Master's thesis

NTNU
Norwegian University of Science and Technology

Ingvild Sørum Henriksen

Detecting network anomalies in Ethernet/MPLS/IP networks using high resolution delay measurements

Master's thesis in Communication technology
January 2022

NTNU
Norwegian University of
Science and Technology

Ingvild Sørum Henriksen

Detecting network anomalies in Ethernet/MPLS/IP networks using high resolution delay measurements

Master's thesis in Communication technology
January 2022

Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU – Trondheim
Norwegian University of
Science and Technology

Detecting network anomalies in Ethernet /MPLS/IP networks using high resolution delay measurements

Ingvild Sørum Henriksen

Submission date: June 2022
Supervisor: Steinar Bjørnstad, NTNU
Co-supervisor: Steinar Bjørnstad, NTNU

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Title: Detecting network anomalies in Ethernet /MPLS/IP networks using

Student: Ingvild Sørum Henriksen

Problem description:

Today an increased number of services are becoming digital, including digital banking and health services. These businesses need sensitive data, like the personal security number of their client, to deliver their service. Data regarding the client is often stored in a data center, which causes sensitive data to be transported through a communication network. Due to people having criminal motives, the network links to the data center should be secured. A goal is to detect a change in the network path, e.g., due to a man-in-the-middle attack, which should be detected as soon as possible to minimize the damage from the attack.

For this master thesis, a detection method using round-trip time (RTT) is presented. The main goal is to identify a change in the network path or a changed network path. This is done by using easily accessible hardware that is not specifically designed for this purpose. The reason for this choice is to make the method available for personnel without the need of buying dedicated equipment for the purpose. The accuracy of the RTT measurements of the equipment may limit the ability to detect changes. To make sure the method proposed gives statistically significant results, many RTT measurements are done to limit the impact of the equipment's accuracy.

Date approved: 2022-02-18

Responsible professor: Steinar Bjørnstad, NTNU

Supervisor(s): Steinar Bjørnstad, NTNU

Abstract

Today an increasing amount of data is becoming digital, and the need to protect networks to keep them secure is crucial. This study aims to investigate whether non-designated hardware and software can detect changes in a network path or a changed network path. The method is designed to be used on networks that are controlled by operators that is familiar with the topology of the network. The network used in this study is located at Norwegian University of Science and Technology (NTNU). A limitation is that the network studied is operated by NTNU, and not by the researchers, which was a constraint when choosing the network paths to study.

To detect a changed network path or a change in the network, delay anomalies at layer one, layer two, and layer three in the OSI model are detected by looking at RTT measurements. The non-designated hardware used is a personal computer, and the non-designated software is HrPing, made for Windows computers to get RTT measurements in the scale of microseconds. Furthermore, the research focuses on what types of requirements are needed to make the chosen hardware and software capable of detecting delay anomalies, including looking at the accuracy of the RTT measurements where the accuracy is defined in terms of standard deviation and 95%-confidence intervals. Probability density functions were used to illustrate the data collected.

The findings of this study were that 1400-byte IP packets give the most accurate results among the packet sizes studied. The packet size should not exceed 1500 bytes since this is the maximum Ethernet frame size, which can make packets fragment. A key finding was that the type of destination pinged affected the accuracy of the measurements. The desired destination type is a server running a few processes, making the ICMP echo response messages being processed in approximately same amount of time. Pinging a server gave more accurate measurements, making it easier to detect a path change when comparing data in two density plots. The way in which the load in the network affects the RTT measurements is also studied, with the conclusion that the load in the network, in the two periods studied, did not add enough delay to make an evident change in the RTTs.

Sammendrag

I dagens samfunn blir fler og fler tjenester digitalisert. Dette fører til at mer data flyter gjennom nettverkene, og et økende behov for å ha sikre nettverk for å unngå å få data på avveie. Denne studien undersøker i hvilken grad maskinvare og programvare som ikke er designet for å oppdage avvik i RTT målinger, faktisk har mulighet til å oppdage dette under noen omstendighet, og i så fall hvilke omstendigheter dette er. Målet er å bruke avvik i RTT målinger til å oppdage en endret nettverksvei eller endringer i en nettverksvei. Metoden er utviklet til bruk i nettverk som kontrolleres av en operator som er kjent med topologien til nettverket.

For å kunne oppdage slike avvik som kan være en endret nettverksvei eller endringer i en nettverksvei er det viktig å vite hvilke behov som er nødvendig for å bruke lett tilgjengelig maskin- og programvare. Dette inkluderer også hvor nøyaktig forsinkelsemålingene er. I denne studien er det valgt å bruke standardavvik og 95%-konfidensintervall da de ikke er mulig å fastslå et faktisk tall på nøyaktigheten til målingene. Måten å sammenligne RTT målinger er å lage en sannsynlighetstetthetsfunksjon for hver av nettverksveiene og sammenligne disse. Ved å sammenligne konfidensintervall og standardavvik for de ulike sannsynlighetstetthets-funksjonene blir det mulig å komme med forslag til hvilke krav som må være til stede for å bruke lett tilgjengelig maskin- og programvaren til å finne avvik i RTT-målinger som kan være forårsaket av endringer i nettet.

Resultatene fra studien viser at 1400-byte IP pakker ga mest nøyaktig målinger ut ifra de pakkelengdene som blir brukt. Studien viser også at IP pakkene som blir sendt i nettverket for å måle RTT ikke må overstige 1500 bytes da dette er grensen for last en Ethernet frame kan frakte. Dersom denne grensen overstiges vil pakkene fragmenteres dersom ikke alle enhetene i nettverket klarer å håndtere såkalte jumbo-pakker. Studien gjort viser at når man skal gjøre slike målinger vil destinasjon være med på å påvirke nøyaktigheten på målingene. Det iser seg at en server vil gi mer nøyaktige målinger enn en ruter, basert på hvilke andre prosesser som kjører på enhetene. ICMP echo response meldingen vil kunne oppleve ulik lengde på hvor lenge den må bli prosessert og ut ifra dette studiet var ventetiden for å bli prosessert av en server mer stabil enn når en ruter ble pinget mot. Dette gjør det lettere å skille på ulike nettverksveier. Til slutt er last i nettverket, og hvordan det påvirker RTT målingene studert. Basert på resultatene er at variasjonen i last i denne studien ikke gir store nok til at maskin- og programvaren klarer å detektere endringene.

Preface

This thesis is a part of the Masters degree in Communication technology at NTNU. A preliminary project were done by the author in the fall semester of 2021 as a preparation for this thesis. The interest for security in network and the need for protecting confidential information was the inspiration for writing this thesis. The need for protecting data is a crucial part of the digitization and requires accessible methods to easily be able to detect changes in the network. The reader should be familiar with networks and network delay.

Acknowledgment

I want to say thank you to my supervisor Steinar Bjørnstad for valuable input on my work guiding me through challenges with his knowledge and being a great partner for discussion. I would also say thank you to Alfred Arouna at Simula Oslo MET for giving advice in the start-up phase on handling the data gathered for this study. I would also like to thank the girls at my office for great conversations in a satisfying working environment inspiring me throughout the process.

Contents

List of Figures

List of Tables

List of Acronyms

ARP Address Resolution Protocol.

CPU Central Processing Unit.

DoS Denial of Service.

HTTP Hypertext Transfer Protocol.

ICMP Internet Control Message Protocol.

IP Internet protocol.

KDE Kernel density estimation.

LAN Local Area Network.

MDS Multidimensional scaling.

MitM Man-in-the-Middle.

MTU Maximum Transmission Unit.

NIC Network Interface Card.

NTNU Norwegian University of Science and Technology.

OSI Open Systems Interconnection.

PDF Probability density function.

PPP Point-to-point.

QoS Quality of Service.

RAM Random Access Memory.

RTT Round-Trip-Time.

SBD Shape-based Distance.

SLA Service Level Agreement.

SMS Short Message Service.

TCP Transmission Control Protocol.

TTL Time-to-live.

TWAMP Two-way Active Measurement Protocol.

UDP User Datagram Protocol.

VPN Virtual Private Network.

Chapter 1
Introduction

1.1 Motivation

Telecommunication has become invaluable in modern society due to the increase in the number of digital services being developed. Today, doctor appointments can be made using video and bank loans, and other financial activities are done through digital services. The increased use of services that involve sensitive information requires a secure network. This is done to ensure that the data is protected. There is also a need to have a server where the data is stored. These servers can be local or part of a larger data center. The network path to transport data should be secure, without any risk of data being leaked or stolen. However, there are many different ways criminals can perform a network attack. According to [JKA+16], there is an increase in the frequency of network attack methods, which can be classified as sniffing, spoofing and Denial of Service (DoS) attacks.

One of the most common reasons for criminals wanting to steal sensitive information about a firm or people is related to money. The criminal can sell the sensitive data on the black market or use them for blackmail. The misuse of sensitive data can be harmful to the people who own the data. For instance, the person can suffer substantial economic losses. A state may also steal and leak sensitive information about another country to harm that government. This can make the state less attractive to the public. Due to the importance of information technology in society, it has been accepted as a crucial element in the maintenance of national critical infrastructure systems, according to [MKH11].

Governments are becoming more aware of how a cyber attack can damage to what extent and what methods may be used. In recent years, emails have leaked from the Norwegian government. In 2021, Norway's Deputy Chairman of the Foreign Affairs and Defence Committee got 4000 emails leaked [RLS21]. Amedia, the second largest media group in Norway, owns or is a part-owner of 80 local and regional newspapers. On 28 December 2021, they were exposed to a cyber attack, which

made their data systems out of service. This made it challenging to produce their newspapers. It took three days before their physical newspaper could be sent out to customers again. The motivation behind the attack was ransom [Ame22].

For this thesis, telecommunication networks are being studied. The main goal is to develop a method that can detect a changed network path or if changes in the network path have occurred. To detect these changes, Round-Trip-Time (RTT) is being studied. Changes in the network path can be known changes, e.g., a new router has been added or replaced by a new router. It can also mean that a device has been added to a path without the consent of the network operator, which must then be further investigated. It may be a device that can be used to perform e.g., a Man-in-the-Middle (MitM) attack. A changed network path is when the data packet takes a new path, e.g., due to an updated routing table in a router. Due to the increased amount of sensitive data that today's telecommunication networks are transporting, this thesis aims to find a reliable and effective method to detect changes in a network. The proposed method should be easily accessible to people who operate a network.

Figure 1.1 is the main idea of the study presented. The figure shows a changed network path, which is what we want to detect. The green arrows show the normal path taken by the data packets between a computer and a server. The red arrow shows the rerouted path. The rerouted path can illustrate an intruder who has acted on the network. Using a computer, we want to send Internet Control Message Protocol (ICMP) echo requests to the server and have the server send ICMP echo responses in return. ICMP is further described in Section 2.3. If we do such RTT measurements regularly on the green network path, and then the data packets suddenly take the red network path, we wish to discover a different RTT and then know that the network path has changed. The change in RTT is referred to as delay anomalies throughout the thesis.

