in Table 5.8.

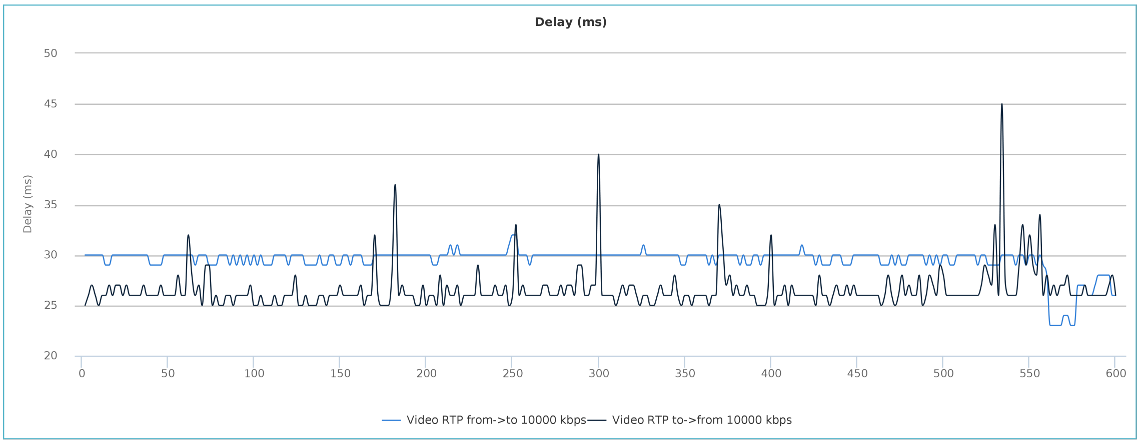


Figure 5.18: Latency findings for 10 minutes of the S4B T1 between the RPD and RPT. Blue line represents UL and black line is the DL.

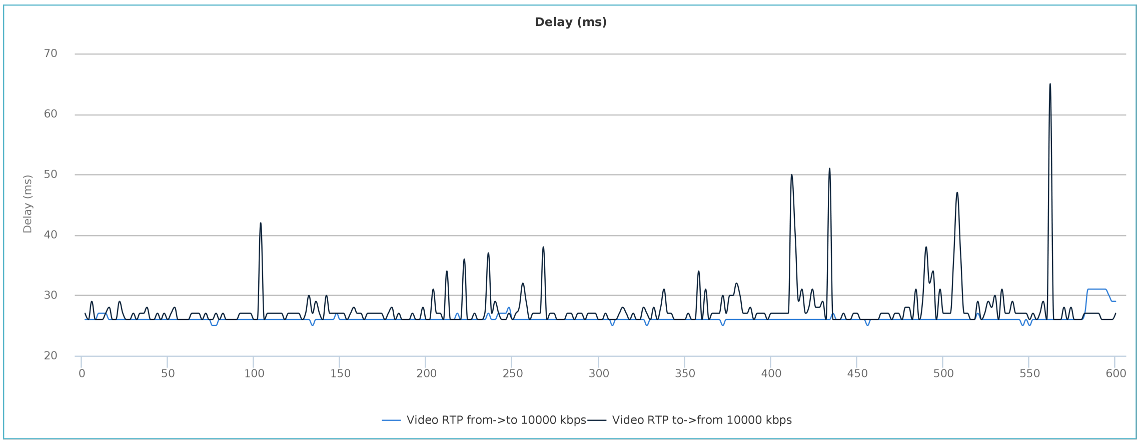


Figure 5.19: Latency findings for 10 minutes of the S4B T2 between the RPD and RPT. Blue line represents UL and black line is the DL.

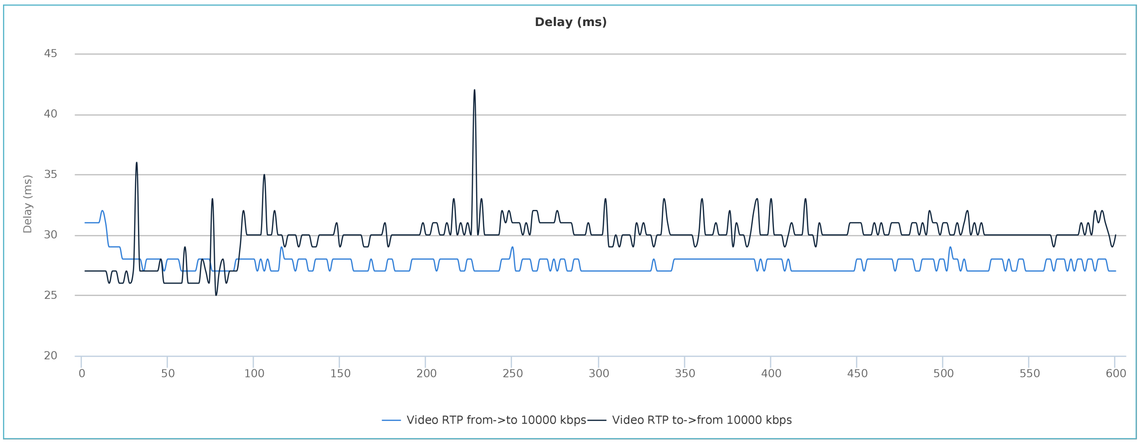


Figure 5.20: Latency findings for 10 minutes of the S4B T3 between the RPD and RPT. Blue line represents UL and black line is the DL.

Table 5.8: S4B latency observations from the three tests.

S4B Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	32	29.49	45	26.59	77	56.08
T2	31	26.15	65	27.7	91.15	28.61
T3	32	27.65	42	29.89	74	57.54

Packet Loss during S4B

Packets were lost for the DL during all three tests of S4B. Similar behavior between the tests. Output from S4B T2 is shown in Figure 5.21. Multiple spikes are observed, with the greatest at 4.26% packet loss. Results from all three tests show a packet loss percentage represented in Table 5.9.

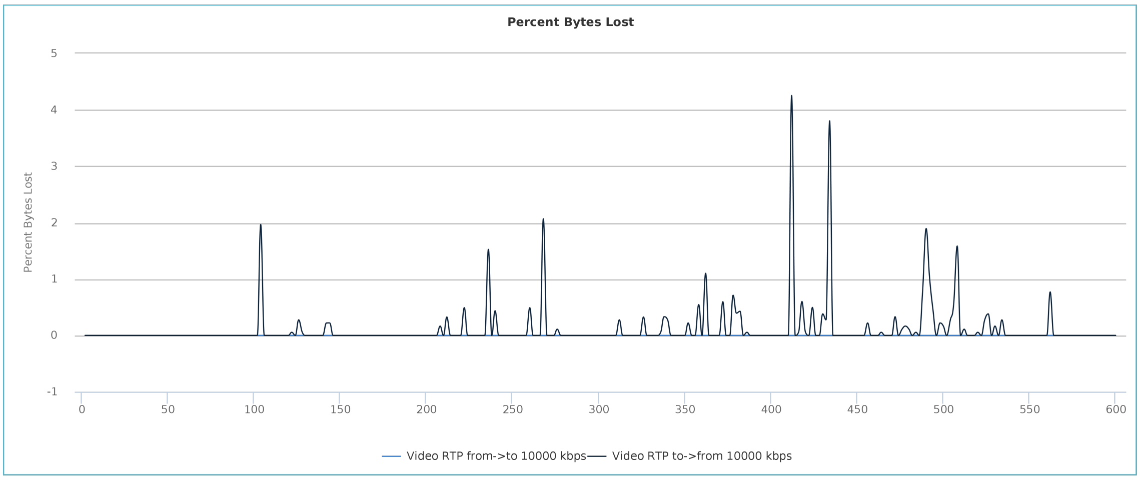


Figure 5.21: Packet loss in percentage during S4B T2.

Table 5.9: Percentage packet loss from three S4B tests.

Packet Loss S4B		
Test	UL (%)	DL (%)
T1	0	0.06
T2	0	0.11
T3	0	0.03

5.3.3 ST test

Findings of the three STs between RPD and RPT are presented in Table 5.10. This Table shows the Average throughput available between the two endpoints over three different periods.

Table 5.10: Average throughput from three STs.

Throughput ST		
Test	UL (kbps)	DL (kbps)
T1	50742	89407
T2	56827	84052
T3	53981	87384

5.3.4 KPI Findings

The communication link between RPD and RPT was monitored, and measurements resulted in the KPIs presented in Table 5.11.

The N-KPI tests for the 5G to LAN scenario showed an average OWD of 24.24 ms for UL and 19.82 for DL, indicating a 25% increase in the UL communication compared to DL. The maximum observed OWD was 30 ms for UL and 25 ms for DL, resulting in a max RTT of 55 ms. Packet loss was only observed during N-KPI T1, where 0.01% of the packets were lost.

For the S4B tests, the 5G to LAN experienced almost the same increase in OWD for DL compared to the N-KPI tests, a 26% rise for average RTT. The average OWD for the S4B tests was 27.76

ms for UL and 28.06 for DL, with max OWD at 32 for UL and 65 for DL. The OWD peak at 65 ms can be seen in Figure 5.19. Also, similar periodical OWD peaks for DL are measured for all S4B tests. Packets were lost for DL through all three S4B tests. 0.06% for T1, 0.11% for T2 and 0.03% for T3.

The speedtests showed an average throughput of 53.85 Mbps for UL and 86.95 Mbps for DL.

Table 5.11: Summary of the KPI's collected for the RPD-RPT test scenario.

KPIs										
	OWD N-KPI (ms)		**OWD S4B (ms)**		**Throughput ST (kbps)**		**Reliability N-KPI (%)**		**Reliability S4B (%)**	
Test	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL
T1	22.66	17.79	29.49	26.59	50742	89407	100	99.99	100	99.94
T2	26.9	20.17	26.15	27.7	56827	84052	100	100	100	99.89
T3	23.15	21.5	27.65	29.89	53981	87384	100	100	100	99.97
Avg	**24.24**	**19.82**	**27.76**	**28.06**	**53850**	**86948**	**100**	**99.997**	**100**	**99.933**

5.3.5 Discussion

The difference in OWD for N-KPI between the RPD - RPT and RPR - RPT scenarios indicates that if RPR sent sensor readings, <100 kbps, to RPT, and then RPT sent it to RPD, it would take 16.17 ms (UL for RPR - RPT) plus 19.82 ms (DL for RPD - RPT). Nevertheless, this is an estimation, and things like processing data for the RPT are not considered. Regardless, the OWD observed from both the N-KPI and S4B tests shows a more stable latency measurement when compared to the results from the SP. But the reliability of the system for the DL between the RPD and RPT is impacted during the S4B tests, which results in a 99.93% reliability.

This scenario showed an increase in latency and a decrease in throughput compared to the RPR - RPT scenario. The UL/DL for N-KPI and S4B tests showed an increase of 49.92%/64.32% and 37.76%/48.60%, respectively. When considering the distance the packets travel, an increase in latency is expected.

The immense impact on the throughput decrease is the UL, where 53.80 Mbps were measured, compared to 75.96 Mbps for the RPR - RPT scenario. This is a 29.10% decrease. In addition, TDC reports the throughput at the area around the office to be 50-100 Mbps for both LTE and 5G, which is comparable to the speeds achieved for the speedtests.

5.4 Scenario 3: RPD LAN - RPT LAN

This section is presents findings from the LAN to LAN scenario, where both the RPD and RPT is connected to the local university LAN via Ethernet. Illustration and description of the scenario was discussed in section 4.2.3. Section 5.4.1, section 5.4.2 and section 5.4.3 presents findings from the N-KPI, S4B and ST tests respectively. Section 5.4.4 summarizes the test scenario with presenting the resulting KPIs. Section 5.4.5 discusses and evaluates the results of these findings.

5.4.1 N-KPI Test

The latency measurements from the three N-KPI tests for this scenario is given in Figure 5.22 for T1, Figure 5.23 for T2 and Figure 5.24 for T3. These figures shows the One Way Delay (OWD) for UL (RPD to RPT) and DL (RPT to RPD). For all tests for this scenario, RPD is represented as the "from" probe, and RPT is represented as the "to" probe. Essential latency metrics from these three tests are represented in Table 5.12. This table shows maximum (Max) and average (Avg) latency for the UL, DL and Round-Trip Time (RTT).

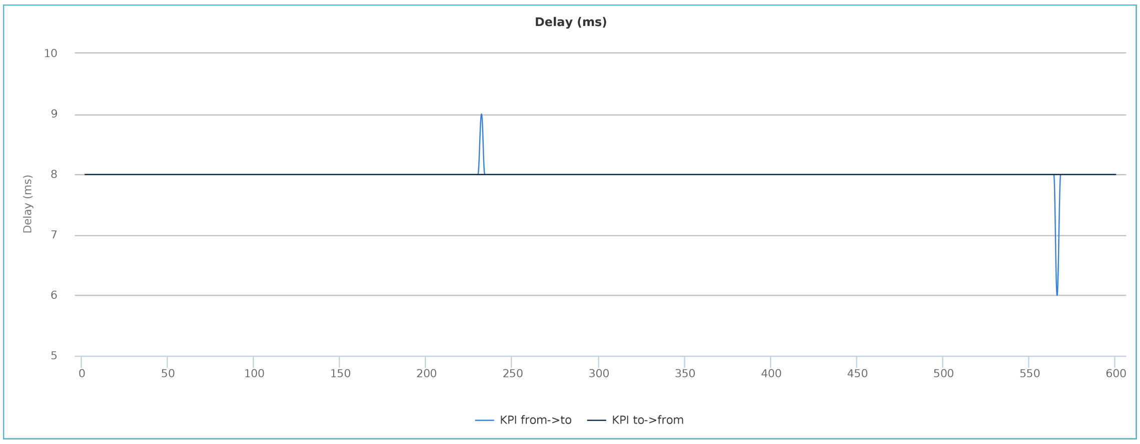


Figure 5.22: Latency findings for 10 minutes of the N-KPI T1 between the RPD and RPT. Blue line represents UL and black line is the DL.

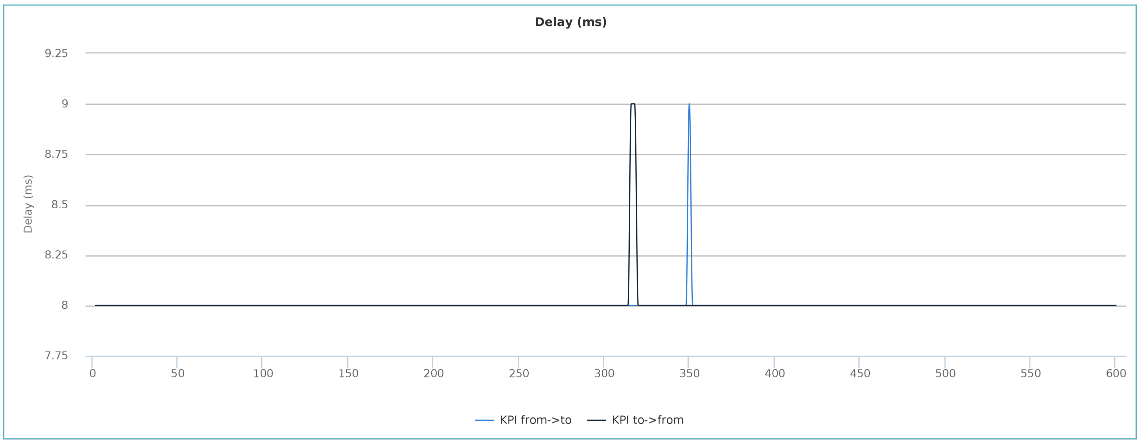


Figure 5.23: Latency findings for 10 minutes of the N-KPI T2 between the RPD and RPT. Blue line represents UL and black line is the DL.

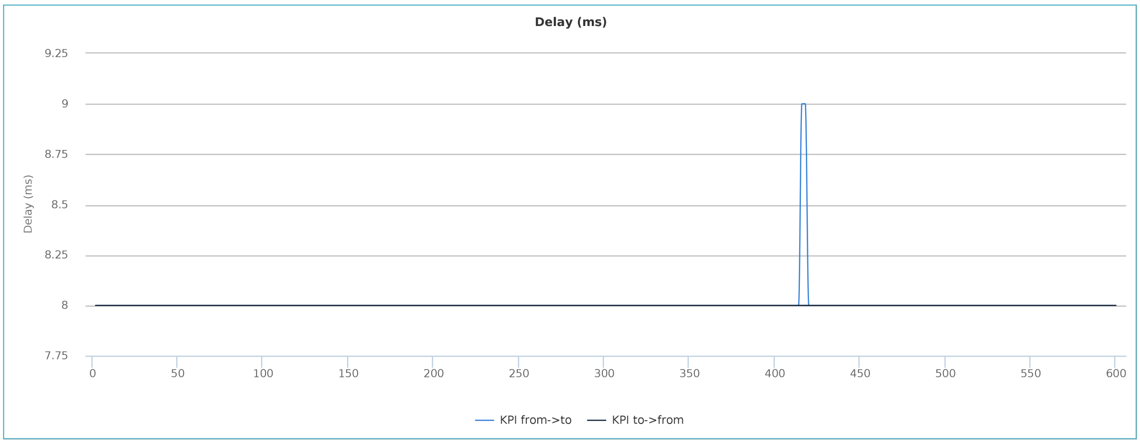


Figure 5.24: Latency findings for 10 minutes of the N-KPI T3 between the RPD and RPT. Blue line represents UL and black line is the DL.

Table 5.12: N-KPI latency observations from the three tests.

N-KPI Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	9.00	8.00	8.00	8.00	17.00	16.00
T2	9.00	8.00	9.00	8.00	18.00	16.00
T3	9.00	8.00	9.00	8.00	18.00	16.00

Packet Loss during N-KPI
No packet loss were observed over N-KPI's three test periods.

5.4.2 S4B Test

A 10 Mbps video-stream was sent both ways between the two endpoints. The streaming services from S4B T1 are represented in Figure 5.25.

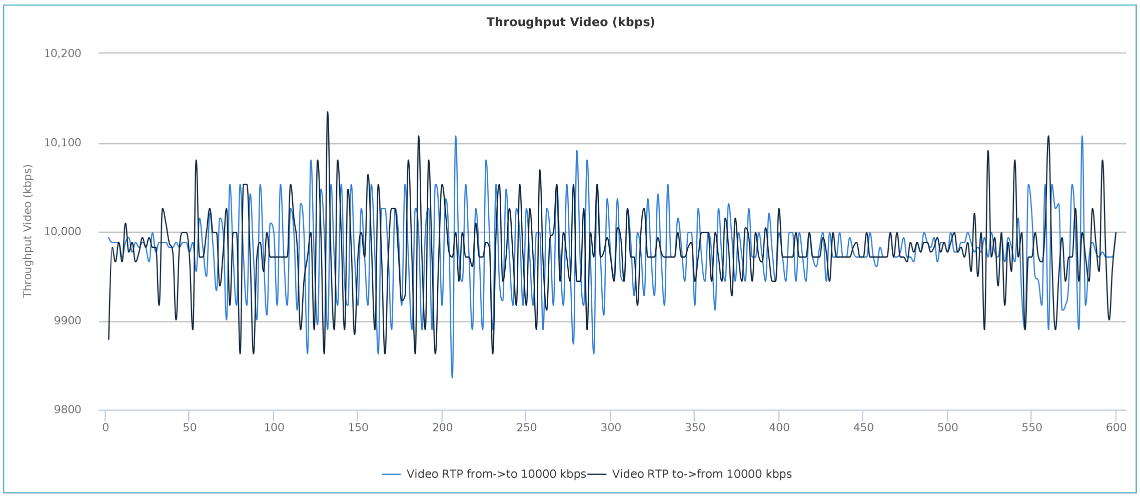


Figure 5.25: The bidirectional streaming service between RPD and RPT.

The OWD of S4B T1, T2 and T3 are presented in Figure 5.26, Figure 5.27 and Figure 5.28 respectively. Essential latency metrics from these three tests are represented in Table 5.13.