The methodology presented is designed for Local Area Network (LAN) where there is an operator managing the network. The operator will know the structure of the components and can use the proposed method to detect whether the structure has changed. The proposed method cannot be applied to a more extensive network, such as the Internet, because there are many processes that we cannot control. Additionally, processes running in the Internet can cause the RTT to vary by several milliseconds. The network studied here is described further in Chapter 4.

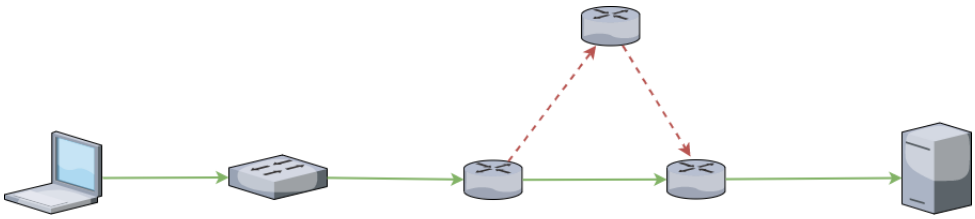


Figure 1.1: Rerouting in a network creating a new path of five hops

1.2 Research questions and hypothesis

The pre-project for this thesis defines three research questions and two hypotheses. The research questions are the focus area to be answered, while the hypotheses are stated as true to support the proposed method. The main goal is to find out whether non-designated types of equipment can do RTT measurements accurately to detect any changes in a network path or detect a changed network path. The research questions are made to specify the research.

> *Research question 1: What is the accuracy of the delay measurement when using a computer connected to the network?*

The first research question focuses on the accuracy of the type of computer used in the study. This means that the question seeks an answer regarding the accuracy of the timestamping of the data packets. A computer will most likely have several other processes running, which may affect the timestamping of data packets, thus affecting the RTT. Some packets receive a more exact timestamping than others. By making thousands of RTT measurements, the results are more likely to show the time that most packets have in a particular path. However, if the results do not show any distinct patterns, the accuracy of the chosen equipment may not be accurate enough.

> *Research question 2: Is it possible to use non-designated hardware and software to detect delay anomalies in the network?*

The second research question aims at summarizing the entire study. The question also tries to reflect on whether it can be stated that the methodology works, and

if not, if different types of equipment would give different results. For example, a computer running Linux is made differently from a Windows computer, and therefore they might give different results. If the proposed method does not give the desired results, it does not mean that non-designated equipment for measuring RTT can not be used. It may be that different types of equipment must be chosen.

> *Research question 3: What conditions must be present to make a computer recognize a change in delay in a network path?*

The third research question aims to study the conditions that must be present to make the types of equipment used capable of detecting delay signatures. A delay signature for this project is the amount of delay that a network device adds to the total RTT. When talking about conditions, it is meant to find out what other processes, or how many other processes the equipment used, can run, in addition to the pinging tool to get accurate measurements. It also includes finding out whether the type of destination pinged affects the delay measurements.

Two hypotheses have been made for this project. The first hypothesis states that a change in the physical path can be recognized based on how much time different packets use through a network, based on differences in the end-to-end delay. A change in the network path is, e.g., that a device has been added, which may increase the end-to-end delay in the network. A changed network path is, e.g., that the packets are now traveling through different network devices, for example a router, which can cause the total end-to-end delay to increase.

> *Hypothesis 1: Based on the delay in the network it is possible to recognize physical path changes in the network*

The second hypothesis states that a hardware device can detect a delay signature in a network. This is especially relevant for research question two. It is confirmed in a study where specialized hardware is used. Since this project aims at using non-designated hardware, this may not be the case.

> *Hypothesis 2: It is possible to recognize a delay signature for a hardware device in the network.*

1.3 Outline

This thesis is divided into seven chapters: Introduction, Background, Related work, Methodology, Experimental work, Discussion, and Conclusion and future work. Chapter two contains background related to the concepts to do RTT measurements and introduces the software used. Related work is them presented to set the work done in a context, and new the elements to this study are presented in Chapter three. Chapter four describes the methodology and introduces the concepts used to analyze the data gathered. The experimental work is then presented, including a section describing the setup and results of the study. Following is the discussion of the results and the research questions are answered. Finally, a conclusion and future work is presented in chapter seven.

Chapter 2
Background

This section presents background related to the concepts and software used in this project. First, a presentation of what network delay is and where the network delay occurs. Then, a section on how to measure the delay is presented, followed by an explanation of the Open Systems Interconnection (OSI) model and ICMP. In the end of this chapter, relevant tools are introduced and an overview of network attacks that can be detected using RTT as a tool.

2.1 What is network delay?

The three primary sources of delay in a network are the transmission delay (also known as serialization delay), the propagation delay, and queuing delay according to [MAB18]. Transmission delay is the time it takes for all the bits in a data packet to be transmitted on the outgoing link of, e.g., a router. The transmission delay depends on the size of the packet and the speed of the link. A high-speed link has a shorter transmission delay than a low-speed link. Propagation delay is defined as the time a signal uses to travel through a data link [MAB18].

The queuing delay is the most significant contributor to the total end-to-end delay. It occurs when data packets wait to be processed by a network device, e.g., a network router. More packets in the network lead to a longer queuing delay. There will be some packets that experience a minimum queuing delay, called *lucky packets*, while others experience a significant queuing delay. Some packets might experience too much delay, which leads to the packets being dropped.

Processing delay is also considered a type of delay in a network, but its contribution to the total delay is negligible. This can also be said about the propagation delay. However, [RWW04] addresses the fact that processing delay has become a significant contributor to the total packet delay. According to the study, the processing delay can be in the order of milliseconds. The increased processing delay is due to the increased use of Virtual Private Network (VPN) tunneling, encryption, and the use of firewalls.

Routers use more time to process data packets, leading to a longer processing delay. The study states that processing delay has become a significant contributor to delay measurements. The observation creates a delimitation for the methodology used for the project. The end-to-end delay is the sum of the different delays in the network, making RTT the sum of the end-to-end delay in both directions of the network path. The different types of delay occur at every hop in the network, both at switches and routers. Figure 1 shows an illustration of the different types. To simplify figure 2.1, the delays are only shown in one place; however, they will occur on all switches and routers in the network)

The expected values of RTTs in the proposed network are in the order of milliseconds, being around 1 ms in a path consisting of two hops in the network studied and the tools used. When VPN is not used, the processing delay is measured in the order of microseconds. Thus, RTT is not significantly affected by the processing delay. If the processing delay increases to be in the order of milliseconds, which can occur when VPNs are used, as stated in [RWW04], RTTs will be influenced. However, if the added processing delay is the same for every path studied, using the proposed methodology on VPN networks would not be a problem. However, the processing delay depends on the processing of the headers of the packet, which may not be the same for every packet.

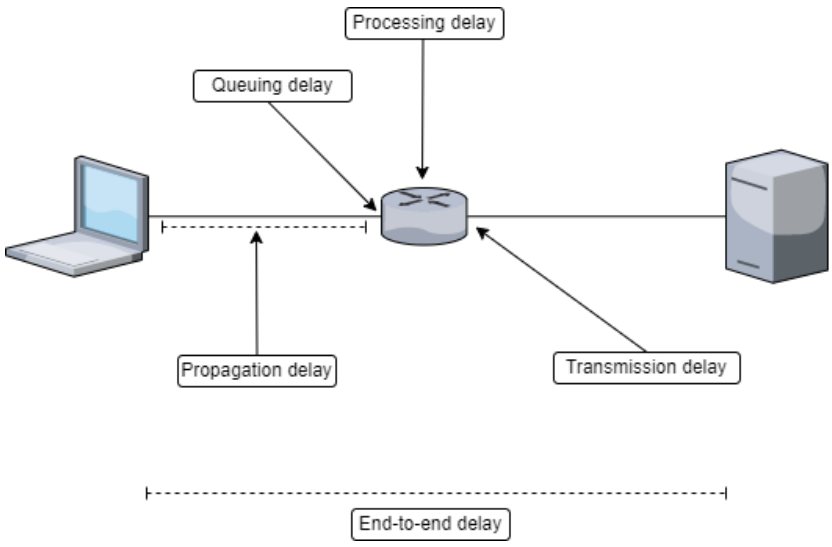


Figure 2.1: Illustration of a network that shows where different delays occurs.

2.1.1 How to measure the Round-Trip-Time

RTT is the time a packet uses between the sender and the receiver in a network, plus the amount of time the packet uses to be acknowledged by the sender, explained in [KR17]. RTT is also known as ping time, as a pinging tool is the most frequent tool used to measure RTT. How a pinging tool works is described further in Section 2.4. Figure 2.1.1 gives an illustrations of how RTT is measured throughout a network path.

There are different ways to measure RTT, using User Datagram Protocol (UDP), Transmission Control Protocol (TCP), ICMP. [WDJG07] studied the difference between using TCP and ICMP when measuring host connectivity, RTT and packet loss rate. For this project, the different methods of measuring RTT are important in case of any delay is added to the RTT depending on which protocol is used. Therefore, it is relevant to know if there is a significant difference between using TCP and ICMP to measure RTT.

[WDJG07] used regular ICMP ping and a ping method for TCP based on TCP SYN/ACK. [WDJG07] found that for paths with a small delay, RTT using TCP had no significant differences compared to using ICMP. However, when pinging on paths with large delays, they saw that the TCP ping gave a larger RTT than the ICMP ping did. The difference was on the scale of tens of milliseconds.

[WDJG07] concluded by stating that the path load and the host load affected which protocol to use. Introducing a value α, where $\alpha = RTT_{mean}/RTT_{min}$. For the light-loaded hosts TCP had a smaller RTT than ICMP with a few milliseconds. Hosts with high loads, where α was smaller than 20, TCP and ICMP gave almost similar results. However, when α was larger than 20, TCP gave higher values of RTT compared to ICMP. Therefore, when selecting which protocol to use when measuring RTT, the choice should be based on the value α according to [WDJG07].

UDP may also be used to measure RTT, but is not the most common application [TRLC15]. UDP is rather used for checking the reachability of a host in a network topology. UDP can be used to measure RTT using Two-way Active Measurement Protocol (TWAMP) [NGSB20]. The way TWAMP works is by first sending a stream of UDP packets to a specified host. To measure the time UDP packets used between the sender and receiver is by having a reflector at the end device, which sends an echo response with a timestamp back to the sender, accoring to [NGSB20].

2.2 OSI model

The OSI stack or OSI model is a conceptual model that standardizes the functions of a telecommunication system [KR17]. Li et. al explains the OSI model in [LLCZ11].

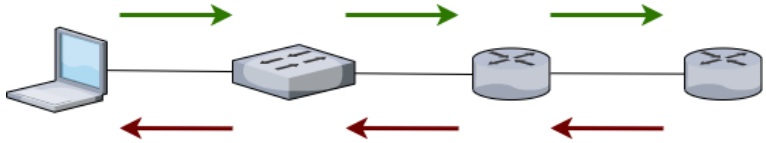


Figure 2.2: Illustration of round-trip-time

The model is divided into seven abstract layers where each layer has a function. Some models are presented with fewer layers to make the structure less complicated. The model is illustrated in Figure 2.2.

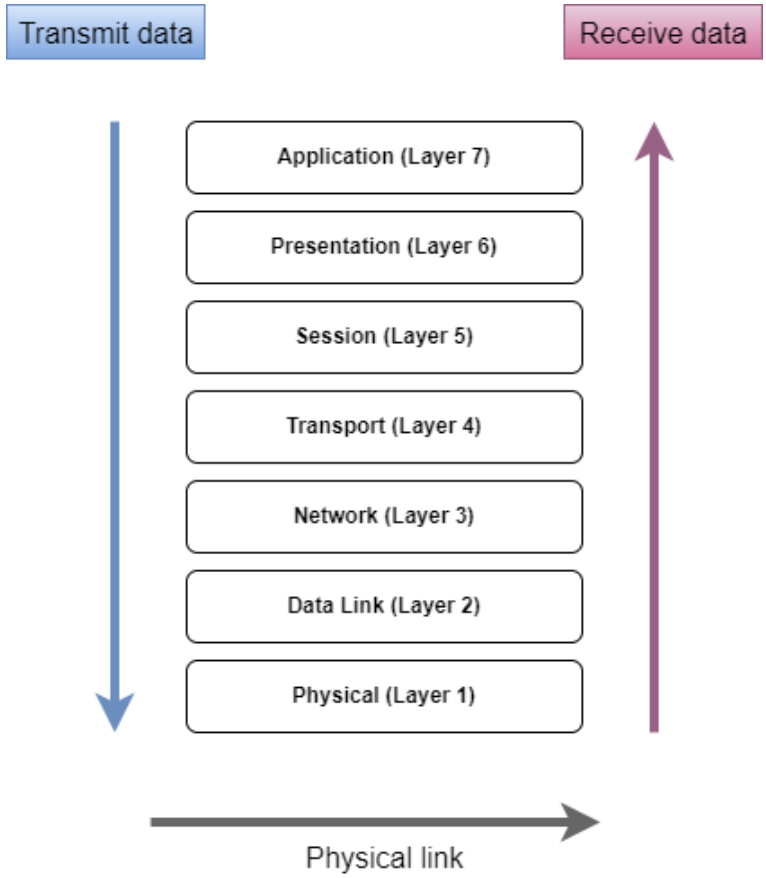


Figure 2.3: The seven network layers

Layer five and layer six are sometimes combined with the seventh layer, making the model only consisting of five layers. Layer 1 represents the layer that transports

the bit streams on a physical medium. The protocol related to layer one is Ethernet. Layer 2 represents the transmission of data between two nodes in a network. This layer is the link connecting two nodes. Network switches are part of layer 2, and protocols related to this layer include Point-to-point (PPP) and IEEE 802.11. The third layer is the network layer and provides the functionality of transferring data between two nodes in different networks. Network routers are part of this layer and the protocols belonging to this layer include Address Resolution Protocol (ARP), Internet protocol (IP), and ICMP [LLCZ11].

Layer four is the transport layer that provides functionality that transports data sequences from a source to a destination. It is in this layer that the functions related to Quality of Service (QoS) are maintained. For instance, the transport layer can have flow control and error control depending on the protocols used. TCP and UDP are among the protocols used in this layer. Layer 5 is often said to be part of the TCP protocol. However, it is said to control the dialogues between the computers communicating. Layer 6 is the part that establishes a connection between the entities of the application layer and translates between the application and the network formats used in the communication. Lastly, layer 7 is the layer that interacts with the software. This layer has functions including identifying communication partners, synchronizing communication, and determining whether resources are available. Hypertext Transfer Protocol (HTTP) is a protocol used here [LLCZ11].

2.3 Internet Control Message Protocol

ICMP is one of the most widely used protocols to measure RTT in a network, and is a layer 3 protocol as stated in Section 2.2. The protocol has several important functions that are used to troubleshoot and announce network problems. The functions include announcing network errors, congestion, and timeouts that lead to packets being dropped [Har00]. There are up to 256 possible control messages to report different variations of errors. However, not every control message is relevant, and some have been deprecated [PJM+]. ICMP differs from the transport protocols such as UDP and TCP in the way that it is not used to exchange data. End-users of network applications also rarely use this protocol. The only exceptions are when networks are diagnosed by using ping and traceroute tools. For this thesis, the type of control message wanted is type-0, also known as Echo reply [Pos81], used for pinging. Getting an echo reply means that a host is reachable.

2.4 Pinging tools

Ping is the most common tool used to test connectivity in a network and measures RTT. Ping works by sending an ICMP echo request to a host and waiting for the

destination host to reply with an ICMP echo response message [Cis06]. If the reply is received, the destination is reachable [Pos81]. Different ping applications are available on most computers, but the features will vary depending on the operating system the computers run. The ping application on a Linux machine is different from the ping tool on a Windows machine. One of the differences is how accurate the measurements are displayed. Different settings are applied to the ping sessions and vary between the tools. Due to ordinary Windows ping only showing the RTT using milliseconds, *HrPing* [cFo22] is used for this thesis. Figure 2.1.1 is an illustration of how ICMP packets travel through the network when a router is pinged. The network in Figure 2.1.1 consists of a host, a switch, and two routers. The host sends an ICMP echo request to the router on the far left, represented by the green arrows. The pinged router then sends an ICMP echo response back to the host, represented by the red arrows.

2.4.1 HrPing

HrPing is created by cFos Software and is a high precision ping utility made for Windows machines [cFo22]. There is already a ping utility on Windows machines, but the features available with HrPing are more advanced. One of the advantages of HrPing is that it is a freeware, which means that the software is available for use at no cost [Mer22]. One of the main differences is the precision of the ping measurements. cFos Software states that HrPing uses the *performance counter* on the machines to time-stamp packets [cFo22]. A performance counter is a program on the computer that helps monitor different processes on the computer [MZK11]. Therefore, HrPing can measure RTT with a precision of microseconds. HrPing also has functionality that makes it easy to store the output of a pinging session, which is helpful when data analysis is done. HrPing has several other functionalities that are different from regular windows ping. The functionality includes sending multiple ICMP echo requests at the same time, and it is possible to choose the interval during which the ICMP packets are sent. HrPing can also be used to send UDP packets [cFo22].

2.4.2 Traceroute

Traceroute is a tool used for network diagnostic [Cis06], similar to Ping. However, ping can only indicate if there is a problem in the network, while Traceroute can pinpoint where in the network the problem exists. Traceroute does this by mapping the route data packets take between hosts and destinations [DF07]. For each hop that the packets take through the network, the RTT used by the data packet to that point is measured by Traceroute. This is done by sending UDP probes into the network having an increasing Time-to-live (TTL). For each router a packet visits, the TTL decremented. If the destination is not reached and ICMP time-exceed message

is sent to the host [DF07]. However, Traceroute runs at layer three, which means that only layer 3 devices are displayed in the provided trace. As stated in Section 2.2 network switches and routers belong to different network layers. Network routers will be shown in the trace, and not network switches. Thus, Traceroute is a tool that can identify routing changes.

2.5 Types of network attacks

Several different network attacks can be detected due to a change RTT in the network, not only a rerouted path. The type of network attacks includes MitM-attacks, DoS-attacks, and network sniffing, which can be caused by having data rerouted through other network devices not originally belonging to the network. Network sniffing and MitM attacks are due to broken link integrity. Link integrity is needed to avoid losing data and having unauthorized people get their hands on the data.

A MitM attack occurs when the attacker eavesdrops on a conversation that takes place between two devices in a network, described in [AH09]. Thus, when such an attack occurs, there is a change in the network path. The added device will add an additional delay between the two devices being eavesdropped on. Thus, when measuring the RTT, the value has increased. Depending on the device used in the attack, the added delay varies but is likely to be in the range of microseconds (μs). According to [AH09], MitM attacks are one of the most fundamental and widespread attacks performed on distributed systems. A MitM attack can be divided into different types of attacks ranging from spoofing to attacks on critical wireless communication [AH09].

DoS attacks are when the attacker floods a target in a network, e.g., a server. Due to a large amount of traffic, the devices crash and is not reachable for legitimate users [Kum16]. If a device crashes, it is not possible to measure the RTT between the host and that device. Thus, the RTT will detect that something has happened since the device is not responding. When talking about network security, the main objectives are to ensure the availability, integrity, and confidentiality of sensitive data [22], also known as the tree pillars of network security. When a DoS attack occurs, availability is disrupted. Knowing that the lack of response is due to a DoS attack, as other types of failure may occur due to other processes running on the network devices. However, network operators now know that something has happened and that it might be due to a DoS attack. One of the main issues with the detection of DoS attacks is that malicious data traffic cannot be distinguished from normal traffic [Kum16].

[TRKF04] explains network sniffing as being when a malicious user uses specialized software to capture data packets in a network. To make the "packet sniffers" able

to capture all packets in the network, they put the Network Interface Card (NIC) on the computer in a mode called the "promiscuous mode." By doing that, the NIC captures all packets and sends them to the kernel. [TRKF04] presents a technique to use RTT to detect such measurements, which are based on the increase in RTT when the host is in promiscuous mode.

Chapter 3
Related work

There are many studies related to finding a method to evaluate the status of a network. This means finding out if anything has happened and trying to diagnose the network to see if something is different from normal behavior. Several studies have used RTT measurements in different ways to diagnose a network; both large-scale and local networks have been studied.

3.1 Large-scale round-trip delay time analysis of IPv4 host around the globe

The main objective of [Gez19] is to investigate the RTTs over the Internet. The investigation was carried out measuring RTT delays around the globe. The measurements were done over a period of five years, during different periods of the day. Doing measurements in different time periods was due to the internet having different loads. The study sheds light on several different aspects of delay. One of the aspects is related to how different delay components affect the end-to-end delay, including transmission and propagation delay.

The similarities between this thesis and [Gez19] is that RTT measurements are done, and different distribution functions are plotted. The type of plot presented is a Probability density function (PDF), which is what this thesis aims to present. The PDF is presented together with a histogram showing the relation between the two types of plot. This way of presenting is to be considered for this thesis. According to [Gez19] finding a specific distribution fitting is difficult to plot for RTTs below 0.1 seconds. Therefore, the fitting parameters must be considered for every distribution. [Gez19] studied how different times of they affects the RTT, which was also carried out for this thesis.