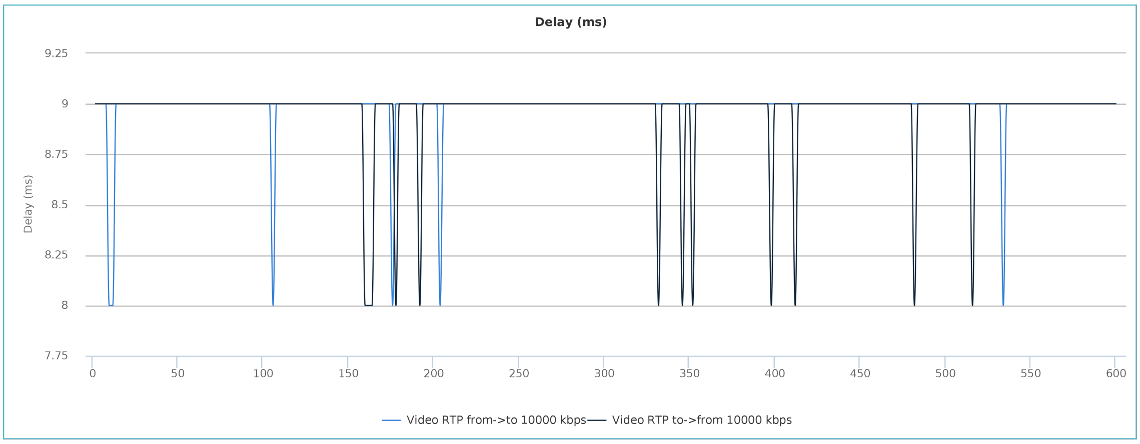


Figure 5.26: Latency findings for 10 minutes of the S4B T1 between the RPD and RPT. Blue line represents UL and black line is the DL.

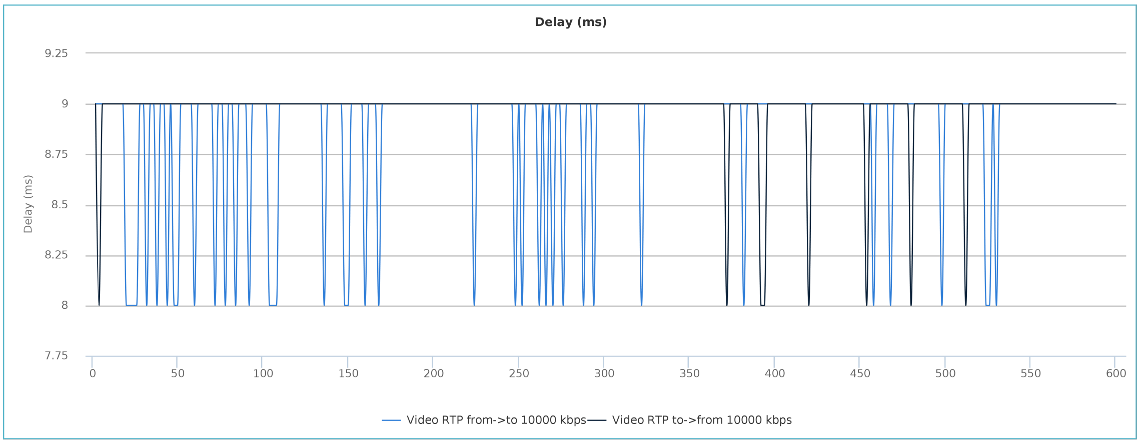


Figure 5.27: Latency findings for 10 minutes of the S4B T2 between the RPD and RPT. Blue line represents UL and black line is the DL.

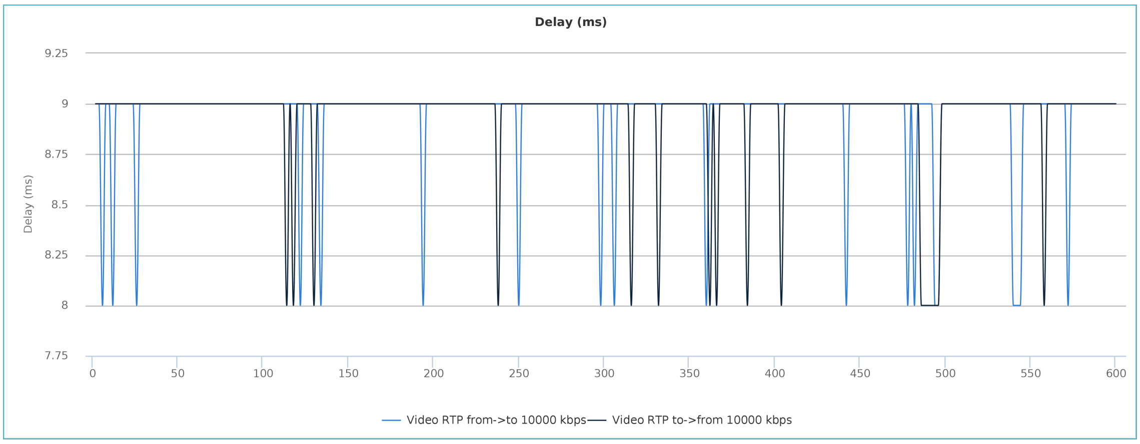


Figure 5.28: Latency findings for 10 minutes of the S4B T3 between the RPD and RPT. Blue line represents UL and black line is the DL.

Table 5.13: S4B latency observations from the three tests.

S4B Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	9.00	8.94	9.00	8.94	18.00	17.88
T2	9.00	8.94	9.00	8.94	18.00	17.88
T3	9.00	8.94	9.00	8.94	18.00	17.88

Packet Loss during S4B

Packets were lost for DL connection during T1. Output from S4B T1 is shown in Figure 5.29. Multiple spikes are observed, with the greatest at 0.17% packets loss. Results from all three tests show a packet loss percentage represented in Table 5.14.

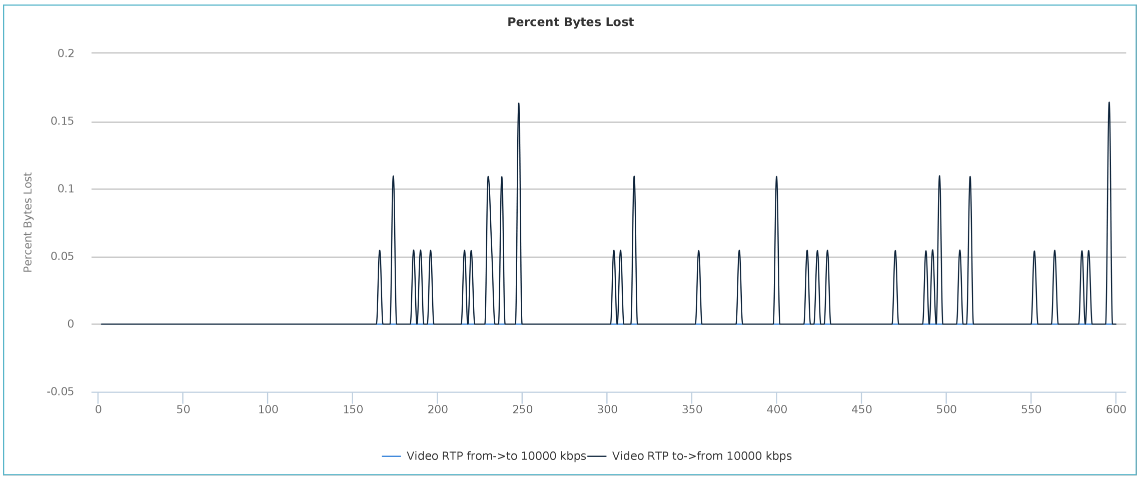


Figure 5.29: Packet loss in percentage during S4B T1.

Table 5.14: Percentage packet loss from three S4B tests.

Packet Loss S4B		
Test	UL (%)	DL (%)
T1	0	0.01
T2	0	0
T3	0	0

5.4.3 ST test

Findings of the three STs between RPD and RPT are presented in Table 5.15. This Table shows the Average throughput available between the two endpoints over three different time periods.

Table 5.15: Average throughput from three STs.

Throughput ST		
Test	UL (kbps)	DL (kbps)
T1	92253	90867
T2	92346	90830
T3	85849	90979

5.4.4 KPI Findings

The LAN to LAN communication link between RPD and RPT was monitored, and measurements resulted in the KPIs presented in Table 5.16.

This scenario was based on a wire-wire connection, and the OWD results from the N-KPI indicate that this is the case, with a stable 8 ms OWD for *both* UL and DL. This results in a 16 ms RTT for the LAN to LAN scenario. No packets were lost for the N-KPI tests.

For the S4B tests, a more alternating pattern was observed, with the OWD for both UL and DL oscillate between 8 and 9 ms, shown in Figure 5.26. The average was 8.94 ms for both UL and DL, concluding in an average of 17.88 ms RTT. Nonetheless, packets were lost during S4B T1 for DL. There can be many reasons for this, e.g., router queues being full, interference on the wires, or switches going down.

The throughput tests show a bidirectional speed of around 90 Mbps.

Table 5.16: Summary of the KPI's collected for the RPD-RPT test scenario.

KPIs										
	OWD N-KPI (ms)		**OWD S4B (ms)**		**Throughput ST (kbps)**		**Reliability N-KPI (%)**		**Reliability S4B (%)**	
Test	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL
T1	8.00	8.00	8.94	8.94	92253	90867	100	100	100	99.99
T2	8.00	8.00	8.94	8.94	92346	90830	100	100	100	100
T3	8.00	8.00	8.94	8.94	85849	90979	100	100	100	100
Avg	**8.00**	**8.00**	**8.94**	**8.94**	**90149**	**90892**	**100**	**100**	**100**	**99.997**

5.4.5 Discussion

Compared to the 5G to LAN scenario, the LAN to LAN is advantageous due to less interference related to the wired connection. This resulted in better performance in terms of KPIs and a 64% decrease in RTT for N-KPI tests and 68% for the S4B tests. Regarding packet loss, the LAN to LAN scenario has 100% availability for the N-KPI tests and 99.997% for the S4B test.

The results showed significant better KPI values for the LAN to LAN scenario, compared to the RPR - RPT scenario, where the two endpoints were located <100 meters apart. This also shows that the latency of the other two scenarios does not represent the expected values of a 5G system.

The throughput tests show a bidirectional speed of around 90 Mbps. Compared to the 5G to LAN scenario, this can indicate that maybe the DL is the limiting factor of the throughput, that being RPT's capability of sending packets to RPD.

5.5 Scenario 4: iPhone 12 5G (indoor) - RPT LAN

This section presents findings from the 1st scenario between the iPhone 12, hereby referred to as i12, and the RPT. In this scenario, the iPhone was located at the office next to the RPD. Illustration and description of the scenario were discussed in section 4.2.4. Section 5.5.1, section 5.5.2 and section 5.5.3 presents findings from the N-KPI, S4B and ST tests respectively. Section 5.5.4 summarizes the test scenario by presenting the resulting KPIs. Section 5.5.5 discusses and evaluates the results of these findings.

For this scenario, a redo of the test was done with the same parameters and same system setup. This test is represented in Appendix C.

5.5.1 N-KPI Test

The latency measurements from the three N-KPI tests for this scenario is given in Figure 5.30 for T1, Figure 5.31 for T2 and Figure 5.32 for T3. These figures show the One Way Delay (OWD) for UL (i12 to RPT) and DL (RPT to i12). For all tests for this scenario, i12 is represented as the ”from” probe, and RPT is represented as the ”to” probe. Essential latency metrics from these three tests are represented in Table 5.17. This Table shows maximum (Max) and average (Avg) latency for the UL, DL, and Round-Trip Time (RTT).

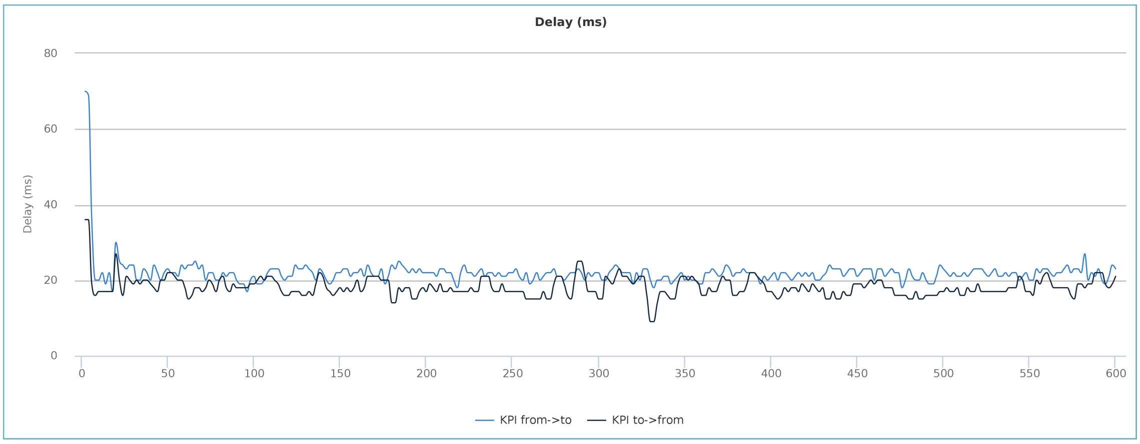


Figure 5.30: Latency findings for 10 minutes of the N-KPI T1 between the i12 and RPT. Blue line represents UL and black line is the DL.

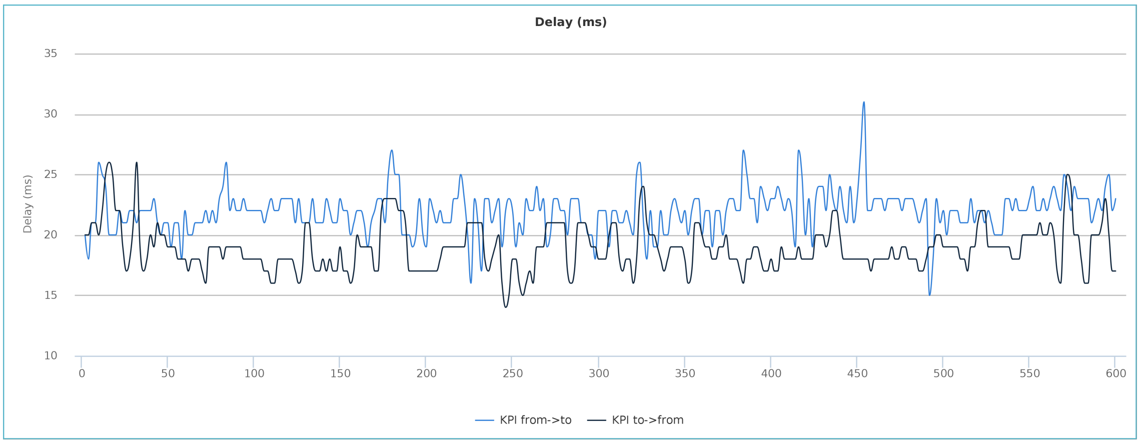


Figure 5.31: Latency findings for 10 minutes of the N-KPI T2 between the i12 and RPT. Blue line represents UL and black line is the DL.

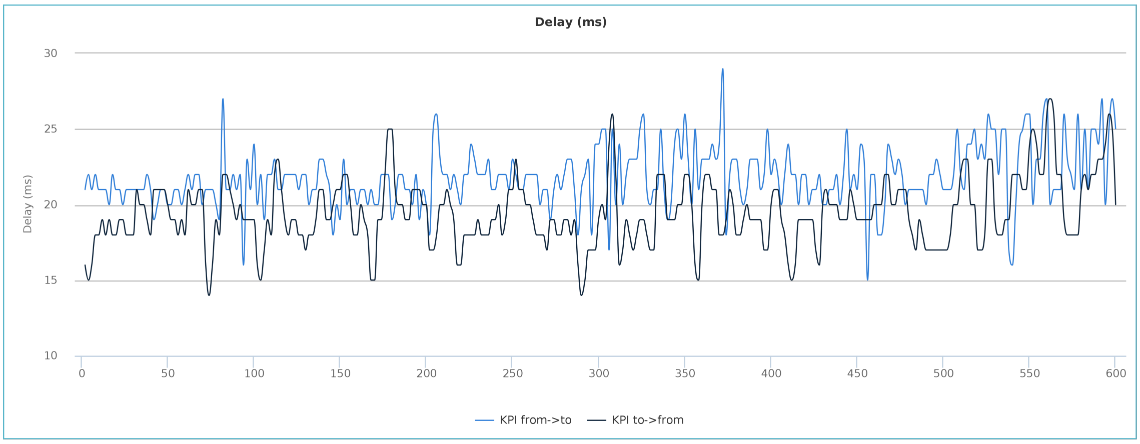


Figure 5.32: Latency findings for 10 minutes of the N-KPI T3 between the i12 and RPT. Blue line represents UL and black line is the DL.

Table 5.17: N-KPI latency observations from the three tests.

N-KPI Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	30	22.04	27	18.17	57	40.21
T2	31	21.95	26	18.87	57	40.82
T3	29	21.84	27	19.43	56	41.27

Packet Loss during N-KPI

Packet loss was observed for the N-KPI T3, and it is presented in Figure 5.33. Y-axis represents the loss in percentage. This presented a packet loss of total 0.02% for T3. No packet loss happened for T1 and T2.

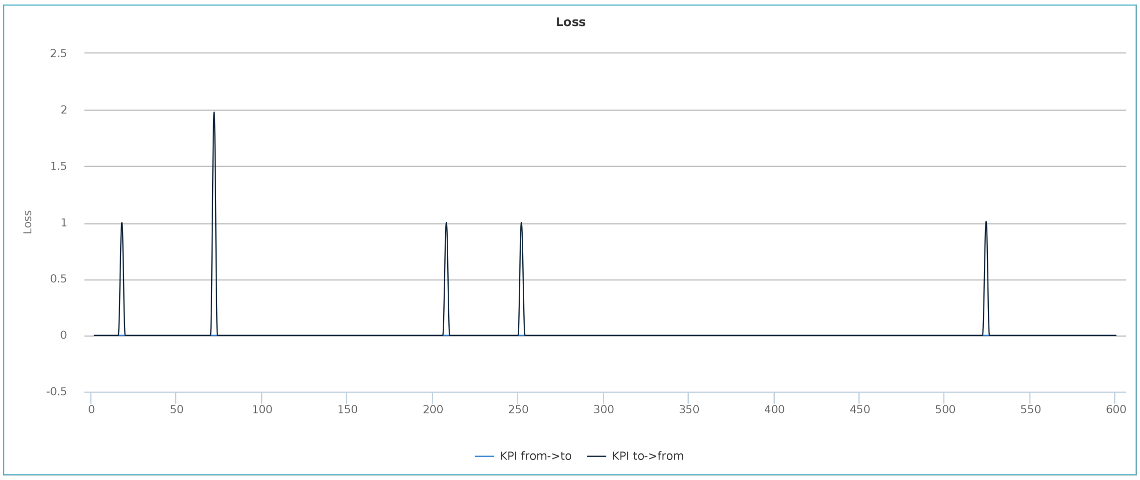


Figure 5.33: The packet loss from N-KPI T3.

5.5.2 S4B Test

A 10 Mbps video stream was sent both ways between the two endpoints. The streaming services from S4B T1, T2 and T3 are represented in Figure 5.34, Figure 5.35 and Figure 5.36 respectively. For T2 and T3, the streaming throughput drops for the UL connection. For T2, the UL cannot withhold a 10 Mbps stream at around the 140-second mark and drops to 2900 kbps before it returns to 10 Mbps again at the 460-second mark. For T3, the UL streaming service gradually decreases from 10 Mbps to the 7000 kbps mark at around 250 to 300-second marks, before gradually increasing again to 10 Mbps at 450 to 550-second marks.