The difference between this thesis and [Gez19] is that the RTT measurements are gathered for a period of five years. For this thesis, the measurements were collected over four weeks. In [Gez19], the IP addresses that have been pinged are located in

the United States, Turkey, and Japan, thus locations across the globe. For this thesis, the network located at NTNU is used; thus, the number of IP addresses is on a much lower scale. Therefore, the transmission and propagation delay results may not be as relevant due to this thesis studying network paths consisting of fewer hops. In [Gez19], a link may be significantly longer compared to links studied in this thesis, and the propagation delay will most likely affecting the end-to-end delay more in [Gez19].

3.2 Round-Trip-Time anomaly detection

The study [BAČ+18] proposes a method to detect anomalies by analyzing RTT. The motivation behind this study is to provide information about the quality of a connection when sending a Short Message Service (SMS) but also to find ways to isolate content-dependent variations. Finding anomalies in big data is also a contribution and motivation for this study. [BAČ+18] states that it is crucial to monitor and measure RTTs of data packets for at least two reasons. The first reason is to maintain the level of quality of the service negotiated in the Service Level Agreement (SLA). The second reason is to have minimal operational costs. The situation used for this study is a banking scenario in which customers log into a bank account with a unique identification number. The bank then looks up the customer in their database, and the customer is sent a verification code as SMS. Then the customer verifies the code and the log-in procedure is completed. An anomaly situation for this scenario is that the SMS containing the verification code is delayed.

The system used in [BAČ+18] works as follows. The bank uses SMS brokers to avoid having connections and agreements with all operators of mobile networks. This is done by the brokers, which manage traffic between network operators of the customers and customers. Since the bank is a critical service, the bank increases the reliability of its service by having two separate SMS brokers. However, the cost of sending messages to one of the brokers is higher than the other. Therefore, every message is sent to the broker at the lowest cost. If the bank loses its connection to the low-cost broker, then the bank switches to the other broker to avoid delays in delivering the SMSs. [BAČ+18] assumes that to prevent a queue from becoming too long, the SMS broker can delay the response time of the traffic.

Previous results have found that RTT measurements generally have few anomalies in terms of spikes of very long or very short RTTs. One of the objectives of [BAČ+18] is to develop a method that can detect longer periods of RTT anomalies and, at the same time, ignoring the short spikes. These periods are considered outliers. The approach used to detect the cluster of outliers was done by measuring the RTT for almost 300000 messages. A median of the measurements is done to provide a stable base level. Also, they state that a cluster of outliers is detected as an anomaly by

their algorithm only when the cluster contains at least $625/2 = 313$ entries. The rule is created for the methodology. The number is derived from the median of 25 values, which is calculated as $25^2 = 625$ measurements in the study. [BAČ+18] get three groups where the responses are slow. However, only two groups have more than 313 entries, so the algorithm only reports these as anomalies.

[BAČ+18] created an algorithm to calculate the standard deviation and mean of the RTT measurements. Then exponential smoothing was used to isolate the variance as an adjustment factor. Then the median of the larger and smaller groups of RTTs was calculated to identify clusters of outliers. This was possible because the algorithm reduced noise in the measurements. They concluded that the system could detect larger clusters of outliers.

There are several similarities between [BAČ+18] and this thesis. First, both studies aim to use RTT measurements to detect changes in delay in a network. The network studied is also similar in terms of size compared to other studies that do RTT measurements over the Internet. There are also several differences between [BAČ+18] and this thesis, for example, in the way measurements are analyzed. In [BAČ+18], clusters are made from the RTT measurements, while this thesis aims to analyze probability density functions.

3.3 Detection and characterization of Network Anomalies in Large-Scale RTT time series

[HHZ+21] presents a method to detect network anomalies by analyzing large-scale RTT time series. The motivation is to easier detect anomalies due to the impact anomalies disrupt connections in a network and may lead to lower network performance. The question of how network disruptions can cause financial losses is also a motivation. To improve the usability and reliability of the network, it is necessary to understand what conditions must be present in the data plane. However, due to the lack of cooperation with network operators, [HHZ+21] obtain this information using ping and traceroute tools.

Using a ping tool, large-scale time series of RTT measurements are collected. However, it is difficult to detect whether a change in RTT is caused by routing changes, path changes, congestion, or an anomalous event that has happened only by looking at the measurements. Therefore, the method proposed by [HHZ+21] analyzes the amplitude of the monitored network to detect anomalies. The method uses an observation that if a failure occurs in the network, the RTT in a large number of links will be affected. [HHZ+21] also states that RTT measurements that pass through the same routers or related links will have similar characteristics if an anomaly occurs.

The proposed method is an unsupervised two-step method, and the workflow presented is as follows. First, the RTT measurements are extracted from the network being monitored. Then the first change detection method is used. The methodology consists of plotting the RTT measurements in a graph based on the time of the measurement and the value of the RTT. Then the change-point method detects the areas in the graph where the measurements change significantly by forming a new time series. Then the second round of detecting change-points is performed on the new time series, and the abnormal periods are marked with a starting and an ending point. However, not every change point is caused by an anomalous event.

The nodes and links responsible for the event are identified for each detected irregular period. First, links with change-points at or near the anomalous period are extracted. Then the relevant links are distinguished from irrelevant links [HHZ+21]. The network links that are related to the same events will also have the same violent fluctuation during the occurrence of an event. Therefore, to obtain the hidden relationships among the links, a Shape-based Distance (SBD) proposed by [PG] is adapted. To visualize the relationships between the links in a two-dimensional space, the advantage of Multidimensional scaling (MDS), described in [VR02], is used [HHZ+21]. Finally, a Kernel density estimation (KDE) is used to find the area that contains the highest density of link relations. This is to characterize the links and nodes related to events.

The results using the method proposed by [HHZ+21] show that $12,5\%$ of the links are related to an anomalous event. However, false alarms were also detected. Anomalous periods are detected as false alarms when the anomalous period occurs at a different time than the actual anomalous event. In addition, a false alarm can occur when the RTT time series to links that are not part of the anomalous event has the same jitters as the RTT time series as links related to the anomalous event.

Three different scenarios are investigated by [HHZ+21], a distributed DoS attack, and the last case is where a misconfiguration of a switching device has caused a network event. The anomalous events are identified for all the different cases using the proposed method. The conclusion made by [HHZ+21] is that the proposed method is successful in terms of detecting network events and characterizing the events for both simulated and real-life data. The methodology proposed by [HHZ+21] is also very accurate in terms of detecting where the anomalous event has occurred, which reduces the time needed for troubleshooting.

The similar elements of the study [HHZ+21] and this thesis are that RTT measurements are used to detect anomalies in a network. However, the RTT measurements gathered is at a larger scale compared to this thesis. This study has developed a new type of tool to detect an anomaly event occurring, while this thesis aims to use

already developed tools.

3.4 Vesper: Using echo analysis to detect man-in-the-middle attacks in LANs

Vesper is a MitM detector created by Mirsky et al. in [MAB18]. The detector is plug-and-play and is designed to be used in a LAN. The proposed solution by [MAB18] uses signal processing and more specific impulse responses to detect network attacks. The main idea is to have the detector modeling the topology of a LAN from the impulse responses, and if the impulse response changes, then something has happened to the network topology. The way impulse responses are used in [MAB18] is by having Vesper capture them by measuring RTT from an ICMP echo request burst. The impulse response from the results of the measurements RTT is then used to model the LAN. If a third party enters the environment, the impulse response between the two hosts communicating in the network will have significant changes.

[MAB18] looks at three different scenarios in which an intruder can attack the network, *end-point, in-line* and *in-point*. End-point is when the intruder adds a new host to the LAN. In-line is when the intruder locates a network cable exposed and used for communication between two hosts. In-point is when the intruder finds a switch used to communicate that is exposed and changes that switch to a new switch. The intruder can use the new switch to manipulate the traffic by having additional logic. Both in-line and in-point required additional hardware. TO perform such attacks physical access to the LAN is required [MAB18].

The presence of a MitM attack is first discovered using the ICMP protocol to capture the time a packet uses between two hosts in the LAN. The RTT used by the ICMP packets between hosts depends on the number of switches in the network path. The current load in the network also affects the RTT. From the sequences of the ICMP echo request, a signal is produced. This signal is then used to produce an impulse response. Thus, when an intruder intercepts the traffic between two hosts, the impulse response changes because the intruder changes the dynamic of the communication channel. This happens because the different hosts use different hardware and software in the LAN [MAB18].

[MAB18] have plotted the RTT before and after a MitM attack using a density function. The graph shows that the two plots have almost the same shape. However, the plot representing the RTT after a MitM occurs, the plot has shifted to the right. This means that the mean RTT is longer during a MitM attack.

The study states that *Vesper* can be deployed in two different ways in a LAN. The first is to have *Vesper* running on one of the two hosts communicating, which

then will protect the link they are using to communicate. However, the host without *Vesper* running will not know the state of the link they use for communication. Therefore, running *Vesper* on all the hosts in the network is necessary to ensure that all links are trusted. Another approach presented is to install *Vesper* on the network gateway, since this will secure inbound and outbound traffic of the LAN.

The conclusion done by [MAB18] states that MitM attacks are a great threat to LANs. However, the proposed technique can detect both end-point, in-point, and in-line attacks. Other attacks such as bypass, replay, and DoS attacks in addition to spoofing are also detectable using *Vesper*. Although *Vesper* works great for detecting MitM attacks, they state that it should only be used as an additional line of protection [MAB18], and secure protocols should be the main protection used.

The similarity to this thesis is the type of network studied. [MAB18] proposes a methodology to use in a LAN, such as the methodology proposed in this thesis. These measurements are then used to plot PDFs based on the results gathered from the RTT measurements. This shows that plotting RTT as a density function may be an appropriate technique. What distinguishes the studies is the use of impulse reposnses and type of equipment used. The method proposed in [MAB18] uses designated hardware to detect if a MitM attack is happening in the LAN.

3.5 Cing: Measuring network-internal delays using only existing infrastructure

[AGR03] is a study from 2003 that uses only existing infrastructure to measure delays in networks. Cing is a tool built to examine ICMP timestamp probes. [AGR03] looks at the possibility of measuring per-link internal queuing delay only by using existing infrastructure. One of the main activities is to examine ICMP timestamp probes. This is done by a tool they call "Cing."

Cing is a tool to measure internal delay in a network using ICMP timestamp probes in routers. Two assumptions have been made by [AGR03], and the accuracy of Cing depends on the assumptions made. The first assumption is that "back-to-back" packets experience the same performance and behavior in the shared network path. The second assumption is that the ICMP timestamp probes have an indistinguishable behavior from the UDP and TCP packets. Thus, the method proposed in [AGR03] may also be applied to other packets as well. The first assumption is stated as true in most cases, as it appears that the differences in queuing delay are negligible. However, within a single router, the time to generate ICMP packets varies, which can lead to a misinterpreted queuing delay. In general, more than 90% of the routers used in [AGR03] introduce this type of distortion in less than 10% of the time.