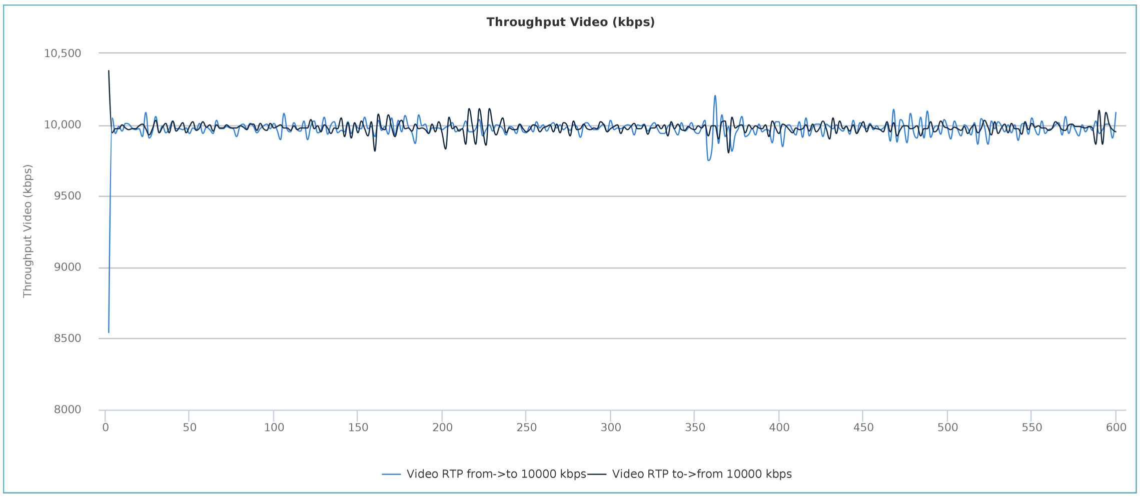


Figure 5.34: The bidirectional streaming service between i12 and RPT for S4B T1.

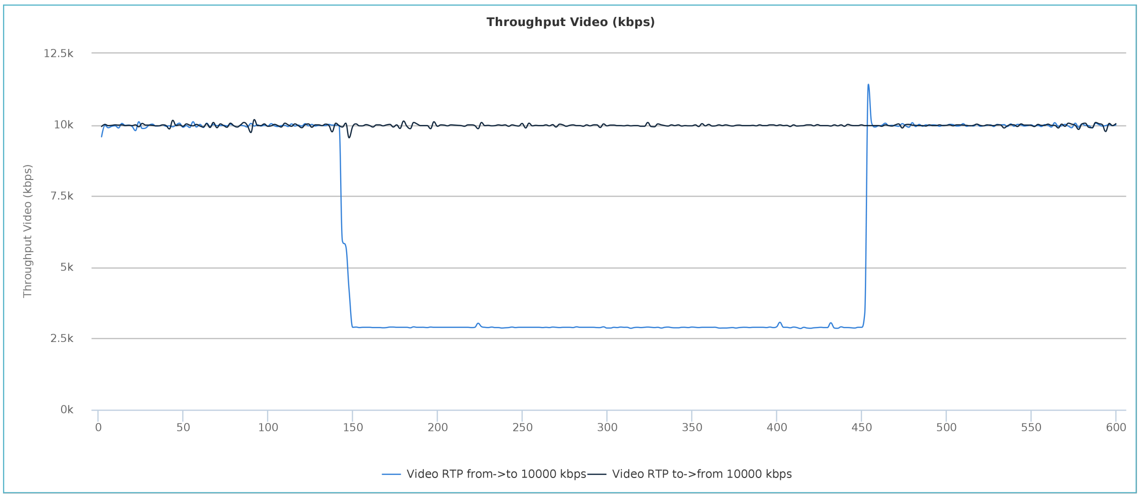


Figure 5.35: The bidirectional streaming service between i12 and RPT for S4B T2.

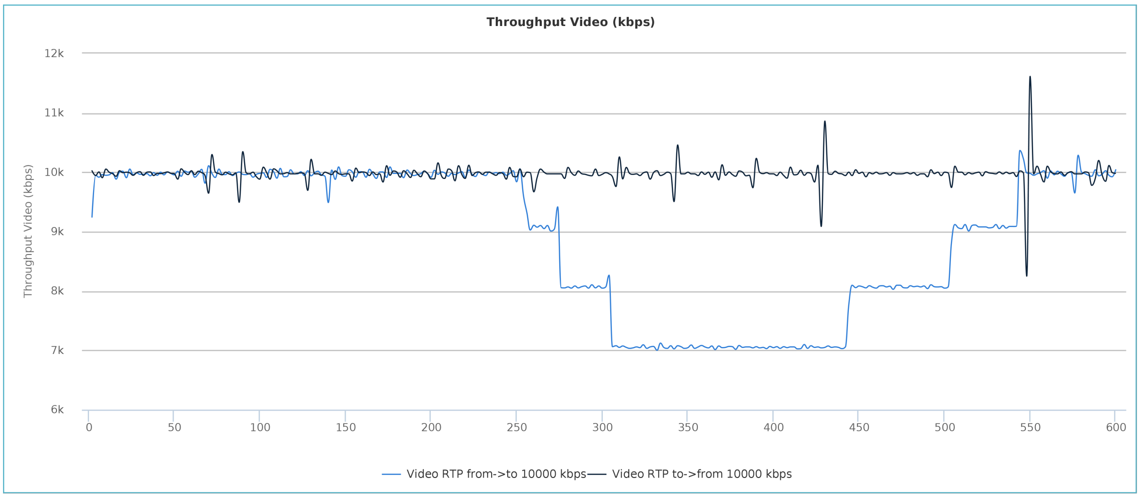


Figure 5.36: The bidirectional streaming service between i12 and RPT for S4B T3.

The OWD of S4B T1, T2 and T3 are presented in Figure 5.37, Figure 5.38 and Figure 5.39 respectively. Similar behavior as the throughput happened with the latency. Essential latency metrics from these three tests are represented in Table 5.18.

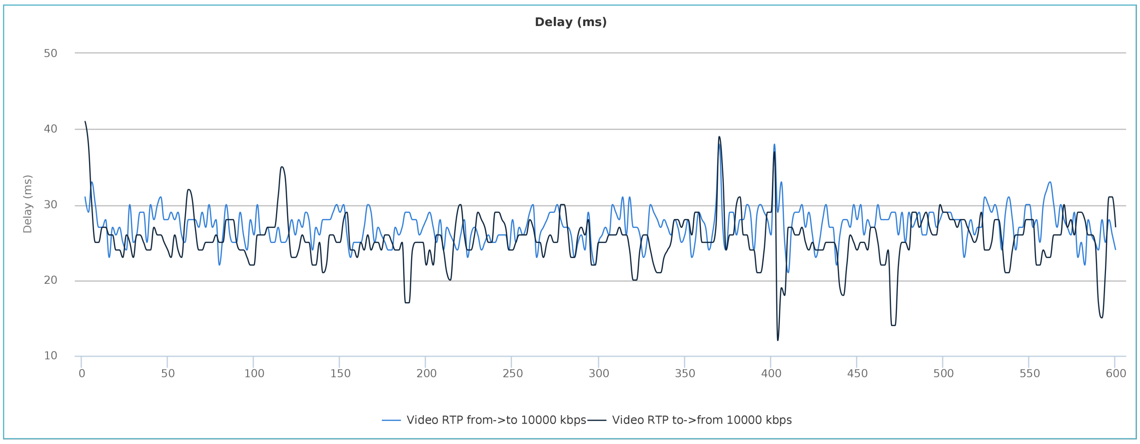


Figure 5.37: Latency findings for 10 minutes of the S4B T1 between the i12 and RPT. Blue line represents UL and black line is the DL.

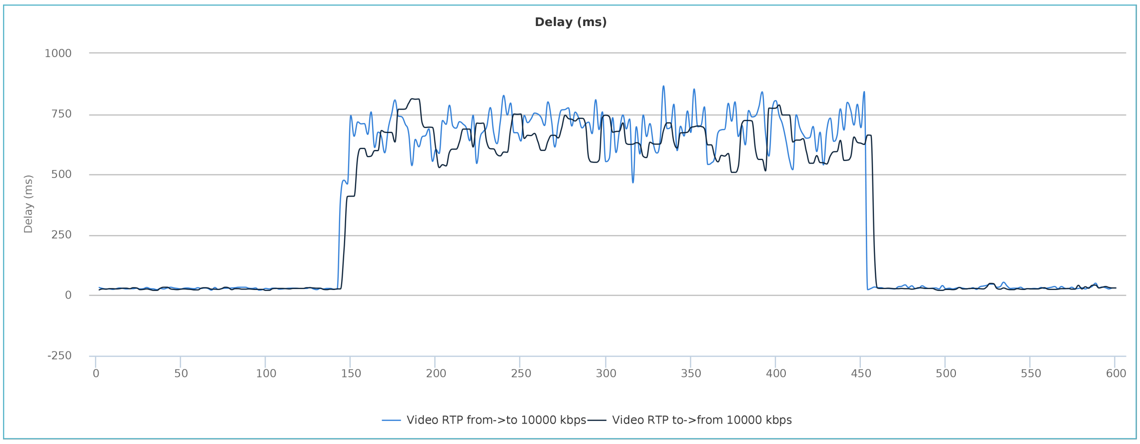


Figure 5.38: Latency findings for 10 minutes of the S4B T2 between the i12 and RPT. Blue line represents UL and black line is the DL.

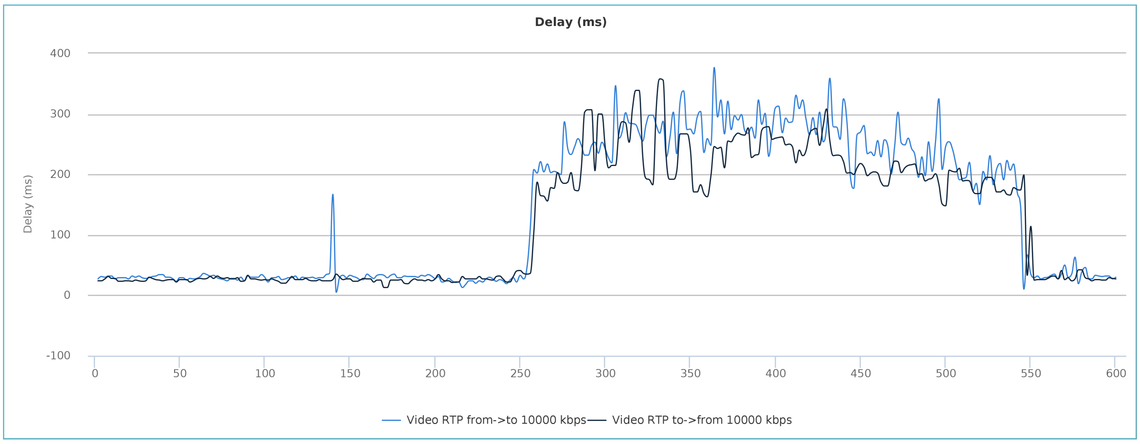


Figure 5.39: Latency findings for 10 minutes of the S4B T3 between the i12 and RPT. Blue line represents UL and black line is the DL.

Table 5.18: S4B latency observations from the three tests.

S4B Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	38	27.21	39	25.47	77	52.68
T2	866	370.43	913	347.06	1779	717.49
T3	377	138.47	358	121.21	735	259.68

Packet Loss during S4B

Packets were lost for DL connection during T1. Outputs from S4B T1, T2 and T3 are shown in Figure 5.40, Figure 5.41 and Figure 5.42 respectively. Y-axis represents the loss in percentage. Similar to the throughput and the latency, the packet loss spikes at the same timestamps. For T2, the packet loss stays at around 70% between 140-460 second marks. For T3, it also follows the gradual increase and decrease of packet loss, which spikes around 30% Results from all three tests show a packet loss represented in Table 5.19.

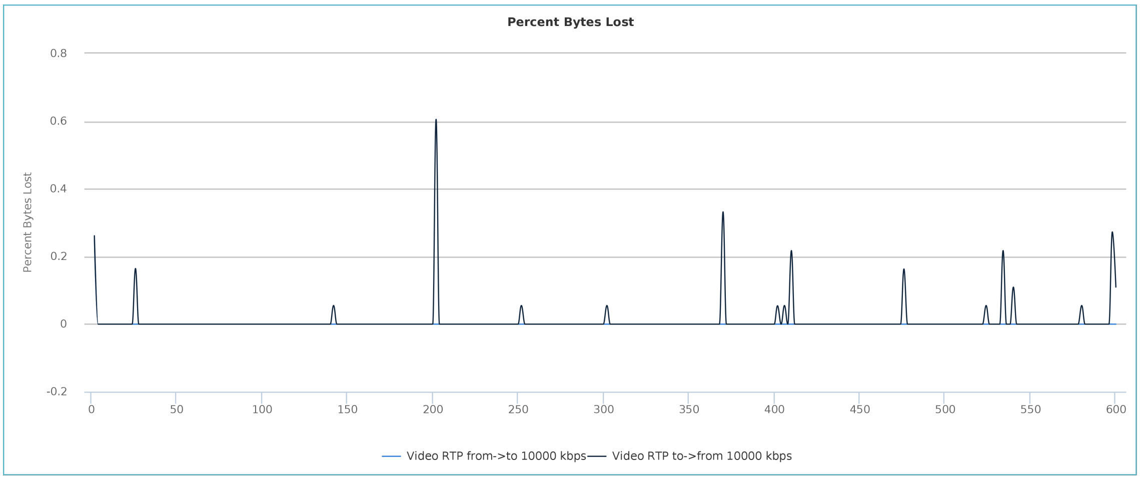


Figure 5.40: Packet loss during S4B T1.

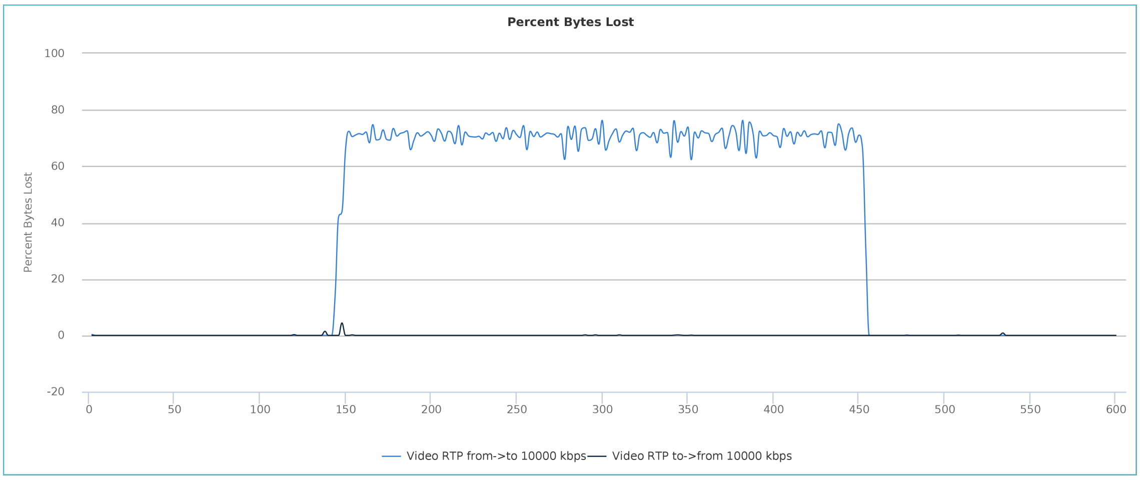


Figure 5.41: Packet loss during S4B T2.

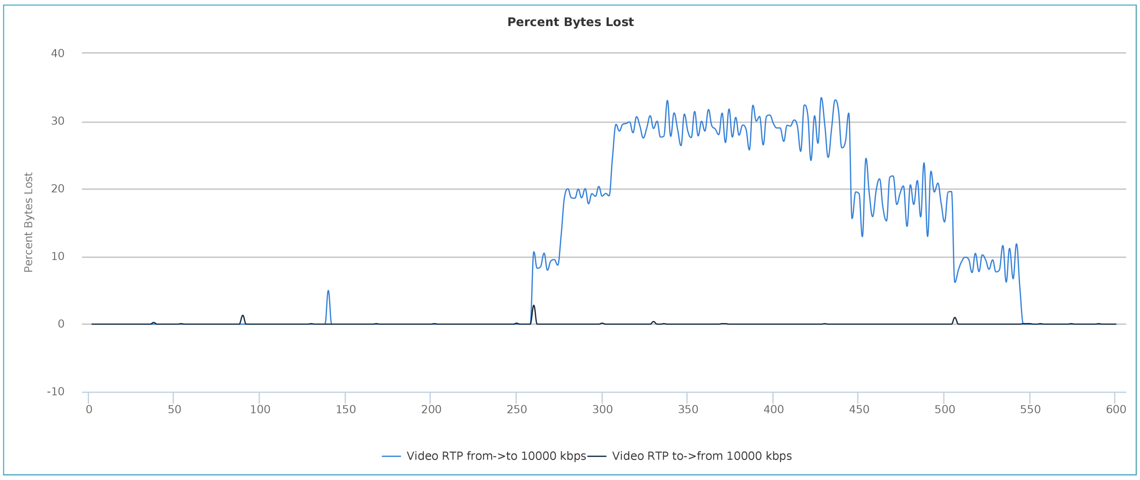


Figure 5.42: Packet loss during S4B T3.

Table 5.19: Percentage packet loss from three S4B tests.

Packet Loss S4B		
Test	UL (%)	DL (%)
T1	0.01	0.01
T2	36.39	0.03
T3	10.50	0.02

5.5.3 ST test

Findings of the three STs between i12 and RPT are presented in Table 5.20. This Table shows the Average throughput available between the two endpoints over three different periods.

Table 5.20: Average throughput from three STs.

Throughput ST		
Test	UL (kbps)	DL (kbps)
T1	18649	47989
T2	17712	90024
T3	12451	83958

5.5.4 KPI Findings

The end-to-end communication link between i12 and RPT was monitored, and measurements resulted in the KPIs presented in Table 5.21.

An average OWD of 21.94 ms for UL and 18.82 ms for DL was observed during the three N-KPI tests. A maximum OWD of 23 ms for UL and 17 ms for DL. This concludes an average RTT of 28.23 ms and a max of 40 ms. T1 was the only test that observed packet loss of 0.01%.

The S4B tests resulted in considerable differences between each test. T1 resulted in the expected results with an average RTT of 52.68 ms, 27.21 ms OWD for UL, and 25.47 ms for DL, measurements represented in Figure 5.37. 0.01% packet loss was measured during T1. For S4B T2, similar behavior was seen until around the 150-second mark, where OWD for both UL and DL raised to around 700 ms, with peaks of 866 ms for UL and 913 for DL. Then at around the 450-second mark, the OWD declined and stabilized at around 30 ms. There was also experienced a rise in packet loss and a drop in throughput for the video stream during the same period. For T3, similar behavior as T2 was measured, but with lower peaks of 377 ms for UL and 358 for DL. The period of the instabilities of T3 was around 300 seconds, similar to T2. The reliability was measured to be 84.37% in total for all the S4B tests, with the greatest reduction induced from T2 were 63.61% reliability.

Due to the instabilities discovered during this test scenario, a retest was executed, see Appendix C. Those retests showed more stable results for S4B T1, T2, and T3 than the initial tests. On average, the OWD for UL 28.19, and the OWD for DL was 27.06. Max OWD for UL was 39 ms and 41 for DL. This means that the RTT was not higher than 80 ms in total. These tests also showed no packet loss for T1 and T2, but a peak in T3 resulted in overall average reliability of 100% for UL and 99.993 for DL.

Table 5.21: Summary of the KPI's collected for the i12-RPT test scenario.

KPIs										
	OWD N-KPI (ms)		**OWD S4B (ms)**		**Throughput ST (kbps)**		**Reliability N-KPI (%)**		**Reliability S4B (%)**	
Test	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL
T1	22.04	18.17	27.21	25.47	18649	47989	100	100	99.99	99.99
T2	21.95	18.87	370.43	347.06	17712	90024	100	100	63.61	99.97
T3	21.84	19.43	138.47	121.21	12451	83958	100	99.98	89.5	99.98
Avg.	**21.94**	**18.82**	**178.70**	**164.58**	**16271**	**73990**	**100**	**99.993**	**84.367**	**99.980**

5.5.5 Discussion

This test scenario achieved KPI values that introduced a retest to evaluate if the KPI values were non-periodical, meaning it would not appear again. However, the original S4B T1 test did not experience similar behavior as T2 and T3. Therefore this can indicate that this was interference from other things than the network connection.

Compared to the RPD - RPT scenario, the original test performed, as previously discussed, very differently. Nevertheless, the retest measured almost the same OWD from both N-KPI tests and S4B tests. Throughput was better for UL for the RPD than iPhone retest, 53.85 Mbps versus 38.68 Mbps. The same reliability for the N-KPI tests of the original and retest scenarios was 100% for UL and 99.993% for DL. This was marginally worse than what was observed for RPD - RPT 5G scenario, where 99.996% were measured.

The reasons for the instabilities of these tests could be many. A UL speed of around 16,27 Mbps and a DL speed of 73.99 Mbps was average for the throughput tests. If the UL dropped under 10 Mbps, this could induce a high delay since Hawkeye tried to stream a 10 Mbps stream towards the RPT. Another possibility is that the iPhone started some background tasks. This is also hard to say because the iPhone has significantly fewer configurable settings than a Raspberry Pi. Applications and updates can take priority of the bandwidth.

From the retest's throughput tests, it is observed that the throughput averaged around 38.68 Mbps, which is over twice as much available throughput as the initial tests. This also supports the theory that during the original S4B T2 and T3 video stream, the available throughput could have dropped below 10 Mbps, such that the iPhone could not throughput the stream to the cellular network.

5.6 Scenario 5: iPhone 12 5G (outdoor) - RPT LAN

This section presents findings from the second scenario between the iPhone 12 and the RPT, presented in section 4.2.4. Here, the iPhone was located outdoor for better RSSI. Section 5.6.1, section 5.6.2 and section 5.6.3 presents findings from the N-KPI, S4B and ST tests respectively. Section 5.6.4 summarizes the test scenario with presenting the resulting KPIs. Section 5.6.5 discusses and evaluates the results of these findings.

5.6.1 N-KPI Test

The latency measurements from the three N-KPI tests for this scenario is given in Figure 5.43 for T1, Figure 5.44 for T2 and Figure 5.45 for T3. These figures show the One Way Delay (OWD) for UL (i12 to RPT) and DL (RPT to i12). For all tests for this scenario, i12 is represented as the "from" probe, and RPT is represented as the "to" probe. Essential latency metrics from these three tests are represented in Table 5.22. This Table shows maximum (Max) and average (Avg) latency for the UL, DL, and Round-Trip Time (RTT).

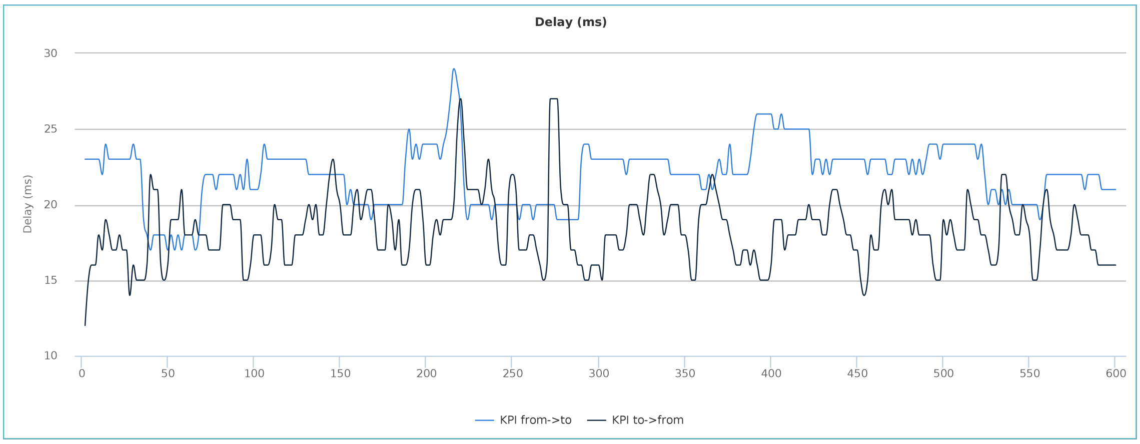


Figure 5.43: Latency findings for 10 minutes of the N-KPI T1 between the i12 and RPT. Blue line represents UL and black line is the DL.

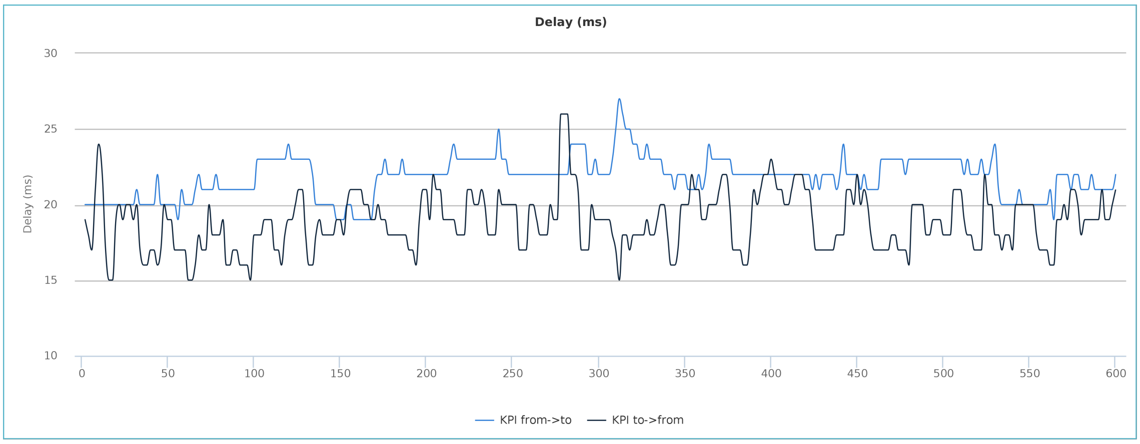


Figure 5.44: Latency findings for 10 minutes of the N-KPI T2 between the i12 and RPT. Blue line represents UL and black line is the DL.

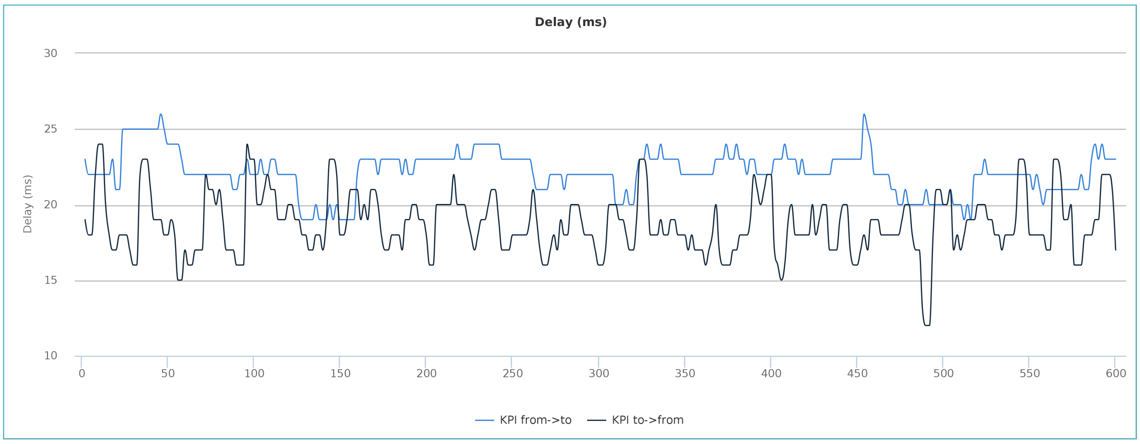


Figure 5.45: Latency findings for 10 minutes of the N-KPI T3 between the i12 and RPT. Blue line represents UL and black line is the DL.

Table 5.22: N-KPI latency observations from the three tests.

N-KPI Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	29	21.97	27	18.31	56	40.28
T2	27	21.81	26	18.84	53	40.65
T3	26	22.16	24	18.7	50	40.86

Packet Loss during N-KPI

Packet loss was observed for the N-KPI T3, and it is presented in Figure 5.46. Y-axis represents the loss in percentage. The figure presents a packet loss in total 0.01% for T3. No packet loss was measured for T1 and T2.

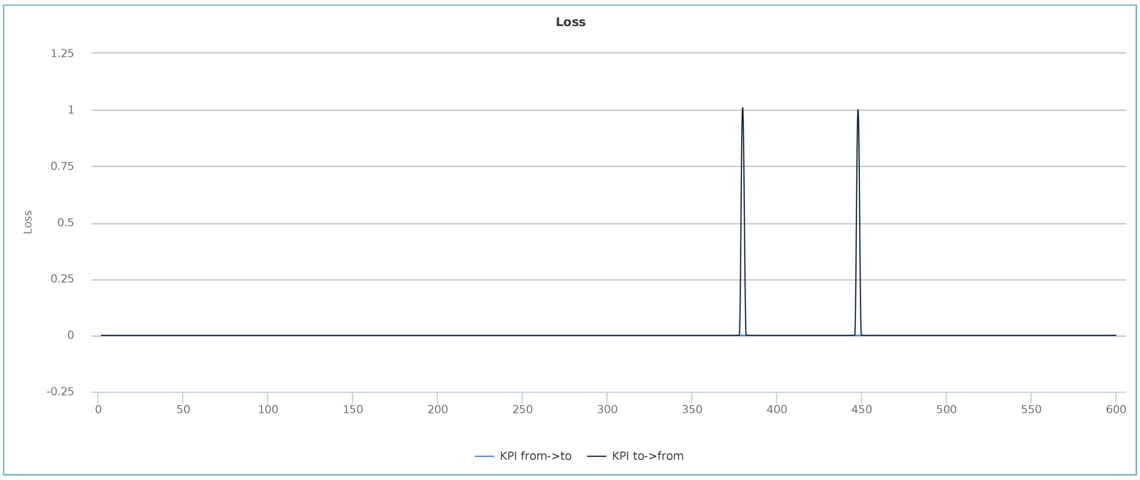


Figure 5.46: The packet loss from N-KPI T3.

5.6.2 S4B Test

A 10 Mbps video-stream was sent both ways between the two endpoints. The streaming services from S4B T1 are represented in Figure 5.47.

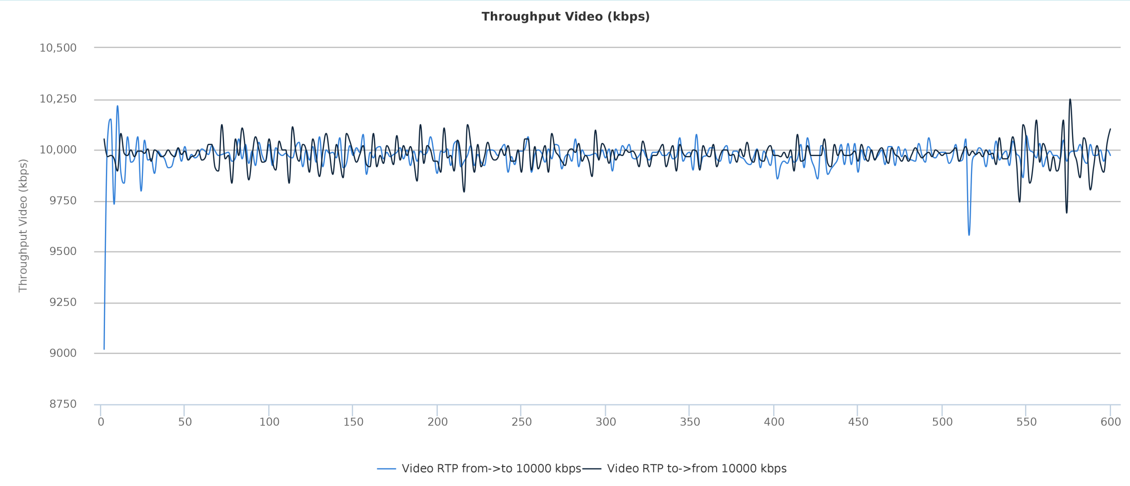


Figure 5.47: The bidirectional streaming service between i12 and RPT for S4B T1.

The OWD of S4B T1, T2 and T3 are presented in Figure 5.48, Figure 5.49 and Figure 5.50 respectively. Essential latency metrics from these three tests are represented in Table 5.23.

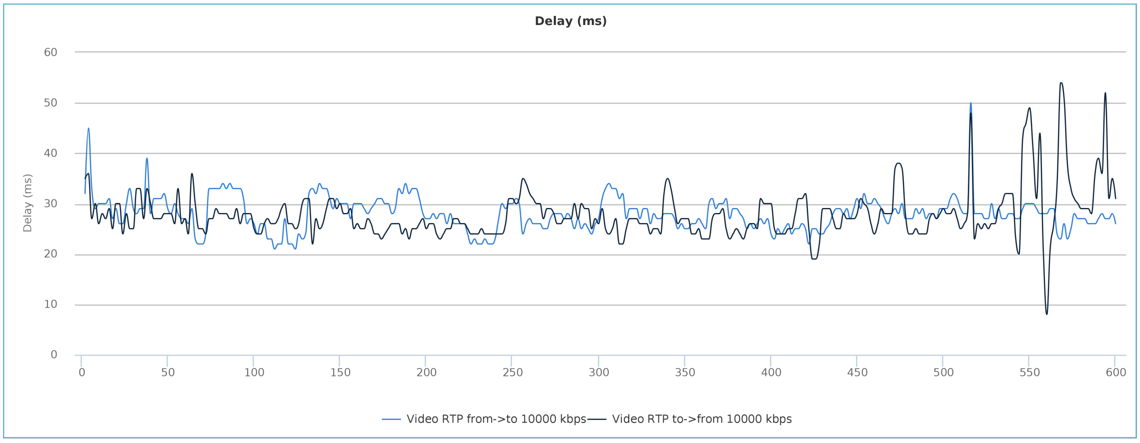


Figure 5.48: Latency findings for 10 minutes of the S4B T1 between the i12 and RPT. Blue line represents UL and black line is the DL.

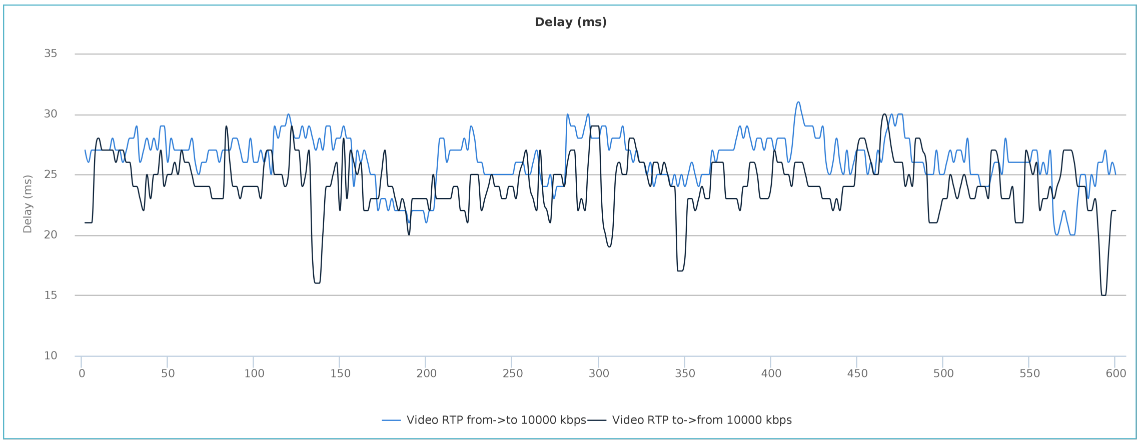


Figure 5.49: Latency findings for 10 minutes of the S4B T2 between the i12 and RPT. Blue line represents UL and black line is the DL.

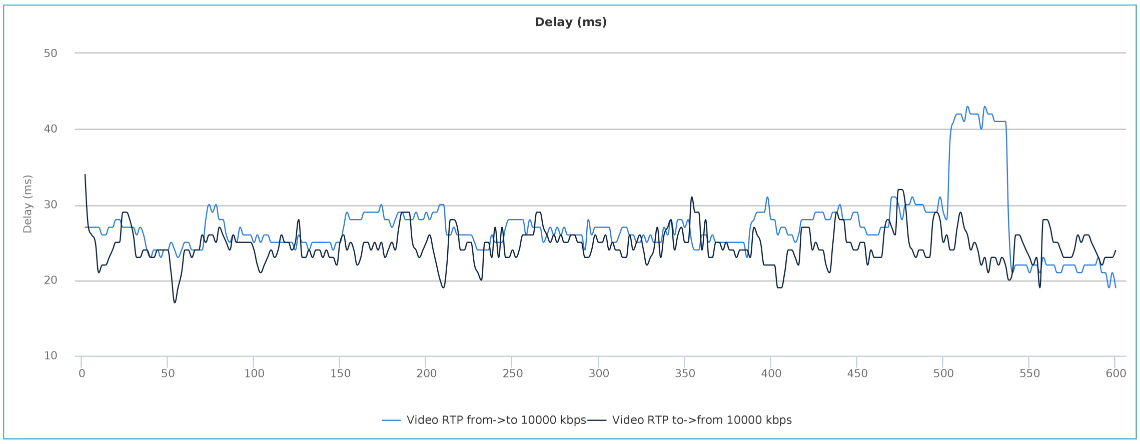


Figure 5.50: Latency findings for 10 minutes of the S4B T3 between the i12 and RPT. Blue line represents UL and black line is the DL.

Table 5.23: S4B latency observations from the three tests.

S4B Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	50	28	54	27.81	104	55.81
T2	31	26.29	30	24.19	56.29	25.1
T3	43	27.02	34	24.55	77	51.57

Packet Loss during S4B

Packets were lost for DL connection during T1. Output from S4B T1 is shown in Figure 5.51. Y-axis represents the loss in percentage.

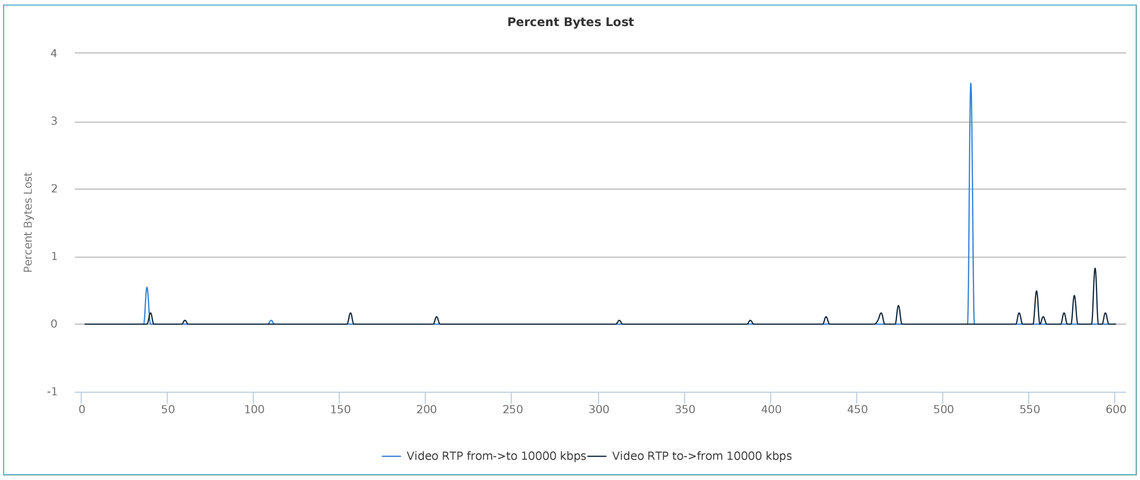


Figure 5.51: Packet loss during S4B T1.

Table 5.24: Percentage packet loss from three S4B tests.

Packet Loss S4B		
Test	UL (%)	DL (%)
T1	0.01	0.01
T2	0	0.01
T3	0	0.02

5.6.3 ST test

Findings of the three STs between i12 and RPT are presented in Table 5.25. This Table shows the Average throughput available between the two endpoints over three different periods.

Table 5.25: Average throughput from three STs.

Throughput ST		
Test	UL (kbps)	DL (kbps)
T1	52221	91054
T2	41117	90249
T3	41043	90830

5.6.4 KPI Findings

The end-to-end communication link between i12 and RPT was monitored for the outdoor scenario, and measurements resulted in the KPIs presented in Table 5.26.

The average OWD for the N-KPI tests was measured to be 21.98 ms for UL and 18.62 for DL. The maximum OWD for UL and DL was measured to be 29 ms and 27 ms, respectively. This resulted in an average RTT of 40.60 ms and a maximum of 56 ms. The overall reliability of the UL/DL links was calculated to be 100%/99.997%.