[AGR03] states that it is possible to accurately measure the queuing delay directly using only existing infrastructure. Here, one-way timestamp measurements were used. The main components of [AGR03] is the queuing delay in internal links, and RTT. There are some interesting results related to the preparation of [AGR03]. One of the stages of preparation is to check the accuracy of the router probes. They want to find out whether ICMP packets that are handled by the slow-path of the routers will get additional delay added. If that is the reality, then the measurements are biased. A study done by [ABF+00] states that routers do not add noticeably delay to ICMP packets when comparing different packet types. This is a relevant result for this project. In [AGR03] does its own study to prove the statement in [ABF+00]. [AGR03] shows that it is possible to use existing infrastructure to do measurements in a network and draw a valid conclusion.

3.6 What is new in this thesis compared to previous studies?

As stated earlier, the main focus of this study is to use non-designated hardware and software to detect a changed network path or changes in the network path. Thus, when identifying what the new elements of this study are, it is to use non-designated hardware and software to measure RTT in a network. The hardware used is a computer running Windows operating system described in more details in Section 5.2, and the software is a pinging tool for Windows machines. As seen in the related work presented, the hardware used in the studies is always designed to perform a specific task. Here, the hardware used has multiple purposes. The software used is *HrPing* which is described further in Section 2.4.1. The reason for choosing the HrPing tool is to obtain more accurate measurements and functionality than the original pinging tool on Windows machines. In summary, the new element of this study is to have a personal computer running an ICMP pinging tool to detect delay anomalies in a network.

Chapter 4
Methodology

This chapter presents the concepts of the methodology used. First, the network used to collect data is illustrated. Then, a short evaluation of how to connect to the network to get the most accurate result is presented. Following are different methodologies for measuring the RTT given. Lastly, the statistical concepts used for this thesis are discussed.

4.1 How to execute the RTT measurements?

The data collection for this study consists of collecting RTT measurements in a network, which are presented in Section 4.4. To study the differences in the RTT measurements, different parameters are studied. The parameters chosen for this study are the number of hops in a network path, time of the day, and the packet size used.

When doing the measurements, it is crucial for the results that only one of the parameters is studied at a time. If the time of the day is studied, it is essential to send the same packet size for every measurement. If not, it is unknown if the results are due to the different times of the day or different packet sizes. Also, if measurements using different packet sizes are studied, this should be done at the same time to avoid the timing of the measurements affecting the results. However, if the results show that the packet size affects the measurements RTT, these studies can be performed at different times of the day to see if the packet size affects the results, regardless of the time of day. This applies when studying different network paths as well. The concrete setup of the measurements is described further in Section 5.1

4.2 Statistical concepts

4.2.1 Distribution functions

PDF is a way to illustrate the probability that a continuous variable is observed with a value within a particular range. For a PDF, the probability is given by the area under the function and above the x-axis. A PDF is non-negative. The process of estimating a PDF can be challenging when the underlying random process described by the function is unknown [FJ18]. A particular functional form is considered out of convenience, especially when the data is limited. If there are specific features in the data, a method that requires a superposition can be used. However, expert knowledge is required to estimate the PDF. To solve situations where the PDF can be difficult to estimate, a non-parametric estimation can be used. KDE is a such non-parametric estimation and is a common way to estimate PDF [FJ18].

A KDE is used to create a smooth function that replaces a histogram and represents the consensus of the data points that exist within local binned regions [FJ18]. The function is made from a linear combination of kernel functions. KDEs are mostly selected to be Gaussian distributions, also known as normal distributions. This may lead to the smoothing of sharp features in the "real" PDF. This is a drawback of using KDE. However, the "smoothing" variable, also known as the bandwidth, can be adjusted to get the desired amount of sharp features [Węg18]. The bandwidth will affect the fitting of the function thus, how smooth the KDE is. A small bandwidth can lead to the function being overfitted, making the function have too sharp features. A too large bandwidth can lead to underfitting, which means that the features of the function are too smoothed out and may not be visible [ZW09]. Therefore, the bandwidth must be considered when creating the KDE function.

Both PDF and KDE are density functions. The method for analyzing these graphs is crucial to get the correct results. Figure 4.1 shows how a KDE may look like. The x-axis represents a data point from the data set, while the y-axis represents the probability density. Since the y-axis does not represent the actual probability, it can be greater than one. However, the area under the graph should be equal to one. To find the probability of an event occurring between two points, the integration between these two points must be calculated [Aar14].

4.2.2 Creating a KDE

A KDE based on a Gaussian kernel is created by creating an individual kernel for every data point in a data set. A kernel has a different meaning for different statistical applications. For example, when talking about Bayesian statistics, it is the form of the density function []. In non-parametric statistics, a kernel is a weighting function, where a weight function is used to give some elements more "weight" than other

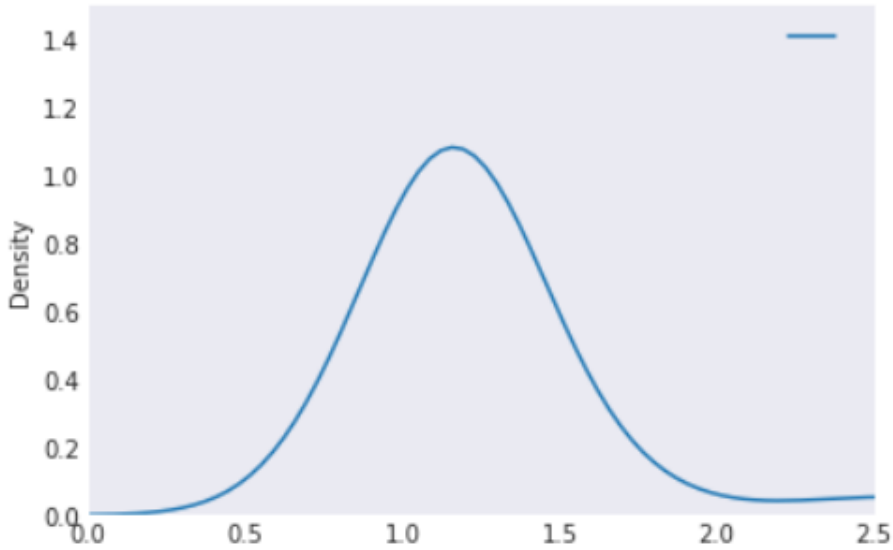


Figure 4.1: Kernel Density Estimation

elements in the same data set. The KDE is used to visualize the sum of all kernels. In an area where there are many kernels, the KDE will illustrate this by giving these values a higher density compared to an area where there are fewer kernels [Węg18] [Sil86]. This area will have a lower density. The way in which KDE gives some values a higher density is illustrated in Figure 4.2.

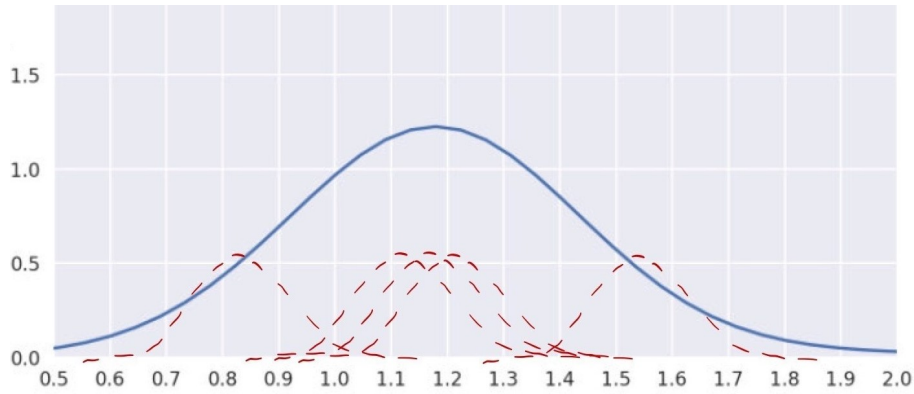


Figure 4.2: Illustration of how a KDE is made

When looking closely at the green plot in figure 4.3 the plot has negative x-values, even though it is known that there are no negative x-values in the data set. The

reason for this is due to how Python creates a density plot. A density plot is a variation of a histogram [Sil86], but gives a more precise location of which value has the highest density. Density plots are made by dividing the values into the data set into "bins" [Sil86]. Depending on the width of the bins, when having values close to zero in the data set, it may seem like the data set contains negative values, since the bin may intersect the x-axis.

4.2.3 Standard deviation

Standard deviation measures the amount of variation of the values in a data set. A data set with a standard deviation more significant than in a different data set means that the difference between the smallest and largest values is wider in the data set with the highest standard deviation [Insa]. A data set with a standard deviation in the range of 0 to 2 will have a more distinct maximum point when being visualized in a density function than data with a standard deviation in the range of 0 to 4. Figure 4.3 is an example of two different plots with quite different standard deviations. The bandwidth is the same for both plots. The data set illustrated by the blue plot in Figure 4.3 has a more defined extreme point, which means the values are closer in range compared to the data illustrated by the green plot.

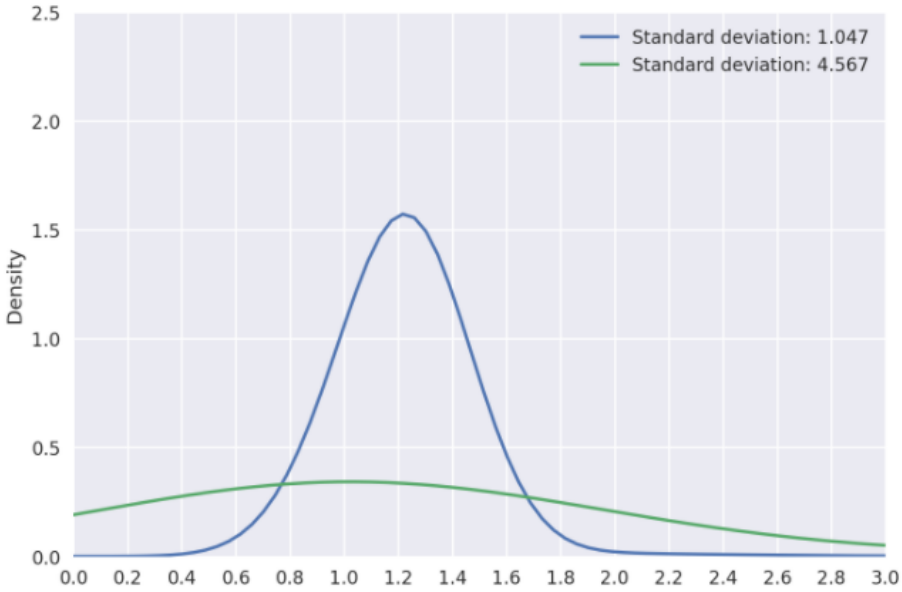


Figure 4.3: Compering plots with different standard deviation

4.2.4 Confidence interval

When talking about a confidence interval in statistics, it is meant as a measure of how reliable an estimation procedure is. A small confidence interval indicates that the estimates are reliable, while a larger confidence interval may indicate more uncertain estimations [Bjø22]. The confidence interval gives a lower and upper limit for the size of the estimation. If two numbers a and b have values such that the stochastic interval (a, b) covers the unknown size is 95%, then the interval (a, b) is said to be a 95% confidence interval for that size [Bjø22]. A common way to express a confidence interval is to use

$$\text{sample mean} \pm range$$

or

$$[min, max]$$

For this study we assume that the parameters being estimated is approximate normal distributed due to the large amount of data. [BGS07] derives the mathematical formula for a $100(1-\alpha)\%$ confidence interval for a normal distribution. The result is:

$$\text{Confidence interval} = \overline{X} \pm t_{n-1,1-\frac{\alpha}{2}} \frac{S}{\sqrt{n}} \tag{4.1}$$

[Insb]. For this confidence interval $1 - \frac{\alpha}{2}$ fractile belongs to the t-distribution with $n-1$ degrees of freedom. Here, $\overline{X}$ is the sample mean, $t_{n-1,1-\frac{\alpha}{2}}$ is a number found in a statistical table, S is the standard deviation of the sample, and n is the sample size. For every data set illustrated in Section 5.3, a $100\%(1-0.05)$ confidence interval is found. When looking into any statistical tables containing information about the t-distribution, looking at degrees of freedom equal to ∞ since $n = 1000$ and $\alpha = 0.05$. This gives that $t_{n-1,1-\frac{\alpha}{2}}$ is 1.960 [Pre02]. These results are used in Section 5.3.