For the S4B tests, the average OWD was 27.10 ms for UL and 25.52 ms for DL. The maximum OWD was 50 ms for UL and 54 ms for DL. This concludes an average RTT of 52.62 ms and a maximum of 104 ms. The reliability of the link during the S4B tests was measured to be 99.997%/99.987% for UL/DL.

The average available throughput between the iPhone 12 and RPT for the outdoor scenario was 44.79 Mbps for UL and 90.71 Mbps for DL.

Table 5.26: Summary of the KPI's collected for the i12-RPT test scenario.

KPIs										
	OWD N-KPI (ms)		**OWD S4B (ms)**		**Throughput ST (kbps)**		**Reliability N-KPI (%)**		**Reliability S4B (%)**	
Test	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL
1	21.97	18.31	28	27.81	52221	91054	100	100	99.99	99.99
2	21.81	18.84	26.29	24.19	41117	90249	100	100	100	99.99
3	22.16	18.7	27.02	24.55	41043	90830	100	99.99	100	99.98
Avg.	**21.98**	**18.62**	**27.10**	**25.52**	**44794**	**90711**	**100**	**99.997**	**99.997**	**99.987**

5.6.5 Discussion

For the N-KPI tests of the outdoor scenarios, the latency measurements showed similar results to the indoor scenarios (both the original and the retest). However, this was not the case when comparing the original S4B tests for the indoor scenario to the outdoor scenario. Nevertheless, almost the same results were observed between the indoor retest and the outdoor scenario, 54.93 ms average RTT for the indoor retest and 52.62 ms for the outdoor scenario. In addition to this, both the indoor and the outdoor scenario presents similar to the RPD - RPT scenario for the N-KPI tests. This is also the case for the S4B test. This indicates consistency regarding the transmission delays between the two locations.

The reliability of the outdoor N-KPI tests was measured to be 100% for UL and 99.997% for DL. This is equal to the RPD - RPT 5G scenario and similar to the indoor scenario. However, the retest's S4B tests of the indoor scenario showed reliability of 100% for the UL and 99.993% for DL, indicating that the outdoor scenario did not necessarily improve the reliability of the communication link.

The throughput was measured to be 44.79 Mbps for UL and 90.71 Mbps for DL for this scenario. This supports the theory that the RPT LAN limits the RPT to transmit more than 90 Mbps (DL for the iPhone 12). Furthermore, the throughput is similar to the two other scenarios regarding the TDC 700 MHz network for both UL and DL.

These results indicate that the RPD LAN - RPT LAN scenario outperforms the 700 MHz 5G network scenarios regarding latency, reliability, and throughput.

5.7 Scenario 6: iPhone 12 5G - RPD LAN

This section present findings from the sixth test scenario described in section 4.2.5. Section 5.7.1, section 5.7.2 and section 5.7.3 present findings from the N-KPI, S4B and ST tests respectively. Section 5.7.4 summarizes the test scenario by presenting the resulting KPIs. Section 5.7.5 discusses and evaluates the results of these findings.

5.7.1 N-KPI Test

The latency measurements from the three N-KPI tests for this scenario is given in Figure 5.52 for T1, Figure 5.53 for T2 and Figure 5.54 for T3. These figures show the One Way Delay (OWD) for UL (i12 to RPD) and DL (RPD to i12). For all tests for this scenario, i12 is represented as the ”from” probe, and RPD is represented as the ”to” probe. Essential latency metrics from these three tests are represented in Table 5.27. This Table shows maximum (Max) and average (Avg) latency for the UL, DL, and Round-Trip Time (RTT).

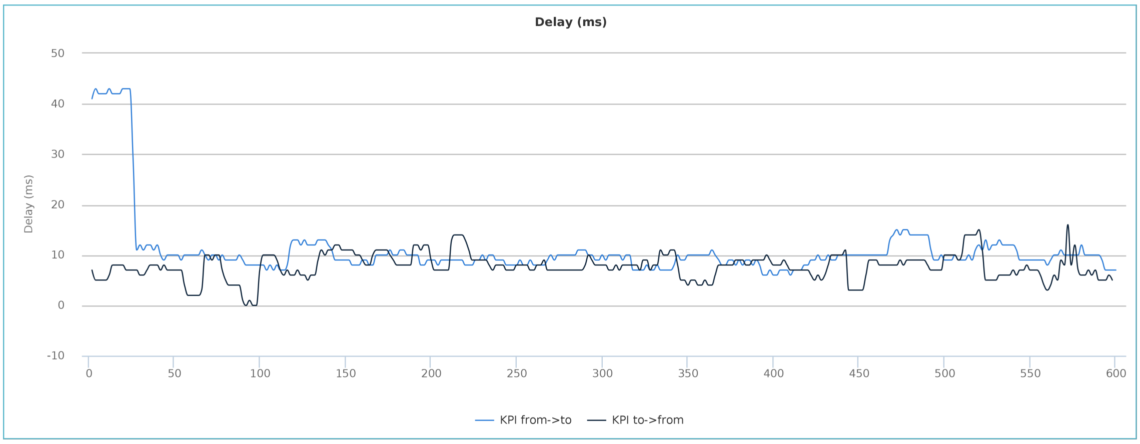


Figure 5.52: Latency findings for 10 minutes of the N-KPI T1 between the i12 and RPD. Blue line represents UL and black line is the DL.

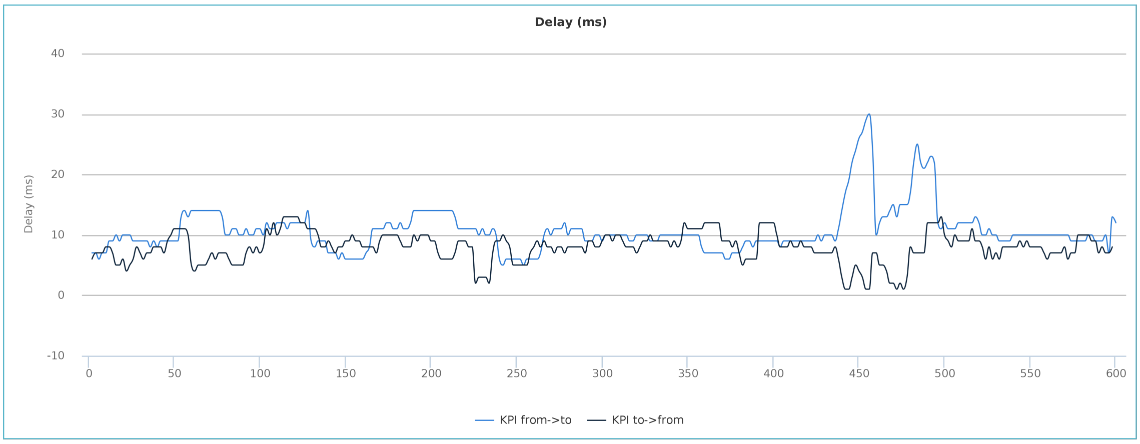


Figure 5.53: Latency findings for 10 minutes of the N-KPI T2 between the i12 and RPD. Blue line represents UL and black line is the DL.

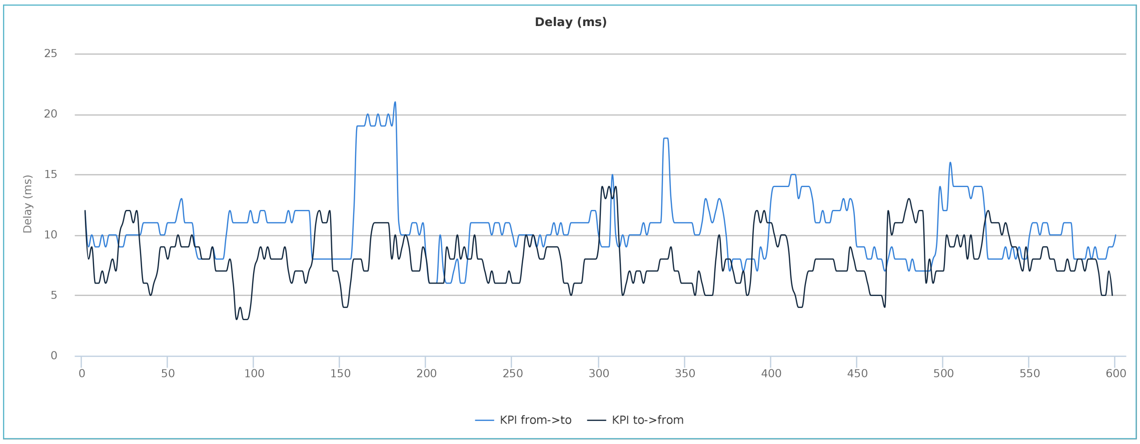


Figure 5.54: Latency findings for 10 minutes of the N-KPI T3 between the i12 and RPD. Blue line represents UL and black line is the DL.

Table 5.27: N-KPI latency observations from the three tests.

N-KPI Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	43	10.94	16	7.7	59	18.64
T2	30	10.76	13	7.93	43	18.69
T3	21	10.55	14	8	35	18.55

Packet Loss during N-KPI
Packet loss was observed for the N-KPI T1 and T2 for DL, and T1 loss is presented in Figure 5.55. Y-axis represents the loss in percentage. The packet loss for DL peaked at 1% three times. No packet loss happened for T3. This is summarized in Table 5.28.

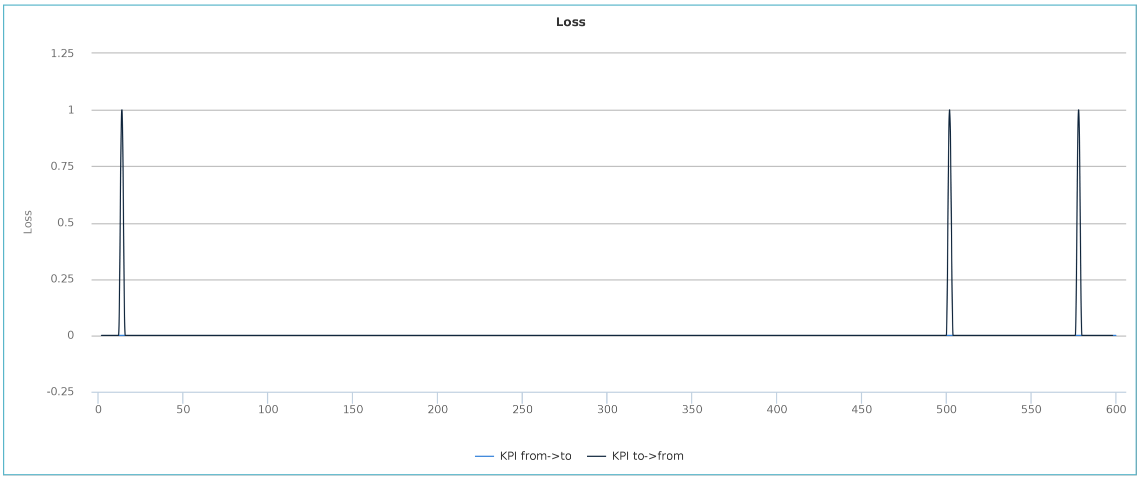


Figure 5.55: The packet loss from N-KPI T3.

Table 5.28: Percentage packet loss from three N-KPI tests.

Packet Loss N-KPI		
Test	UL (%)	DL (%)
T1	0	0.01
T2	0	0.01
T3	0	0

5.7.2 S4B Test

A 10 Mbps video-stream was sent both ways between the two endpoints. The streaming test from S4B T1 is represented in Figure 5.56.

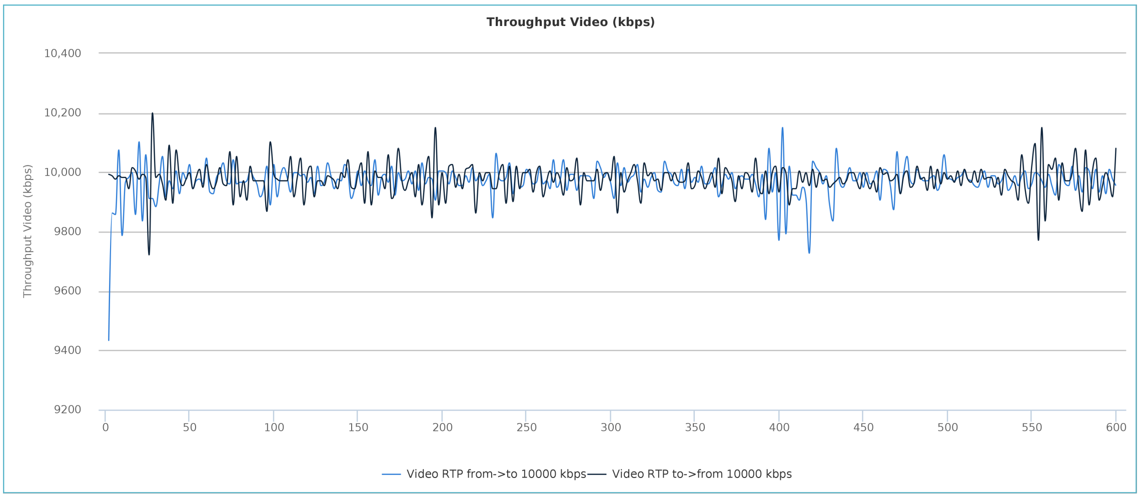


Figure 5.56: The bidirectional streaming service between i12 and RPD for S4B T1.

The OWD of S4B T1, T2 and T3 are presented in Figure 5.57, Figure 5.58 and Figure 5.59 respectively. Essential latency metrics from these three tests are represented in Table 5.29.

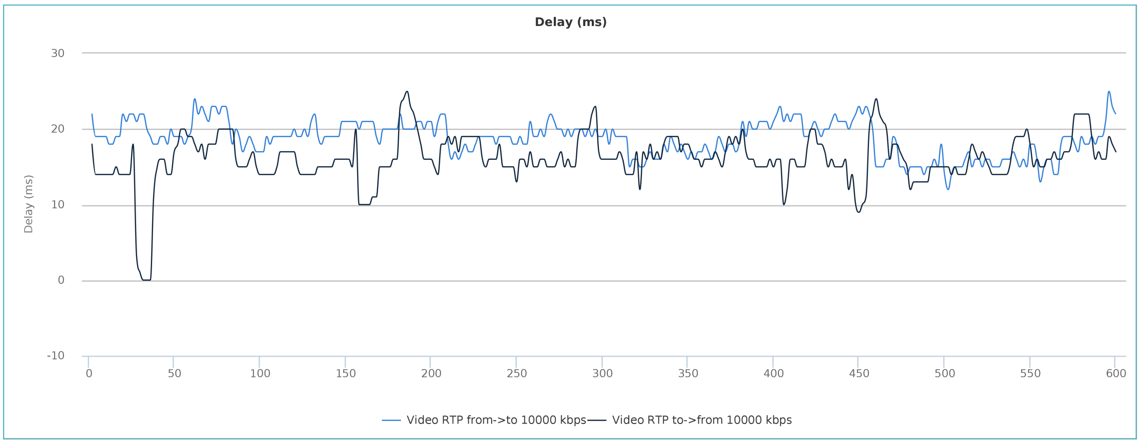


Figure 5.57: Latency findings for 10 minutes of the S4B T1 between the i12 and RPD. Blue line represents UL and black line is the DL.

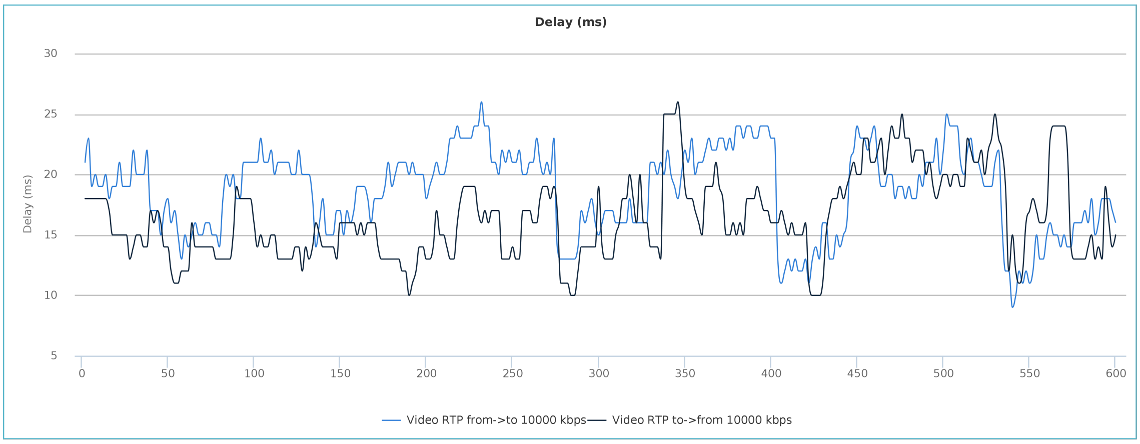


Figure 5.58: Latency findings for 10 minutes of the S4B T2 between the i12 and RPD. Blue line represents UL and black line is the DL.

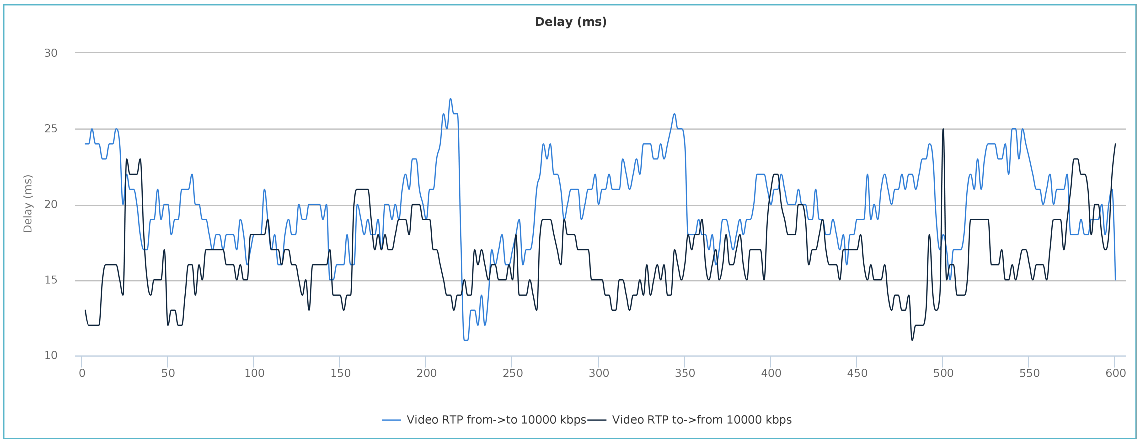


Figure 5.59: Latency findings for 10 minutes of the S4B T3 between the i12 and RPD. Blue line represents UL and black line is the DL.

Table 5.29: S4B latency observations from the three tests.

S4B Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	25	18.73	25	16.07	50	34.8
T2	26	18.64	26	16.44	44.64	17.74
T3	27	19.95	25	16.47	52	36.42

Packet Loss during S4B

Packets were lost during T1, T2, and T3. Packet loss from S4B T3 is shown in Figure 5.60. Y-axis represents the loss in percentage.

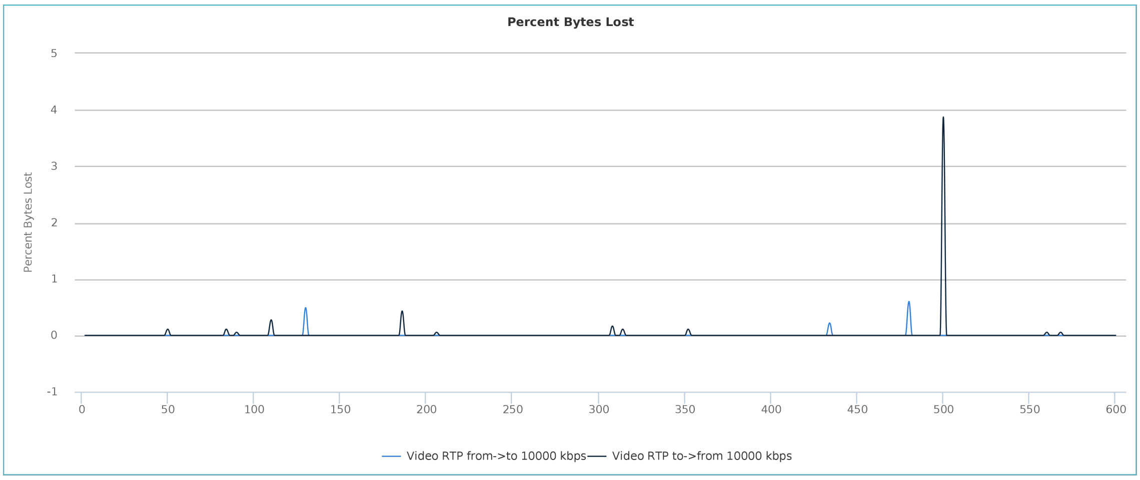


Figure 5.60: Packet loss during S4B T3.