4.3 Creating density plots using Python

Python is one of the most popular programming languages used today. This is because it is a language that can be used for many different purposes and is easy to learn for beginners to programming. Due to a large number of users, there exists a lot of documentation [Pyt22]. Many users have led to the development of different libraries for additional functionality. For this thesis, Python is used to create density plots made from RTT measurements due to the existing libraries and documentation.

For this thesis, the module *re* and the libraries *Matplotlib* [mat22],*Pandas* [pan22] and *NumPy* [Num22] have been used. Re, which stands for *regular expressions*, is a module that is made for matching text patterns. Here, the module helps sort out every RTT, which are the relevant data in this study. Thus, the data set becomes easier to work with because superficial data are removed. Then the Pandas library is used to create a DataFrame. Pandas is a tool built on top of Python and is created to analyze and manipulate data. A DataFrame is a data structure that organizes the data into a two-dimensional table.

When the data has been transformed to be a DataFrame, it can then be processed. This is where Matplotlib and NumPy are used. Matplotlib is the Python library for visualization of data by creating plots. Numpy is a mathematical extension of Python containing numerous mathematical functions, e.g., a function for computing the sample mean. The Python code is available in Appendix A.1.1 and Appendix A.1.2. Several plots can be presented in the same diagram, making it easier to compare the different results. There exist similar libraries to the ones chosen. For instance, is *seaborn* a library made for statistical data visualization. The chosen libraries are used due to the amount of documentation available online.

4.4 An explanation of the network used

The experiment is carried out using the network located at the chosen campus. Using this network is due to investigate whether the methodology presented works, which requires a real-life network. To make a simulated network as realistic as a real-life network, many different parameters must be implemented. This includes every parameter that affects the RTT in the switches and routers. Not all of these parameters are known. This includes the switching capacity or the forwarding performances of the switches. These parameters may not be the same values for the same type of network devices. Therefore, a simulated environment is not possible to be set up realistically. Thus, a real-life network is chosen because of the complexity of creating a realistic simulated environment.

A brief explanation of the network is presented, focusing on the relevant aspects related to the experiment. The network is divided into five areas, and each area has an area router connected to the edge switches. There are 15 to 20 edge switches for each router. For every switch, the number of connected users varies from 10 to 1000. Each area router is connected to a router which allows it to be connected to all other area routers in the network shown in Figure 4.5. Thus, if a host in area one wants to communicate with a computer in area two, the data packets go through two switches and three routers. Figure 4.4 shows a simplified version of an area in the network containing the area-router, edge switches, and hosts. Figure 4.5 shows how two areas

are connected through a router. In this study, we have chosen two locations to do measurements, located in two different areas.

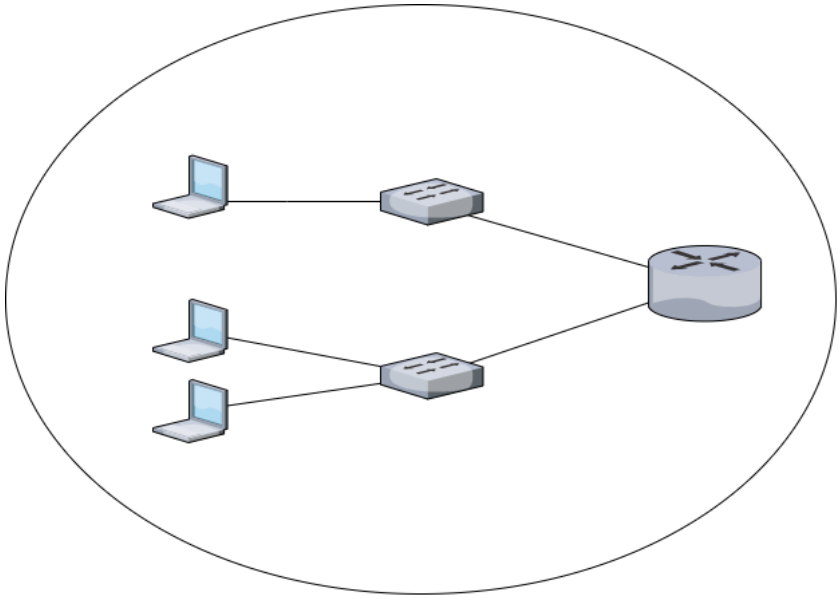


Figure 4.4: Simplified version of an area in the network

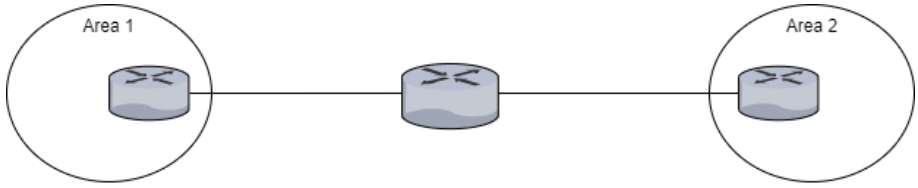


Figure 4.5: Two areas being connected

4.5 Ethernet frames

When using ICMP to measure RTT on a network, several other protocols are used to send request and response messages through the network. The ICMP messages are transported as payload. First, an IP header is added to the ICMP packet in layer 3. Then the IP packets are moved down to layer 3 in the OSI-model, and Ethernet frame headers are added to the data packet, making the IP packet being transferred as payload through the physical links in the network [KR17]. Therefore, when we are talking about 28-byte, 1400-byte, and 2000-byte packets throughout this study, the size of the IP packets are being referred to. To find the Ethernet frame size, add the sum of the Ethernet frame, the payload and the Frame Check Sequence, which is $14bytes + payload + 4bytes$. The minimum frame size is 64 bytes, making the minimum payload size $46bytes$ [16]. When the payload is less than the minimum frame size, a sufficient amount of $0x00$ padding is added to the payload to meet the requirement [16].

4.5.1 Why IP payload size affects the RTT measurements

When studying RTT in a real-life network, it is essential to consider the different limitations that exist, which may affect the measurements. In standard 802.3 [MC18], it is stated that the maximum Ethernet payload size is 1500 bytes. If the payload exceeds 1500 bytes, the packet may be fragmented, depending on whether the switches and routers support *jumbo packets*. It is the Maximum Transmission Unit (MTU), which is the unit that decides when the packets are fragmented. A Jumbo packet is an Ethernet frame where the payload is more than 1500 bytes. If only one network device does not support jumbo packets, the packets will be fragmented. The MTU for the computer used is set to 1500 bytes. Thus, from this information we know that when the IP packets are set to 2000 bytes, the computer will fragment the Ethernet frames. Research question 2 aims to gather information about the conditions that must be present to detect changes in a network path or a changed network path. Thus, analyzing how much fragmentation is affecting the RTT is included in the experiment presented in Chapter 5.

Fragmentation of Ethernet frames

When an Ethernet frame is fragmented, the data packet is divided into multiple smaller packets when the network do not allow the size of a current data packet [PRO81]. An internet packet can have a flag set to "don't fragment", but then the packets will be discarded instrad of deliverd to its destination if fragmentation is needed to deliver the packet [PRO81]. Figure 4.6 shows a simplification of how the Ethernet frames are divided when the payload exceeds 1500 bytes. As shown in Figure 4.6, the Ethernet frame is divided into two Ethernet frames. Thus, there are

two data packets sent through the network. This is affecting the RTT since we have to wait until both packets are registered by the computer.

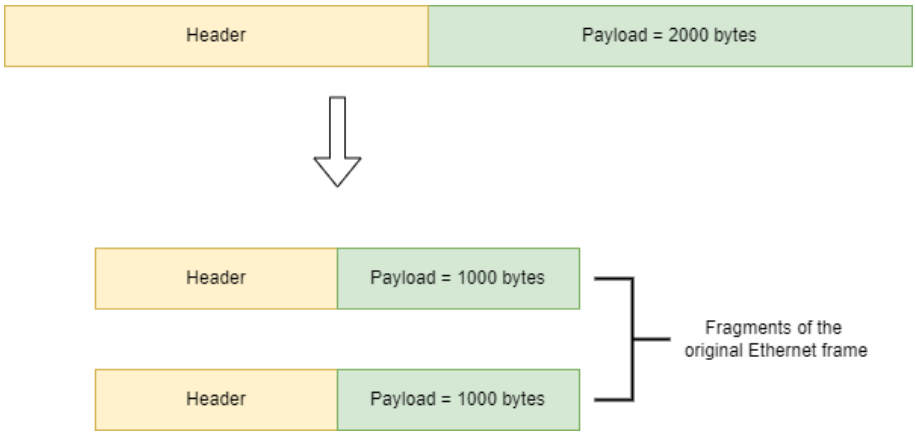


Figure 4.6: Fragmentation of an Ethernet frame

4.6 Execution

The experimental work presented in Chapter 5 is executed in the following manner:

1. First, a preliminary study was conducted. The study consisted of finding out the differences in connecting to the network using Ethernet and WiFi. From the preliminary results, the decision to connect using Ethernet was made.

2. To study two different paths in the network, we had to find two locations in the network that would give different paths. The destinations that were being pinged were already specified. It was necessary, as described further in Section 5.1.1, for the points of connection to have the same properties in terms of capacity.

3. The third task consisted of creating a method to store the results of a pinging session. This task was done using HrPing, which already had pre-made functionality for storing the results in text files.

4. The text file from the first point contains all the information from the pinging session. The relevant information to be used for further analysis was the RTT. Therefore, Python code was written to select only the RTTs from the result file. See Appendix A.1.1.

5. Then, the python code for creating density plots was made. The code also includes functions to find the standard deviations and sample mean of the data sets used to create the density plots. See Appendix A.1.2.

6. The time slots for the pinging sessions to be executed were then decided. One of the goals of the study was to study how the measurements RTT were affected by the time of day. Thus, two different time slots were set to execute RTT measurements. The time slots are described in further detail in Chapter 5.

7. The next decision consisted of deciding what packet sizes should be used for the different pinging sessions. From Section 4.5.1 we know that MTU decides when packets are fragmented. Thus, packets close to this limit were chosen, namely 1400- and 2000-byte IP packets were used. The smallest packet size possible was also used, namely a 28-byte IP packet.

8. Then, deciding what type of destination to be used was considered. Both a server and a router were chosen as destinations to compare how RTT measurements are affected by different types of destinations.

9. The last decision was to decide on the number of pings to be executed for every situation. Both 200, 500, 1000, and 2000 pings were tested. The number of pings used for each pinging session was chosen to be 1000. This is further described in Section 5.1.3.

10. The pinging sessions were then executed and analyzed.

4.7 Limitations

As stated in Section 2.1, technologies such as firewalls, encryption, and the use of VPN networks are increasing the processing time of the data packets. Given that the processing delay added was the same for every packet in the networking devices processing, there would be no problem using this methodology, for example, in VPN. However, the processing delay is not constant and varies for every packet. This makes it difficult to analyze the RTT, since there is one more varying delay in the path. The processing delay can create significantly large differences in RTT depending on the processing time at each network device.

As briefly discussed, the methodology presented cannot be performed on a large-scale network such as the Internet. The reason is that many processes are running on the devices that make up the Internet, making the delay variations vary by multiple milliseconds. Additionally, if a server located in the USA is pinged, we have no idea how many devices the path consists of. Therefore, a new hop in the network will be invisible when the delay variations are more significant than the delay added. We do not control whether a device is replaced or not in the Internet, making it challenging

to detect delay anomalies. Also, we do not know the exact topology of the paths pinged due to Traceroute only detecting devices at layer 3.

Chapter 5
Experimental work

In this section, the experiment is presented. The first section is about the setup of the experiments. Following are the results from the experiments presented. Here, we focus on finding out what is affecting the RTT measurements, such as load in the network, change of network paths, and packet sizes. Throughout, both "the server" and "the router" are referred to. When referring to "the router," it is the area router located in Area 2. When referring to "the server," it is a server located in Area 2.

5.1 Setup

5.1.1 How to connect to the network

We want the least amount of varying delay in the network path for this study. This is because the varying delay affects the RTT measurements and their accuracy. Therefore, finding out how to connect to the network is essential to determine which connection to choose to avoid as much varying delay as possible. A preliminary study was conducted to choose the type of connection that satisfies the requirements. There are several ways to connect to the network, either WiFi or wired Ethernet.

The preliminary study was carried out by pinging the same server, once connected using WiFi and once connected using Ethernet. The ping measurements were performed in the same period, using the same IP packet size. We wanted the pinging sessions to experience as similar delay in the network as possible. For the study, one thousand pings were executed. The reason was to see if there were any patterns in the measurements. When studying the results, the RTT with the standard deviation for the measurements were observed. When observing the results from the different pinging sessions, using WiFi resulted in a much larger standard deviation than using Ethernet. The standard deviations for these exact measurements presented in Figure 5.2 and 5.1 are, respectively, 0.130 and 16.140. When using Ethernet the standard deviation is approximately 124 times larger than when using WiFi.

It is common knowledge that WiFi has milliseconds of jitter due to being a broadcast medium. Jitter is variations in delay and can be caused by different loads in the network, packet sizes used, queuing delay [SJ95]. The study was conducted to illustrate the difference. WiFi being a broadcast medium [MC08], means that resources are shared, and one transmitter can send data to multiple recipients simultaneously. When connecting using Ethernet a designated link is used. When connecting through Ethernet, resources are not shared [16], and therefore RTT measurements are more precise compared to measurements done using WiFi. Therefore, the connection through wired Ethernet is the preferred method of connection because it gives more accurate measurements than when the connection is via WiFi.

When choosing the locations in the network to ping from, it is crucial to ensure that the capacity is the same at both locations. A link with a capacity of 100 Mbps will not be able to handle as much traffic as a link with a capacity of 1 Gbps, making the RTT measurements longer in the smaller capacity link. If we first use a connection point of 1 Gbps and then switch to 100 Mbps, it will look like something has happened in the network, e.g., a path change, even though the same path was pinged due to the different amount of capacities. Thus, having the same capacity in the links used is a condition for a computer to recognize a delay anomaly.

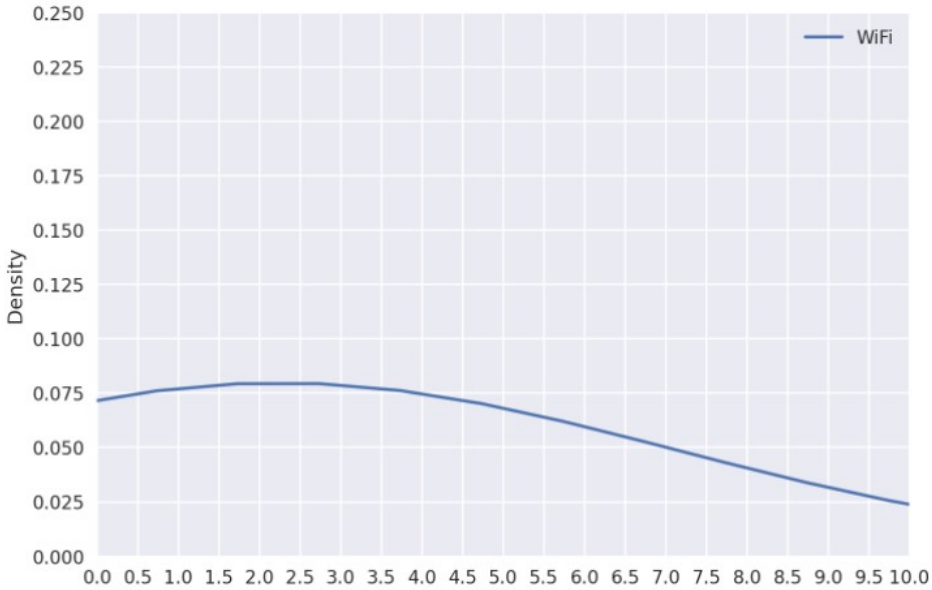


Figure 5.1: Distribution of RTT measurements using WiFi

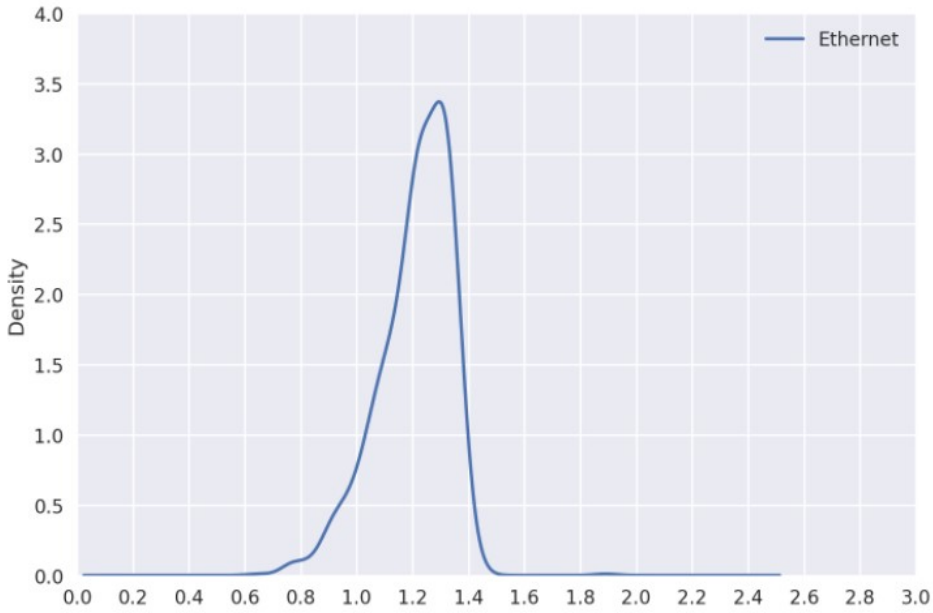


Figure 5.2: Distribution of RTT measurements using Ethernet

5.1.2 The topology of the network paths

Two different Ethernet connection points are chosen as locations for RTT measurements. These locations are in two different areas in the network, namely in Area 1 and in Area 2. The exact locations are chosen because they have the same capacity and give different network paths to pinged destinations in terms of the number of hops.

Figure 5.3 illustrates the network path between Area 1 to the server. In contrast, Figure 5.4 illustrates the network path the the Ethernet frames travel when pinging the server from Area 2. Traceroute was first used to get an overview of how many routers were located between the different locations. The IT administrator was then contacted to get the exact structure of the network. This is because network switches are not present in the traceroute because they are part of layer two in the OSI model; described in Section 2.2. Figures 5.5 and 5.6 illustrate the corresponding network paths when pinging the router.