Table 5.30: Percentage packet loss from three S4B tests.

Packet Loss S4B		
Test	UL (%)	DL (%)
T1	0.01	0.01
T2	0.005	0.02
T3	0.005	0.02

5.7.3 ST test

Findings of the three STs between i12 and RPD are presented in Table 5.31. This Table shows the Average throughput available between the two endpoints over three different periods.

Table 5.31: Average throughput from three STs.

Throughput ST		
Test	UL (kbps)	DL (kbps)
T1	34077	257586
T2	46210	214820
T3	45536	211862

5.7.4 KPI Findings

The communication link between i12 and RPD was monitored, and measurements resulted in the KPIs presented in Table 5.32.

For the N-KPI tests, an average OWD of 10.75 ms and 7.88 ms was measured for the UL and DL, respectively. This resulted in an average OWD at 18.63 ms. In addition, reliability of 100% for UL and 99.993% for DL was measured.

For the S4B tests, the iPhone 12 - RPD scenario measured an OWD latency of 19.11 ms for UL and 16.33 ms for DL, which concluded a 35.43 ms RTT. It was measured a 99.993% reliability for UL and 99.983% for DL.

The Hawkeye Speedtests measured an average throughput of 41.94 Mbps for UL and 228.08 Mbps for DL for the communication link.

Table 5.32: Summary of the KPI's collected for the i12-RPD test scenario.

KPIs										
	OWD N-KPI (ms)		**OWD S4B (ms)**		**Throughput ST (kbps)**		**Reliability N-KPI (%)**		**Reliability S4B (%)**	
Test	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL
1	10.94	7.7	18.73	16.07	34077	257586	100	99.99	99.99	99.99
2	10.76	7.93	18.64	16.44	46210	214820	100	99.99	99.995	99.98
3	10.55	8	19.95	16.47	45536	211862	100	100	99.995	99.98
Avg.	**10.75**	**7.88**	**19.11**	**16.33**	**41941**	**228089**	**100**	**99.993**	**99.993**	**99.983**

5.7.5 Discussion

For the N-KPI tests, the average RTT for the RPR - RPT scenario at 28.23 ms is a 52% increase from the average RTT observed in this scenario. In addition, the iPhone 12 - RPD OWD shows a 16% increase in latency from the RPD LAN - RPT LAN scenario. Nevertheless, the maximum measured RTT was 59 ms which, compared to the 40 ms for the RPR - RPT scenario, was an almost 50% increase in RTT. In addition, reliability of 100% for UL and 99.993% for DL was observed, where the 3500 MHz scenario achieved a 100%/100% for UL/DL.

Compared to the 39.04 ms achieved for the RPR - RPT scenario, this was a 9.2% decrease in latency. In addition, Max RTT observed was 53 ms compared to 58 ms for the RPR - RPT scenario. The reliability of the S4B tests also showed a decrease in the reliability of the communication link compared to the RPR - RPT scenario. It was measured a 99.993% reliability for UL and 99.983% for DL, where the RPR - RPT scenario resulted in a 99.997%/99.993% for UL/DL for the same tests.

This scenario was the only one where RPT was not the *To* endpoint. For this scenario, the DL throughput is increased by over 100% when compared to the other scenarios. This further supports the theory that the Raspberry Pi on the NTNU LAN could not output more than 90 Mbps for the UL.

For this scenario, it was observed three times an OWD around 0 ms for the DL. Two times during the N-KPI T1 and one time during the S4B T1 test. This was also observed for the RPR - RPT scenario.

6 Comparison and Discussion of the Findings

This project work carries out measurements on a Data Transmission Network (DTN) that were established to utilize 5G technology. This DTN was configured for enabling remote control of the KMR iiwa robotic system at NTNU from an operator at DTU. Four endpoints were set up, and traffic was sent and received by these endpoints. Measurements were captured by Hawkeye, a networking analysis tool from Keysight. Three different Hawkeye Test types were used; Network KPI, Skype4B, and Speedtest. The Hawkeye Test types captured data which were interpreted and used for founding the Key Performance Indexes (KPIs), discussed in section 4.2.1.

This section considers a discussion of the Cyclictest, and a significance evaluation of the networking tests. Three kernels were tested using Cyclictest, and the results are discussed in section 6.1. A comparison of the findings of the test cases is presented in section 6.2. Section 6.3 presents the value of the results with regards to the industry 4.0 and 5G.

6.1 Cyclictest

The three Raspberry Pis was configured using the 5.4.93-rt51 kernel. This was done with regards to the results in section 5.1. The results showed a lower median latency for the 5.4.93-rt51, when compared to the two other kernels. The 4.19 kernel showed an increase in median latency for core4 compared to the other cores, and similar for 5.10 kernel, where the core1 experienced the increase. Yet, all the kernels are averagely experiencing the same core overhead of around 16-18 μs.

The results can conclude that the Raspberry Pis' kernels would not be the cause of high latency spikes in the networking test scenarios. This is because, the latency is measured in milliseconds, where the kernel shows microseconds delays, which is $10^{-3} ms$.

6.2 Comparison of Network Tests

This section presents a comparison between all the tests and the KPI values of each test scenario. It will present and compare findings in tables from the three test types; Network KPI, Skype4B, and Speedtest, regarding the given test scenarios; 1 to 6. These six scenarios is listed section 4.2 Table 4.3. Section 6.2.1, section 6.2.2 and section 6.2.3 presents the comparisons of the three tests Network KPI, Skype4B and Speedtest respectively.

6.2.1 Network KPI

The N-KPI test evaluated the latency and loss during a small load on the network (100 kbps). The latency of the N-KPI tests for DL/UL for the six scenarios is presented in Table 6.1. This table shows the average OWD from T1, T2, and T3 for all test scenarios. It also shows the average OWD, average RTT, maximum OWD, and maximum RTT for T1, T2, and T3 in total for all scenarios.

The reliability of the N-KPI tests was calculated based upon the loss that happened throughout the tests. This is presented for DL/UL for all scenarios in Table 6.2. This table shows the reliability of each test and the average of all tests.

Table 6.1: Comparison of the latency during the three N-KPI tests for each test scenario.

Latency Network KPI (ms)

	RPR 5G - RPT LAN		*RPD 5G - RPT LAN*		*RPD LAN - RPT LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	16.61	11.82	22.66	17.79	8.00	8.00
T2	15.85	12.54	26.90	20.17	8.00	8.00
T3	16.04	11.82	23.15	21.50	8.00	8.00
Avg OWD	**16.17**	**12.06**	**24.24**	**19.82**	**8.00**	**8.00**
Avg RTT	**28.23**		**44.06**		**16.00**	
Max OWD	**23.00**	**17.00**	**30.00**	**25.00**	**9.00**	**9.00**
Max RTT	**40.00**		**55.00**		**18.00**	

	1st: i12 5G - RPT LAN		*2nd: i12 5G - RPT LAN*		*i12 5G - RPD LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	22.04	18.17	21.97	18.31	10.94	7.70
T2	21.95	18.87	21.81	18.84	10.76	7.93
T3	21.84	19.43	22.16	18.70	10.55	8.00
Avg OWD	**21.94**	**18.82**	**21.98**	**18.62**	**10.75**	**7.88**
Avg RTT	**40.77**		**40.60**		**18.63**	
Max OWD	**31.00**	**27.00**	**29.00**	**27.00**	**43.00**	**16.00**
Max RTT	**58.00**		**56.00**		**59.00**	

Table 6.2: Comparison of the reliability during the three KPI tests for each test scenario.

Reliability - Network KPI (%)

	RPR 5G - RPT LAN		*RPD 5G - RPT LAN*		*RPD LAN - RPT LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	100	100	100	99.99	100	100
T2	100	100	100	100	100	100
T3	100	100	100	100	100	100
Avg	**100**	**100**	**100**	**99.997**	**100**	**100**

	1st: i12 5G - RPT LAN		*2nd: i12 5G - RPT LAN*		*i12 5G - RPD LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	100	100	100	100	100	99.99
T2	100	100	100	100	100	99.99
T3	100	99.98	100	99.99	100	100
Avg	**100**	**99.993**	**100**	**99.997**	**100**	**99.993**

6.2.2 Skype4B

The Skype4B tests evaluated the latency and loss during a video streaming scenario of 10 Mbps. The latency of the S4B tests for DL/UL for all the six scenarios is presented in Table 6.3. This table shows the average OWD from T1, T2, and T3 for all test scenarios. It also shows the average OWD, average RTT, maximum OWD, and maximum RTT for T1, T2, and T3 in total for all scenarios.

The reliability of the Skype4B is presented for DL/UL for all scenarios in Table 6.4. This table shows the reliability of each test and the average of all tests.

Table 6.3: Comparison of the latency during the three Skype4B tests for each test scenario.

Latency Skype4B (ms)						
	RPR 5G - RPT LAN		*RPD 5G - RPT LAN*		*RPD LAN - RPT LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	19.78	18.73	29.49	26.59	8.94	8.94
T2	20.11	19.01	26.15	27.70	8.94	8.94
T3	20.57	18.91	27.65	29.89	8.94	8.94
Avg OWD	**20.15**	**18.88**	**27.76**	**28.06**	**8.94**	**8.94**
Avg RTT	**39.04**		**55.82**		**17.88**	
Max OWD	**28.00**	**30.00**	**32.00**	**65.00**	**9.00**	**9.00**
Max RTT	**58.00**		**97.00**		**18.00**	
	1st: i12 5G - RPT LAN		*2nd: i12 5G - RPT LAN*		*i12 5G - RPD LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	27.21	25.47	28.00	27.81	18.73	16.07
T2	370.43	347.06	26.29	24.19	18.64	16.44
T3	138.47	121.21	27.02	24.55	19.95	16.47
Avg OWD	**178.70**	**164.58**	**27.10**	**25.52**	**19.11**	**16.33**
Avg RTT	**343.28**		**52.62**		**35.43**	
Max OWD	**866.00**	**913.00**	**50.00**	**54.00**	**27.00**	**26.00**
Max RTT	**1779.00**		**104.00**		**53.00**	

Table 6.4: Comparison of the reliability during the three Skype4B tests for each test scenario.

Reliability - Skype4B (%)

	RPR 5G - RPT LAN		*RPD 5G - RPT LAN*		*RPD LAN - RPT LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	99.99	99.98	100	99.94	100	99.99
T2	100	100	100	99.89	100	100
T3	100	100	100	99.97	100	100
Avg	**99.997**	**99.993**	**100**	**99.933**	**100**	**99.997**

	1st: i12 5G - RPT LAN		*2nd: i12 5G - RPT LAN*		*i12 5G - RPD LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	99.99	99.99	99.99	99.99	99.99	99.99
T2	63.61	99.97	100	99.99	99.995	99.98
T3	89.50	99.98	100	99.98	99.995	99.98
Avg	**84.367**	**99.980**	**99.997**	**99.987**	**99.993**	**99.983**

6.2.3 Speedtest

The Speedtests evaluated the available throughput between the two endpoints. The throughput for UL/DL for all test scenarios is presented in Table 6.5. This table shows the average throughput for each test and the average throughput of all tests.

Table 6.5: Comparison of the throughput during the three Speedtests for each test scenario.

Throughput - Speedtest (kbps)

	RPR 5G - RPT LAN		*RPD 5G - RPT LAN*		*RPD LAN - RPT LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	79165	90755	50742	89407	92253	90867
T2	78322	89856	56827	84052	92346	90830
T3	70383	89800	53981	87384	85849	90979
Avg	**75957**	**90137**	**53850**	**86948**	**90149**	**90892**

	1st: i12 5G - RPT LAN		*2nd: i12 5G - RPT LAN*		*i12 5G - RPD LAN*	
Test	UL	DL	UL	DL	UL	DL
T1	18649	47989	52221	91054	34077	257586
T2	17712	90024	41117	90249	46210	214820
T3	12451	83958	41043	90830	45536	211862
Avg	**16271**	**73990**	**44794**	**90711**	**41941**	**228089**

6.3 Significance of the Network Findings

This thesis handles work with two different 5G NSA commercial networks. The possibility of remote control is valuable for the robotic industry. 5G promised to deliver higher speeds, lower latency, and higher reliability. As discussed in the paper [29], the QoS for Industry 4.0 is listed in Table 6.6. Different requirements are listed based on the usage. For instant, the AR latency requirement is 10 ms, and motion control latency is 0.25-1 ms. The reliability is presented such that 1e-5 means that the reliability should be a minimum of 99.999%, also called the five-nine reliability. Table 6.6 is complicated to fulfill without a Standalone network, but it provides aims for wireless communication technology. The findings from the test scenarios are an evaluation of the system setup used for this thesis, and Table 6.6 can provide a comparison for the measured KPIs.

Table 6.6: Quality of Service for Industry 4.0 [29].

	Motion Control	**Condition Monitoring**	**Augmented Reality (AR)**
RTT	0.25-1 ms	100 ms	10 ms
Packet loss	1e-8	1e-5	1e-5
Data Rate	kbit/s - Mbit/s	kbit/s	Mbit/s - Gbit/s

Now all scenarios are referred to as scenario 1-6, as displayed in Table 4.1, and also shown from left to right in the tables in section 6.2. Results from scenario 1 show an improvement in the Wi-Fi network used for the specialization project. Furthermore, the results validate the scenario when performing non-time-sensitive tasks, like condition monitoring, with regards to Table 6.6. This is because the average RTT was 28.23 ms for N-KPI tests, which is more than what is required for Motion Control and AR in the table. This was the case for all scenarios, but scenario 3 is closest to the 10 ms mark for AR, with 16 ms RTT.

When comparing the reliability in Table 6.6 to the results from scenarios 1 to 6, the N-KPI tests showed better results than the S4B tests. For the N-KPI tests, the UL of all scenarios showed a 100% reliability, which validates all use cases in Table 6.6. Nevertheless, this was not the case for DL, where only scenarios 1 and 3 met reliability requirements. For the S4B tests, only scenario 2 UL and scenario 3 UL met the requirements for reliability.

The S4B tests for the scenario 4 retests of the indoor scenario measured a reliability of 100% for the UL and 99.993% for DL. Compared to the scenario 5 where the iPhone was outdoors, which measured at 99.99% for UL and 99.987% for DL, this indicates that the outdoor scenario did not necessarily improve the reliability. Yet, it was measured an increase in the throughput for UL, compared to the indoor test and retest scenarios.

All scenarios showed Mbit/s of data rate capabilities for throughput, which meets the requirements for all use cases. Although the data rate for AR can require Gbit/s traffic, which none scenarios achieved, scenario 6 resulted in a DL throughput of 0.228 Gbit/s. Also, the higher the throughput of the AR stream is, the more details of the AR can be transported to the operator, which can help an industrial company.

Overall this concludes that according to Table 6.6, only scenarios 1 and 3 would be capable of meeting all the requirements from the condition monitoring use case. These results indicates that the LAN to LAN scenario outperforms the 5G network scenarios, with regards to latency and reliability. It also showed greater stability with the LAN to LAN scenario, because of the less oscillating behavior. Nevertheless, no scenarios would be accepted for Motion Control or Augmented Reality according to this table, but according to [29], to meet all these requirements was not possible with the wireless communication technology in 2017.

7 Discussion of the Experimental Approach

This section will evaluate the experimental approach. It focuses on how the telecommunication setup and testing contributed to the goal of this thesis. Section 7.1 evaluates the setup, meaning how it was designed and configured to represent a scenario related to the industry and which improvements could have been made to the setup. Section 7.2 handles the testing scenarios and how valid the test results are when evaluating the whole system.

7.1 Telecommunication Setup

The telecommunication setup represented a remote control scenario of a robotic system (KMR iiwa) located at NTNU in Trondheim. Three Raspberry Pis (RPis) and one iPhone 12 represented the test setup. The RPis were all configured to have real-time capabilities, with a real-time patch applied and using the ROS2's real-time publish-subscribe protocol for the RPR. Nevertheless, the iPhone 12 was not configured for a real-time use case. Few differences were observed regarding the difference in KPIs between the iPhone 12 and a Raspberry Pi on the same 5G network. The Raspberry Pi achieved a 20 % higher throughput for the uplink communication, indicating that the antennas of the 5G HAT were better than the iPhone's. The new chips from Qualcomm, either x60 or x65, that is expected to be introduced in the iPhone 13, could introduce URLLC capabilities to the iPhone. Nevertheless, the new chips from Qualcomm could also be introduced in new 5G HATs.

The setup was configured such that the RPT act as a HUB, where new robotic systems could connect, thus making it easy to connect to multiple robots, as a typical robotic plant would be. This scalable setup can be configured by simply connecting a new Raspberry Pi with a 5G HAT to a new robot. The scalability also facilities remote access to the robotic network. A remote operator could access the systems from a phone or a workstation. The network makes it an adaptable setup, which can be used for multiple robotic use cases. Although, this thesis does not focus on the security issues related to connecting a robotic system to the network. Hackers could access the robotic system if end-to-end encryption is not established. This issue becomes more significant when the end-to-end communication is over a larger distance and connected to a commercial network. Moreover, the greater the distance, the more routing, queuing, and propagation delay will occur. A private 5G network would make it more difficult to hack.

The 5G networks used for the setup were two commercial 5G networks from Telenor and TDC. They were both Non-Standalone (NSA) networks, meaning they rely on the 4G infrastructure. The Standalone (SA) 5G networks can achieve lower latency than the NSA because of the new 5G Core Network. The deployment of the 5G SA network at NTNU was delayed, and therefore it was decided to use the commercial NSA network. As this project did not test a 5G to 5G connection, there will be differences between the findings of this project and the expectations of such a 5G communication network. The 5G to 5G communication also includes fewer hops between the endpoints. Nevertheless, robotic industries' expectations regarding industry 4.0 and IIoT will probably have to wait longer for standardized and commercial deployment of 5G Standalone URLLC solutions for remote robotic operations. This is because the latest release from 3GPP included the standardization of IIoT and URLLC, which has been a subject for robotic applications for many years.

One of the issues with the network being a commercial network is that as a user, one can not control the traffic on the broadband. This can impact the robotic system, suddenly not reacting because an overload on the network has occurred. With a private network, one can set up more routers for better coverage, where, e.g., trucks blocking the signal between the robot and the router. In addition, another router can transmit from a different angle. This can prevent crucial packet loss during operation.

In addition to the current telecommunication setup, it could be beneficial to install surveillance cameras at the lab, such that the operator could see what is happening if, e.g., the robotic system is not responding. Also, setting up an experimental setup from another country makes installation, configuration, and services more difficult because the remote operator needs someone at the lab to help with all this. This can also help for scheduling work as such that other work is not interfering with either reliability, latency, or throughput bidirectionally between robot and operator.

7.2 Test Executions

The telecommunication network was evaluated with the Hawkeye internet troubleshooting tool. There was set up a Hawkeye Server that all the endpoints connected to for tests to be executed between them. Hawkeye does not evaluate where the latency occurs in the system; therefore, a communication link between DTU and NTNU can be hard to evaluate with Hawkeye unless the packets get transferred between only Hawkeye Endpoints. Also, the software endpoints of the Hawkeye system are limited compared to the Hardware Endpoints that Ixia sells. However, they have more functions and allows for tracing the packets between the endpoints. Similarly, Hawkeye is limited when evaluating whether the source endpoint or destination endpoint induces latency. It also reports the latency with only milliseconds accuracy and packet loss with 0.01% accuracy. Hawkeye synchronizes the endpoints with UDP packets, and the Hawkeye Server acts as an NTP server. The Hawkeye User Guide manual states that the synchronization can be anywhere between a few milliseconds to tens of milliseconds. This indicates that the synchronization of the test scenarios could be inaccurate by tens of milliseconds. This is critical for time-sensitive tasks, as a robotic system is.

Because the Raspberry Pis were set up equally, it could introduce the hypothesis that the Application delay and Serialization delay would be equal for the three RPis. Nevertheless, one of the RPis was located in the Manulab, where other industrial operations occur, such as welding, which can interfere with the radio waves. This also indicates that the tests between RPR and RPT were conducted without ensuring that nothing is interfering with the test at the lab. Furthermore, commercial networks provide uncertainties when it comes to interference. This can be seen from the test results, which indicates that the results from a given test can have immense variations compared to the next test.

The actual test types represented industrial robotic scenarios, where the Network KPI test could be an indication for how the system will perform under a motion control or condition monitoring operation, see Table 6.6. Similarly, the Skype4B test evaluated the communication link during a 10 Mbps stream sent bidirectionally between the endpoints. The difference between these tests showed a more considerable increase in latency when the Skype4B test ran than the Network KPI tests. This shows the importance of testing the communication link with a realistic load. Using the Internet Control Message Protocol (ICMP), a ping usually only transmits 512 bits/s, almost 200 times less than the 100 Kbps the Network KPI test is sending. Sensor readings, and other critical information about a robotic system, are generally more data than 512 bits. Nevertheless, it would be easier to pass a 512 bits/s data stream end-to-end than a 10 Mbps stream when considering a commercial network.

For this experimental approach, the Raspberry Pis were set up, using real-time patches to make Linux act as a Real-Time Operating System (RTOS). This was done to curtail the latency of the kernel. As discussed in section 2.2, all packets go through the stack, where different types of delay occur. An improvement to the test execution for the scenarios is to follow the packets through the networking stack and measure the latency for each level. This would strengthen the results to evaluate, e.g., that the CPU is not a bottleneck for the measured RTT between the endpoints. Furthermore, the routing and queuing delay would impact the system for remote control operations, where the packets travel through several routers and switches. Therefore, it would be beneficial to trace the packets to observe if a router is a cause for most of the measured RTT.

The results from this thesis showed that the Skype4B streaming test was validated for both Motion Control, Condition Monitoring, and AR, according to Table 6.6. Together with reliability and latency, only Condition Monitoring was realistic, primarily because of the delay demands for AR and Motion Control. If one were to improve data collection, the tests should be executed over a longer time. This project did not do this because sending many packets over a commercial cellular network requires economic support. A single Skype4B test sent over 6 Gb of data, and if this test were to run for 24 hours, this would be 864 Gb of data, and that is for only one scenario. This is around ten times more data than a standard monthly subscription. Nevertheless, more data from longer tests would provide more accurate findings. It is not unlikely for an industrial plant to run non-stop throughout a day, so if the tests ran for 24 hours, this will lead to an overview of the trends in the cellular network. A private cellular network could also support this overview, with the way it can be configured and monitored. For example, connecting a new robotic system, that needs a significant bandwidth to a cellular grid, can cause overload on the grid, which can make industrial systems unstable, which furthermore can lead to shut downs, and/or dangerous situations. Therefore a private network would be preferable, so things can be connected in a controllable manner.

Conclusion

PART 4

8 Conclusion

This section concludes the conducted thesis work. Section 8.1 considers suggestions for improvement to the system setup, and how to move forward with the work. Section 8.2 summarizes the thesis work, with a presentation of what was achieved, and the conclusions.

8.1 Further Work

This thesis evaluated two commercial Non-Standalone 5G networks. It is expected that a Standalone 5G network could lower the latency further than what was experienced in this work. So a suggestion for further work would be to investigate a 5G SA network, which can be compared and evaluated against the results shown in this thesis.

Furthermore, this thesis did not evaluate a 5G to 5G only communication link. The design of the system setup is configured to support such communication, and it will be beneficial to test this way of communication too. It can provide findings considering the 5G IoT industry, where instruments have end-to-end communication over a local 5G network. Such experimental results can also present findings for comparison with the requirements from Industry 4.0. In combination with this, also testing out new generations of 5G-enabled smartphones and Single-Board Computers as the Raspberry Pi, could introduce valuable results around 5G technology. The new generations of smartphones will likely be equiped with newer generations of Qualcomm chips, also meant for URLLC, and in the future, rumors says that Apple will make their own chip for cellular communication.

It would be valuable to evaluate a mesh network of multiple robotic systems connected to the same 5G network. These results can provide an understanding of how the bandwidth is affected by all the connected systems. Given that the robotic systems is mobile, the results can measure how the coverage area of the 5G network test scenario is concerned with robotic systems moving around. KPIs such as reliability would be interesting because the robotic systems can block each other's communication with the 5G cell. Furthermore, this provides opportunities around setting up a laboratory setup such that the reliability requirements for industry 4.0 are met with, e.g., implementing more 5G cells to cover different transmission angles. This can further be expanded to a remote scenario where multiple users want to capture the data from multiple robotic systems at the same time, over 5G.

The thesis work did not consider encryption of the communication link. An encrypted link would provide more security, which is needed for keeping the remote control scenario unavailable for hackers. It could be fatal if someone were to interfere with a robotic system. Implementing such security measures is essential, but the measures could also induce more latency and lower throughput to the system setup, which is undesirable.

To make the setup more user-friendly, it can be favorable to set up an API that can be used to access all the robots on the network. This can be an application that can be installed on a phone, tablet, or laptop. It would be helpful if the application can configure, control and retrieve important information, such as a notification when a system has failed or similar. If this application is configured such that security is handled, this will make the robotic systems accessible from anywhere, which is an ambition of many companies.

The test scenarios of this thesis were evaluated with the Hawkeye networking analysis tool from Keysight. It can be beneficial to conduct similar experiments with another method of capturing the data. For example, iPerf3 is a popular tool to measure network behavior and is commonly used by many researchers. Furthermore, Hawkeye is synchronizing the endpoints similarly to NTP synchronization. Another synchronization method that can improve accuracy such as PTP, can be desired for a given robotic application. This can also evaluate if the synchronization method of Hawkeye itself induced significant variability of latency measurements in the system.

8.2 Concluding Remarks

The main objective of this thesis was to establish a remote control communication network with 5G and evaluate Key Performance Indexes (KPIs) for this setup. Three Raspberry Pi 4Bs with 5G HATs and one iPhone 12 defined four endpoints. With these four endpoints, six test scenarios were composed to evaluate the 5G networks. Two 5G networks were utilized for the test scenarios; TDC's 700 MHz 5G network and Telenor's 3500 MHz 5G Network. The two networks were Non-Standalone commercial networks.

This thesis provided a theoretical overview of the telecommunication equipment, and from this a unique system setup for tests was designed and configured. Which worked as intended. Packets reached the desired destinations during test executions. The test scenarios used 5G communication, but not with a 5G to 5G configuration, meaning that both endpoints did not communicate over 5G at the same time. The 5G NSA networks provided a 5G communication that used one of the surrounding LTE core networks.

The kernels of the Raspberry Pis were configured with regards to the results from a Cyclictest. This test measured the overhead latency in the kernel operations itself, of the Quad-Core processor in the Raspberry Pi 4Bs. The results showed a kernel latency in microseconds from the three kernels that were tested. This implies that the latency of the kernel would not significantly impact the total latency of the conducted network test scenarios.

A networking analysis tool was used systematically for measurements of the KPIs; latency, throughput, and reliability. Three test types were used, a low-impact KPI test, a streaming test (10 Mbps video-stream), and a throughput test. The results from the tests indicate that most test scenarios only supports a few industrial robotic application, when comparing to the Industry 4.0 requirements. The streaming test displayed how larger throughput affected the network KPIs, where higher latency and more loss were observed.

Few differences were observed regarding the difference in KPIs between the iPhone 12 and the Raspberry Pi on the same 5G network. The Raspberry Pi achieved a 20 % higher throughput for uplink communication.

Measurements showed that the Raspberry Pi connected to the NTNU LAN could not achieved throughput of more than 90 Mbps for the uplink and downlink communication. This was caused by the LAN-switch at NTNU, which are 100-Mbps-switches. This was the case in all test scenarios, where it was not shown results greater than 100 Mbps for UL and DL. The one scenario that excluded the NTNU LAN achieved around 228 Mbps for DL.

This thesis developed a telecommunication test setup that is relevant for further testing and evaluation for industrial use. It is recommended to use this setup with the 5G private Standalone network that is being installed at NTNU. This designed test setup and methodology can be reused for this Standalone network, and findings could be compared to those in this thesis. This will provide results regarding the qualities of the private 5G Standalone network for industrial use.

Bibliography

[1] 3GPP. 3gpp - a global initiative, 2021. URL https://www.3gpp.org/.

[2] D. Toal A. Kaknjo, E. Omerdic. Measurement of network latency in remote presence applications, 2016. URL http://srv.uib.es/public/CAMS2016/media/files/0052.pdf".

[3] Ali T. Al-khwaldeh. Time synchronization for data communication in the wireless network systems, 2015. URL http://www.meacse.org/ijcar/archives/74.pdf.

[4] Mohammed M. Alani. *TCP/IP Model*. Springer International Publisher, 2014.

[5] M. Aleksy, F. Dai, N. Enayati, P. Rost, and G. Pocovi. Utilizing 5G in Industrial Robotic Applications. In *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 278–284, August 2019. doi: 10.1109/FiCloud.2019.00046.

[6] Jin Cao, Pu Yu, Maode Ma, and Weifeng Gao. Fast authentication and data transfer scheme for massive nb-iot devices in 3gpp 5g network. *IEEE Internet of Things Journal*, 6(2):1561–1575, 2018.

[7] Nina Marie Wahl Charlotte Heggem. Mobile navigation and manipulation - configuration and control of the kmr iiwa with ros2, 2020.

[8] Son Chee Loon Cheah Wai Zhao, Jayanand Jegatheesan. Exploring iot application using raspberry pi, 2015. URL https://ijcna.org/Manuscripts/Volume-2/Issue-1/Vol-2-issue-1-M-04.pdf.

[9] Jiangfeng Cheng, Weihai Chen, Fei Tao, and Chun-Liang Lin. Industrial iot in 5g environment towards smart manufacturing, 2018. ISSN 2452-414X. URL http://www.sciencedirect.com/science/article/pii/S2452414X18300049.

[10] Tobias Blaß et al. Daniel Casini. Response-time analysis of ros 2 processing chains under reservation-based scheduling, 2019. URL https://drops.dagstuhl.de/opus/volltexte/2019/10743/pdf/LIPIcs-ECRTS-2019-6.pdf.

[11] Peter L Dordal. *An Introduction to Computer Networks*. Department of Computer Science, Loyola University Chicago, 2021.

[12] Endre Erős, Martin Dahl, Kristofer Bengtsson, Atieh Hanna, and Petter Falkman. A ros2 based communication architecture for control in collaborative and intelligent automation systems, 2019. ISSN 2351-9789. URL http://www.sciencedirect.com/science/article/pii/S2351978920300469. 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing.

[13] Kelechi G Eze, Matthew NO Sadiku, and Sarhan M Musa. 5g wireless technology: A primer. *International Journal of Scientific Engineering and Technology*, 7(7):62–64, 2018.

[14] Alessandra Flammini and Paolo Ferrari. Clock synchronization of distributed, real-time, industrial data acquisition systems, 2010. URL https://www.researchgate.net/publication/221909507_Clock_Synchronization_of_Distributed_Real-Time_Industrial_Data_Acquisition_Systems.

[15] Joseph Gehring. Packet analysis using wireshark, 2011. URL http://itech.fgcu.edu/faculty/zalewski/projects/files/WiresharkReport2011.pdf.

[16] Robert Ingalls. Socket tutorial. URL https://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html.

[17] A Keysight Buisness Ixia. Hawkeye, 2020.

[18] Filip Holik Josef Horalek, Tomas Svoboda. Analysis of the wireless communication latency and its dependency on a data size, 2016.

[19] KUKA. Lbr iiwa, 2014. URL xpert.kuka.com.

[20] KUKA. Kmp 200 omnimove, 2017. URL xpert.kuka.com.

[21] KUKA. Kuka sunrise cabinet, 2019. URL xpert.kuka.com.

[22] KUKA. Kmr iiwa omnimove, 2020. URL xpert.kuka.com.

[23] Jing Li, Zheng Luo, David Ferry, Kunal Agrawal, Christopher Gill, and Chenyang Lu. Global edf scheduling for parallel real-time tasks. *Real-Time Systems*, 51(4):395–439, 2015.

[24] Miroslav Lichvar. Combining ptp with ntp to get the best of both worlds, 2016. URL https://www.redhat.com/en/blog/combining-ptp-ntp-get-best-both-worlds.

[25] SIMCom Wireless Solutions Limited. Sim8200ea-m2 hardware design, 2020.

[26] Frank Mademann. The 5g system architecture. *Journal of ICT Standardization*, 6(3):77–86, 2018.

[27] D.L. Mills. Network time protocol (ntp), 1985. URL https://pdfs.semanticscholar.org/737d/aef6ae01426a1ec2325ecbcd469542136e16.pdf?_ga=2.184068642.588056429.1605446308-2117399736.1605446308.

[28] Patrick Moorhead. How the snapdragon x50, world's first 5g modem, puts qualcomm ahead of the curve. URL https://moorinsightsstrategy.com/how-the-snapdragon-x50-worlds-first-5g-modem-puts-qualcomm-ahead-of-the-curve/.

[29] Jingon Joung Murtaza Ahmed Siddiqi, Heejung Yu. 5g ultra-reliable low-latency communication implementation challenges and operational issues with iot devices, 2019. URL https://www.mdpi.com/2079-9292/8/9/981/pdf-vor.

[30] Subin Narayanan, Dimitris Tsolkas, Nikos Passas, and Lazaros Merakos. Nb-iot: a candidate technology for massive iot in the 5g era. In *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6. IEEE, 2018.

[31] O3b Networks. What is network latency - why does it matter?, 2010. URL http://www.satmagazine.com/story.php?number=1342678280.

[32] Håvard Persson. Remote operations and testing of kmriiwa using raspberry pi 4 with wi-fi, 2020.

[33] Qualcomm. Qualcomm announces new features in snapdragon x65 5g modem-rf system for global 5g expansion, 2021. URL https://www.qualcomm.com/news/releases/2021/05/19/qualcomm-announces-new-features-snapdragon-x65-5g-modem-rf-system-global-5g.

[34] Thomas G. Robertazzi. *Introduction to Networks*, pages 1–16. Springer International Publishing, Cham, 2017. ISBN 978-3-319-53103-8. doi: 10.1007/978-3-319-53103-8_1. URL https://doi.org/10.1007/978-3-319-53103-8_1.

[35] Route-XP. Osi model vs tcp/ip model, 2020. URL https://www.routexp.com/2020/03/osi-model-vs-tcpip-model.html.

[36] seeed.com. Raspberry pi 4 computer module b 4gb, 2021.

[37] 5G Solutions. Explore 5g solutions, 2019. URL https://www.5gsolutionsproject.eu/explore/.

[38] Ramjee Prasad Sriganesh K. Rao. Impact of 5g technologies on industry 4.0, 2018.

[39] William Stallings. Data and computer communications, 2013.

[40] Stephane Teral. 5g best choice architecture. *IHS Markit Technology, London, UK, White Paper*, 2019.

[41] Alan Truly. iphone cellular modem: Can apple really compete with qualcomm on 5g?, 2020. URL https://screenrant.com/iphone-modem-apple-qualcomm-5g-chip/.

[42] F. Voigtländer, A. Ramadan, J. Eichinger, C. Lenz, D. Pensky, and A. Knoll. 5G for Robotics: Ultra-Low Latency Control of Distributed Robotic Systems. In *2017 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, pages 69–72, October 2017. doi: 10.1109/ISCSIC.2017.27.

[43] Waveshare. Sim8200ea-m2 5g hat, 2021. URL https://www.waveshare.com/sim8200ea-m2-5g-hat.htm.

[44] Wikipedia. Udp header, 2019. URL https://no.wikipedia.org/wiki/UDP.

[45] Kittipong Yaovaja and Jakkarin Klunngien. Teleoperation of an industrial robot using a non-standalone 5g mobile network. In *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pages 320–323. IEEE, 2019.

[46] Takuya Azumi Yuya Maruyama, Shinpei Kato. Exploring the performance of ros2, 2016. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7743223.

Appendices

Table of Content

- A - Raspberry Pi 4B Setup
- B - Hawkeye Endpoint Setup
- C - Test Scenario 4 Re-test

A Raspberry Pi 4B Setup

This Appendix describes setting up a Raspberry Pi according to the experimental setup for this thesis. Some requirements need to be available for being able to follow this guide.

Setup Requirements

- A Raspberry Pi 4, Power Supply, an Ubuntu 20.04 computer with a micro SD card reader and a micro SD Card
- A SIM8200EA-M2 5G HAT for Raspberry Pi
- An internet connection

Setup Guide

Step 1: Insert SD Card into a work station and download the Raspberry Pi OS 64-bit image from `https://downloads.raspberrypi.org/raspios_arm64/images/raspios_arm64-2020-05-28/2020-05-27-raspios-buster-arm64.zip`

Step 2: Write the ISO image to the SD card, using the Raspberry Pi Imager software.

Step 3: Preempt-rt Patching: Follow the guide at `https://www.instructables.com/64bit-RT-Kernel-Compilation-for-Raspberry-Pi-4B-/` but download the latest stable RT patch for kernel 5.4 instead in step 3

Step 4: Control the kernel version by writing

```
uname -a
```

The output should say

```
Linux raspberrypi 5.4.83-rt51-v8+ #1 SMP PREEMPT_RT
```

or corresponding to the patch that was downloaded.

Step 5: *This step is only for the Raspberry Pi connected to the AGV (RPR)*:
Set the Ethernet IP address to correspond with the AGV's 172.31.1.x address. This is done by configuring the dhcpcd.conf file:

```
sudo nano /etc/dhcpcd.conf
```

Add the following lines:

```
interface eth0
static ip_address=172.31.1.x/24
```

Save the configuration file
Install Python3:

```
sudo apt install python3
```

Install ROS2 on the Raspberry Pi by following this guide `https://docs.ros.org/en/foxy/Installation/Ubuntu-Development-Setup.html` but before the building create a file

```
nano ~/.colcon/defaults.yaml
```

Insert the following:

```
build:
  cmake-args:
    - -DCMAKE_SHARED_LINKER_FLAGS='-latomic -lpython3.7m'
    - -DCMAKE_EXE_LINKER_FLAGS='-latomic -lpython3.7m'
    - -DCMAKE_BUILD_TYPE=RelWithDebInfo
```

Save the file and continue the guide.

Step 6: Install the SIM8200EA-M2 5G HAT for Raspberry Pi by following the guide at https://www.waveshare.com/wiki/SIM8200EA-M2_5G_HAT for Raspberry Pi.

Step 7: To get remote access to the RPis, enable VNC and SSH by running

```
sudo raspi-config
```

B Hawkeye Endpoint Setup

This Appendix describes setting up Hawkeye Endpoint on Raspberry Pi 4B model and endpoint for the iPhone 12.

Install Hawkeye Endpoint on Raspberry Pi 4B Download the Ixia endpoint packet file for Raspberry Pi from `https://downloads.ixiacom.com/products/ixchariot/endpoint_library/9.6sp2/pelinux_rasp_pi_2_96.tar` and unpack it

```
tar -xvf pelinux_rasp_pi_2_96.tar
```

Navigate to the *endpoint.ini* file and edit it. Remove the comment and before REGISTRATION_SERVER_ADDRESS, and define it as

```
REGISTRATION_SERVER_ADDRESS x.x.x.x
```

Where x.x.x.x is the server IP address of the Hawkeye server for the project. Save the file. While inside the endpoint folder. Execute the following command to start the endpoint and register to the Hawkeye server.

```
sudo ./endpoint
```

A command prompt will appear and to confirm the probe registration on the Hawkeye server, validate that the output is:

```
The configured Registration Server is x.x.x.x
Connected to the Registration Server
```

Where x.x.x.x is the address previously configured in the *endpoint.ini* file. Also, make sure that the endpoint, and its required IP addresses, are visible on the Hawkeye Server under the Probe Management User Interface (PMUI).

Install Hawkeye Endpoint on iPhone 12 To install the Hawkeye Endpoint on iPhone 12, download IxChariot EP developed by Ixia from App Store. This App lets one define the Registration Server in the GUI and display "Connected" if it is connected to the given server. Ensure that this is the case in the PMUI for the Hawkeye Server.

C Test Scenario 4 Re-test

This section presents the results gathered from the retest of Scenario 4, where the iPhone 12 is located at the office at DTU, connected to TDC's 700 MHz 5G network. Furthermore, the RPT is connected to the NTNU LAN. This test setup is the same as described in section 5.5, and tests are executed in the same way. Section C, section C and section C presents findings from the N-KPI, S4B and ST tests respectively. Section C summarizes the test scenario by presenting the resulting KPIs.

N-KPI Test The latency measurements from the three N-KPI tests for this scenario is given in Figure C.1 for T1, Figure C.2 for T2 and Figure C.3 for T3. These figures show the One Way Delay (OWD) for UL (i12 to RPT) and DL (RPT to i12). For all tests for this scenario, i12 is represented as the "from" probe, and RPT is represented as the "to" probe. Essential latency metrics from these three tests are represented in Table C.1. This Table shows maximum (Max) and average (Avg) latency for the UL, DL, and Round-Trip Time (RTT).

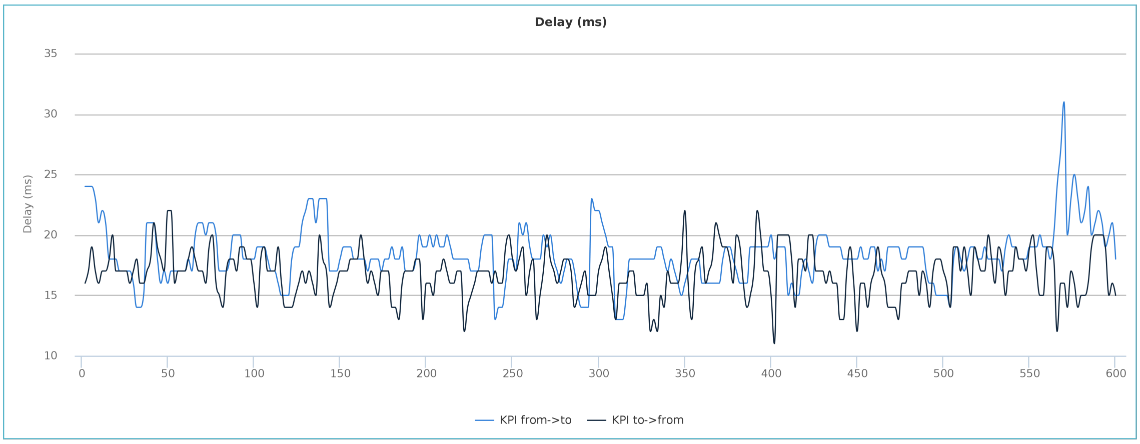


Figure C.1: Latency findings for 10 minutes of the N-KPI T1 between the i12 and RPT. Blue line represents UL and black line is the DL.

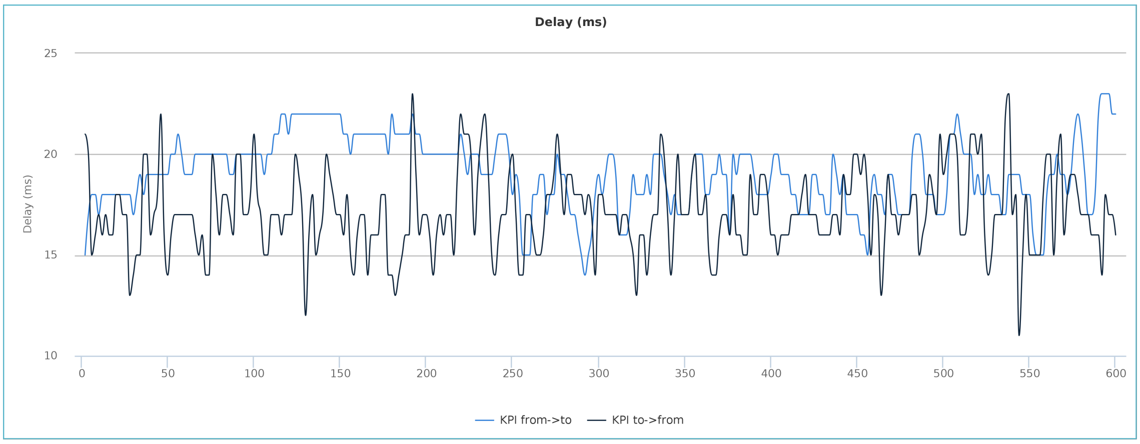


Figure C.2: Latency findings for 10 minutes of the N-KPI T2 between the i12 and RPT. Blue line represents UL and black line is the DL.

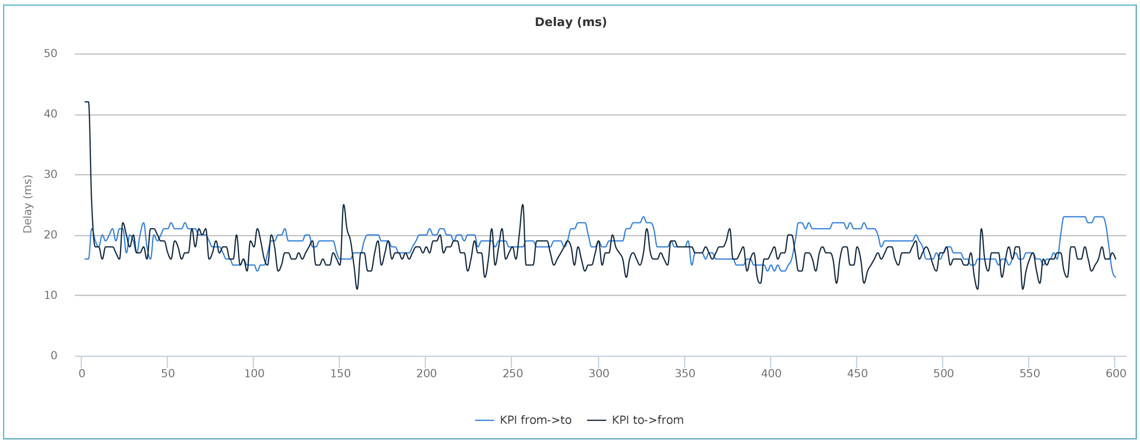


Figure C.3: Latency findings for 10 minutes of the N-KPI T3 between the i12 and RPT. Blue line represents UL and black line is the DL.

Table C.1: N-KPI latency observations from the three tests.

N-KPI Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	23	18.55	42	17.08	65	35.63
T2	23	19.06	23	17.12	46	36.18
T3	31	18.41	22	17.00	53	35.41

Packet Loss during N-KPI
Packet loss was observed for the N-KPI T1 and T2, and T1 is presented in Figure C.4. Y-axis represents the loss in percentage.

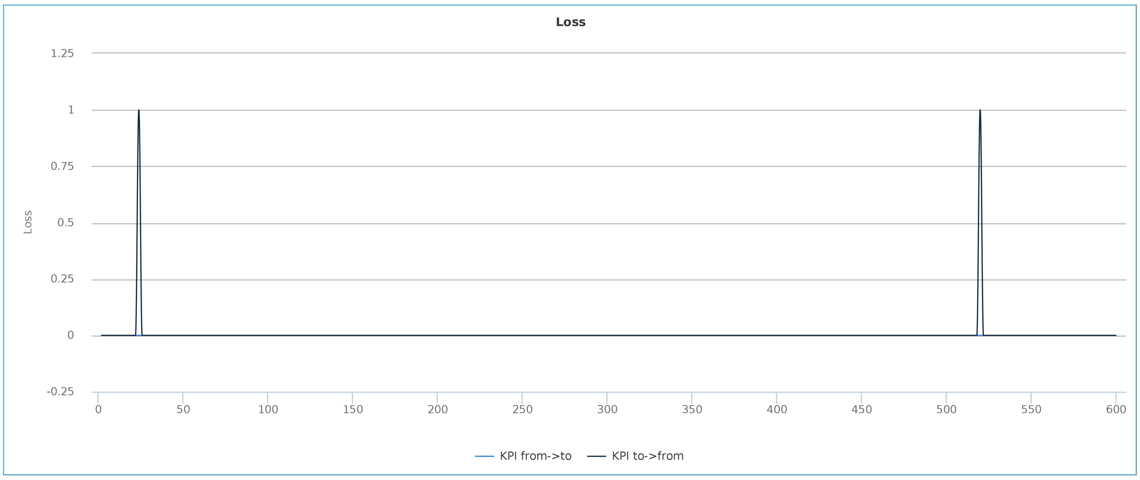


Figure C.4: The packet loss from N-KPI T1.

Table C.2: Percentage packet loss from three N-KPI tests.

Packet Loss N-KPI		
Test	UL (%)	DL (%)
T1	0	0.01
T2	0	0.01
T3	0	0

S4B Test A 10 Mbps video-stream was sent both ways between the two endpoints. The streaming services from S4B T1 is represented in Figure C.5.

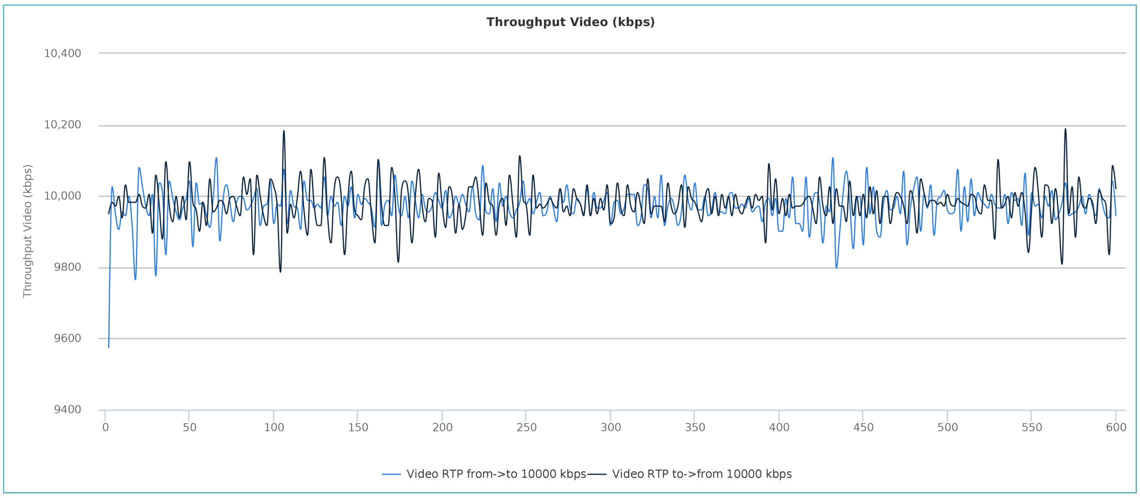


Figure C.5: The bidirectional streaming service between i12 and RPT for S4B T1.

The OWD of S4B T1, T2 and T3 are presented in Figure C.6, Figure C.7 and Figure C.8 respectively. Essential latency metrics from these three tests are represented in Table C.3.

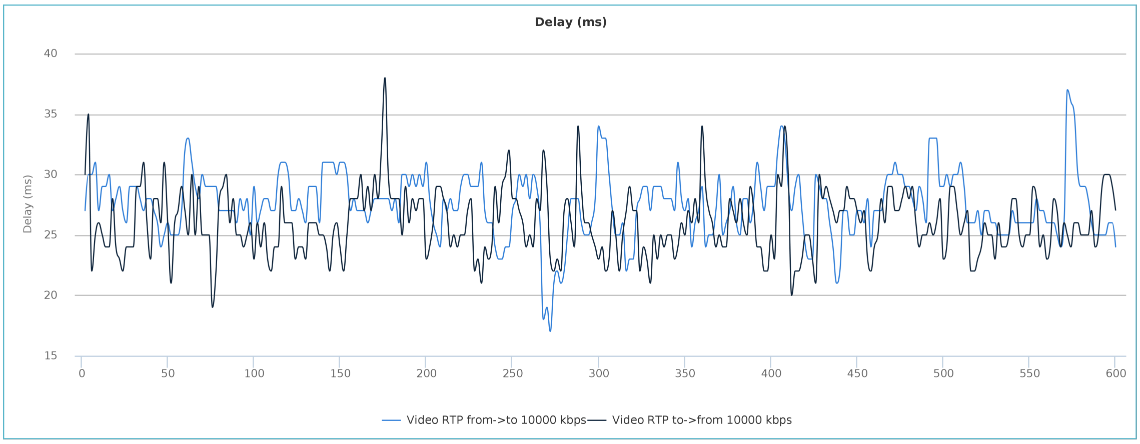


Figure C.6: Latency findings for 10 minutes of the S4B T1 between the i12 and RPT. Blue line represents UL and black line is the DL.

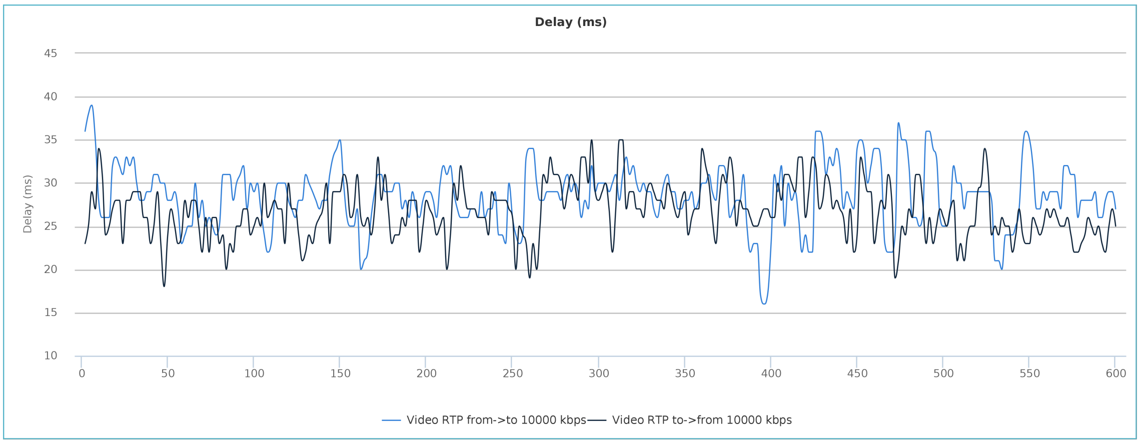


Figure C.7: Latency findings for 10 minutes of the S4B T2 between the i12 and RPT. Blue line represents UL and black line is the DL.

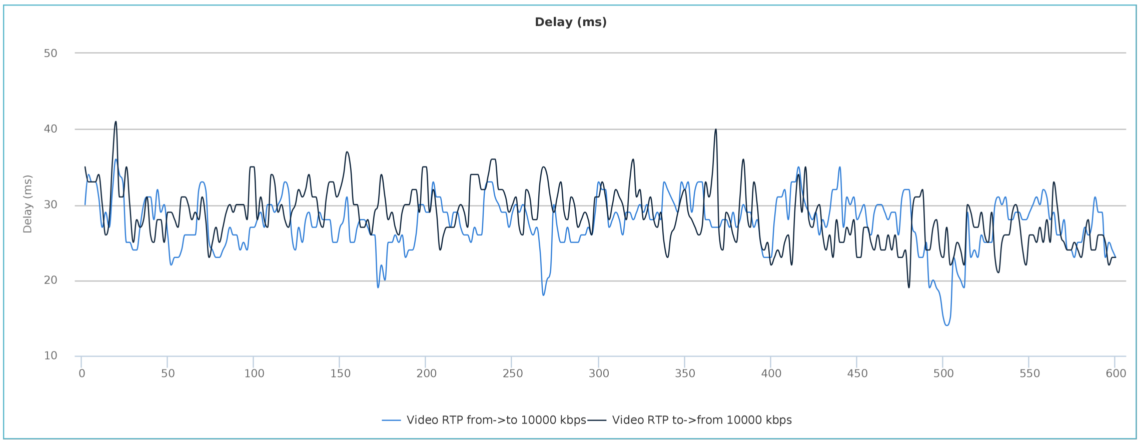


Figure C.8: Latency findings for 10 minutes of the S4B T3 between the i12 and RPT. Blue line represents UL and black line is the DL.

Table C.3: S4B latency observations from the three tests.

S4B Delay						
	UL		DL		RTT	
Test	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)	Max (ms)	Avg (ms)
T1	39	28.53	35	26.70	74	55.23
T2	37	27.54	38	25.97	75	53.51
T3	36	27.56	41	28.50	77	56.06

Packet Loss during S4B Packets were lost for DL connection during T3. Outputs from S4B T3 is shown in Figure C.9. Y-axis represents the loss in percentage. Results from all three tests show a packet loss represented in Table C.4.

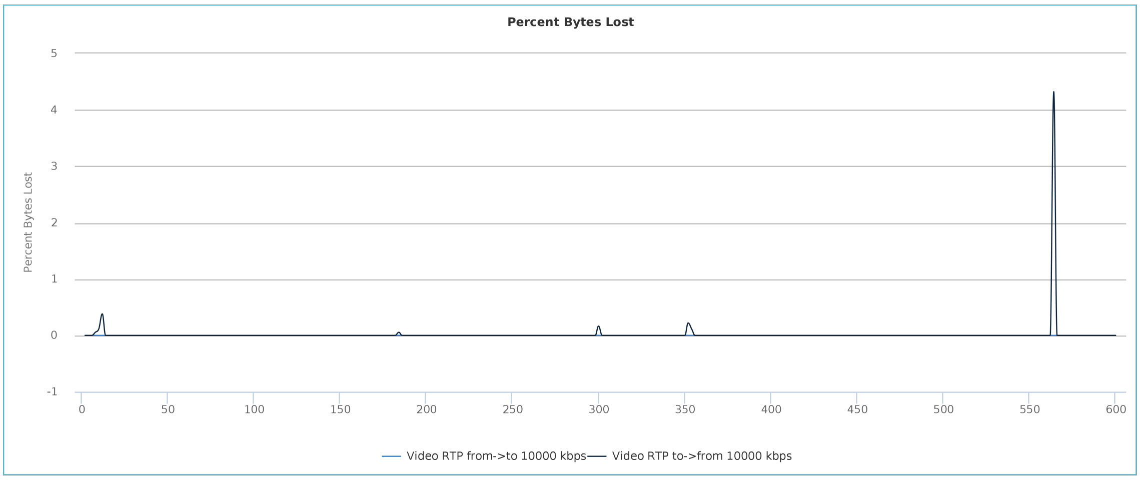


Figure C.9: Packet loss during S4B T3.

Table C.4: Percentage packet loss from three S4B tests.

Packet Loss S4B		
Test	UL (%)	DL (%)
T1	0	0
T2	0	0
T3	0	0.02

ST test Findings of the three STs between i12 and RPT are presented in Table C.5. This Table shows the Average throughput available between the two endpoints over three different periods.

Table C.5: Average throughput from three STs.

Throughput ST		
Test	UL (kbps)	DL (kbps)
T1	39900	90736
T2	36661	90549
T3	39488	89538

KPI Findings The retest to retest communication link between i12 and RPT was monitored, and measurements resulted in the KPIs presented in Table C.6.

Table C.6: Summary of the KPI's collected for the i12-RPT test scenario.

KPIs										
	OWD N-KPI (ms)		**OWD S4B (ms)**		**Throughput ST (kbps)**		**Reliability N-KPI (%)**		**Reliability S4B (%)**	
Test	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL
T1	18.55	17.08	28.53	26.70	39900	90736	100	99.99	100	100
T2	19.06	17.12	27.54	25.97	36661	90549	100	99.99	100	100
T3	18.41	17.00	27.56	28.50	39488	89538	100	100	100	99.98
Avg.	**18.82**	**17.07**	**27.88**	**27.06**	**38683**	**90274**	**100**	**99.993**	**100**	**99.993**