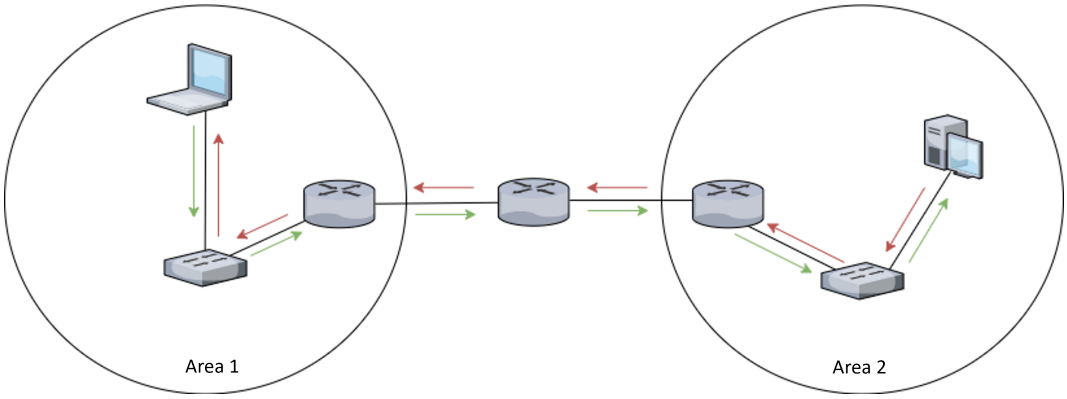


Figure 5.3: Network path when pinging the server from Area 1

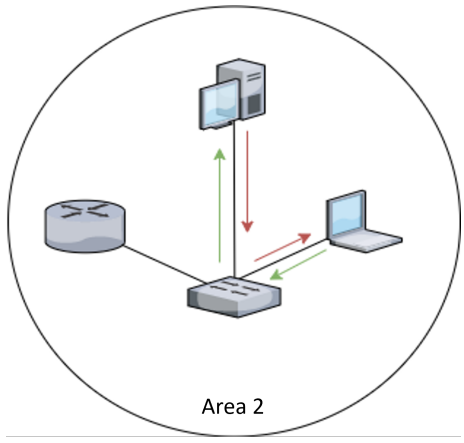


Figure 5.4: Network path when pinging the server from Area 2

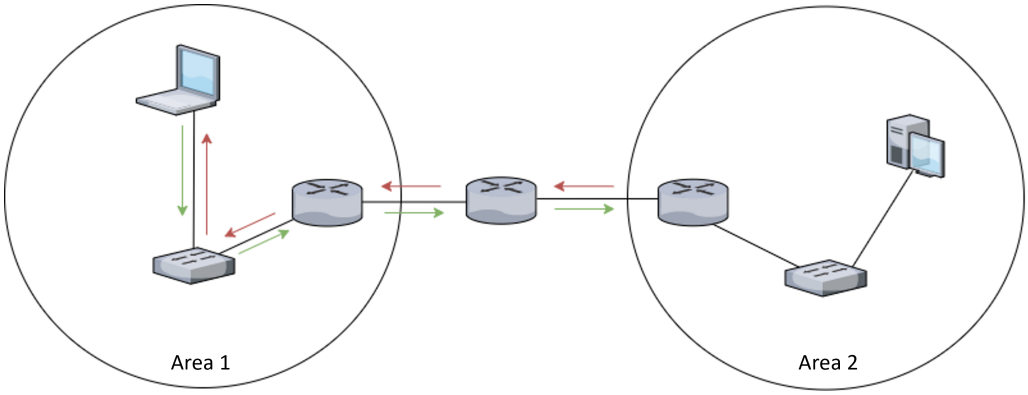


Figure 5.5: Network path when pinging the router from Area 1

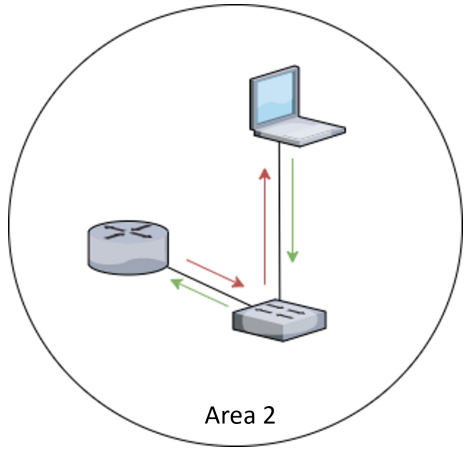


Figure 5.6: Network path when pinging the router from Area 2

5.1.3 Performing RTT measurements

When performing the RTT measurements, the computer must run the least amount of processes possible to make sure the RTT measurements are getting as accurate as possible. This can be justified both theoretically and also through observations. The theoretical reasoning is that the Central Processing Unit (CPU) only can perform one task for each clock signal. The clock signal coordinates the tasks for the CPU. The more processes run by the computer, the longer it takes for a data packet to be 'clocked in' by the processor, as it has to wait until its turn in the sequence of processes to be carried out [WS]. During the setup phase, RTT measurements were done, having many processes running, meaning running a web browser and a code editor, and the computer only running the necessary processes. The standard deviation when many processes were running was approximately two times larger than when only necessary processes were running.

In addition, the Random Access Memory (RAM) on the computer can affect the RTT measurements. If RAM does not have sufficient memory for the CPU, the computer can slow down. Thus, the data packets may be "clocked in" later than if the RAM were less full. This observation was made once during the preliminary study. The computer became very slow when doing other tasks, such as writing. Thus, the computer was restarted before the pinging sessions were started to avoid such situations. The lack of enough RAM is often due to having many other processes running at the same time [NAGY03] or due not emptying the RAM between sessions. Thus, these requirements go hand in hand.

When the computer has restarted, the terminal was opened as *administrator*. The terminal chosen for this experiment is the standard Windows terminal. If the terminal is not running as an administrator, an error message is displayed when the pinging program is running. The error message indicates that a socket is not accessible due to access authorization.

When the computer is connected to the Ethernet at the desired location, the ping settings are ready to be configured. For this experiment, several situations are being studied to see if they affect the accuracy of the measurements. The different cases are as follows.

- Does the time of day affect the RTT measurements?
- Does the packet length affect the measurements?
- Does the type of destination pinged affect the RTT?

The HrPing, described in Section 2.4.1, are configured as follows:

hrping <ip address of destination> -n <number of pings> -L <size> -F <location to store output>

We are using a Ping tool for this experiment because we wanted to detect changes in delay caused at layer one, layer two, or layer three. As described in Section 2.4.2, Traceroute can only detect changes in layer three. Additionally, Traceroute for Windows measures only RTT in milliseconds, making the measurements less accurate compared to the measurements using microseconds.

The IP destination address is set to the server or the router in Area 2. The number of pings being performed has been developed to *1000*. Pinging sessions of 500 samples and 2000 samples were also tested. However, when using 500 samples, the data was very affected by having packets experiencing less queuing delay than most packets that experienced more buffering. However, the data sets had very similar results when using 1000 and 2000 samples. Thus, we chose the number of samples to be 1000 since it made the process quicker. When 1000 samples were used, the pinging session took approximately eight minutes, while for 2000 pings it took 15 minutes.

5.2 The non-designated hardware used

As stated in Section 5.1.3 the processor and RAM affect the RTT measurements. Therefore, knowing how much processing power is running on the computer is needed to determine whether it can be reduced. The computer is a laptop made by ASUS. The machine is running Windows 10 and is a 64-bit operating system. The size of the RAM is 8 GB. The processor used is an Intel Core i5-7200U CPU. 91 other processes are active in addition to the pinging during sessions. The total processing power used when the HrPing runs is 3%, and the memory used is 45%. When similar studies are conducted, this information can be used to decide what types of hardware can be used for the methodology presented.

5.3 Results

In this section, the results of the experimental work are presented. The results are divided into two main parts. In Section 5.3.1, the main goal is to illustrate how different packet sizes affect RTTs when the measurements are done in the same path. Section 5.3.1 also attempts to answer how the time of day affects the results. Thus, RTT measurements are performed between 09:00 AM - 09:30 AM and 06:00 PM - 06:30 PM. Section 5.3.2 focuses on detecting different network paths and what packet size should be chosen to easiest detect a changed network path. In both sections, we do the measurements towards the server and the router to illustrate how the type of destination affects the results.

Throughout section 5.3.1, the blue plot illustrates the data sets with IP packet sizes of 28 bytes, the green plot data sets using 1400-byte IP packets. The red plot a data set consisting of 2000-byte IP packets. In Section 5.3.2, for each figure, the blue plot represents a path consisting of two hops, while the green plot illustrates paths consisting of four or six hops. This information is also presented in the upper right corner of every plot. A table containing information about each data set is specified for each figure presented.

For each table, the column named *RTT (ms)* is the RTT with the highest density in the data set analyzed. It is this RTT we are using as a basis for comparisons of data sets. The reason for this choice is that we want to study the RTT value most data packets experience through the network path. The PDFs illustrates the value by being the maximum point of the respectively plot. The sample mean is affected by some packets experience very little delay or very high delays. Thus, the sample mean do not always represents the RTT that is most common for the data packets to experience.

In Section 5.3.1 and Section 5.3.2 we are comparing data sets, using the RTT, standard deviation and the 95%-confidence intervals. The standard deviation illustrates the spread of the RTT values in the data set. This value is desired to be as small as possible, meaning that the values in the data set are close in range. The column with marked as *95%-confidence interval size (ms)* is showing the range of the interval of what is the expected value μ for the RTT in a specific path.

5.3.1 Comparing how packet sizes and time of the day are affecting the RTT

Pinging the server from Area 1

The following figures, Figure 5.7 and Figure 5.8 are made to illustrate if and how the time of day a pinging session is done affects the results. The path used in Figure

5.7 is: PC Area 1 => switch => router Area 1 => router connecting two areas => router Area 2 => switch => server in Area 2. Thus, the host is a computer located in Area 1, and the pinged destination is the server located in Area 2. The number of hops is six.

Figure 5.8 illustrates the same path as in Figure 5.7. The difference between these plots is that the pinging sessions are from different times of the day. There are two periods being studied, namely 09:00 AM - 09:30 AM and 06:00 PM - 06:30 PM. These two periods are chosen because it is assumed that they have different loads in the network. The period 09:00 AM - 09:30 AM is assumed to have a larger load than 06:00 PM - 06:30 PM due to a larger amount of students and employees being present at campus, making the activity in the network increase. Figure 5.7 illustrates the data collected between 09:00 AM - 09:30 AM, and Figure 5.8 illustrates the data captured between 06:00 PM - 06:30 PM.

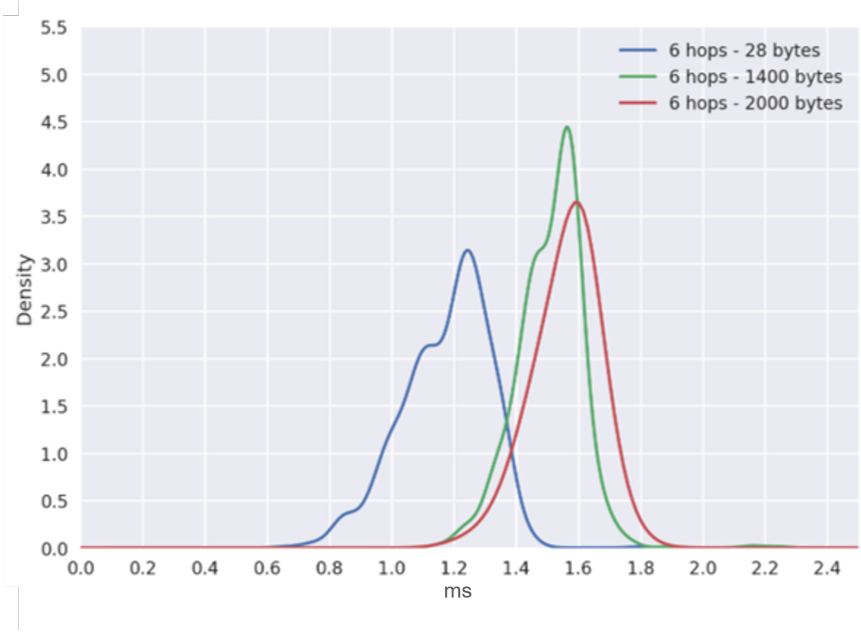


Figure 5.7: Pinging the server in the time period 09:00AM to 09:30 AM, six hop path

Packet size (in bytes)	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
28	1.23	0.139	1.174	±0.009
1400	1.57	0.122	1.512	±0.008
2000	1.60	0.250	1.560	±0.016

Table 5.1: Results from figure 5.7

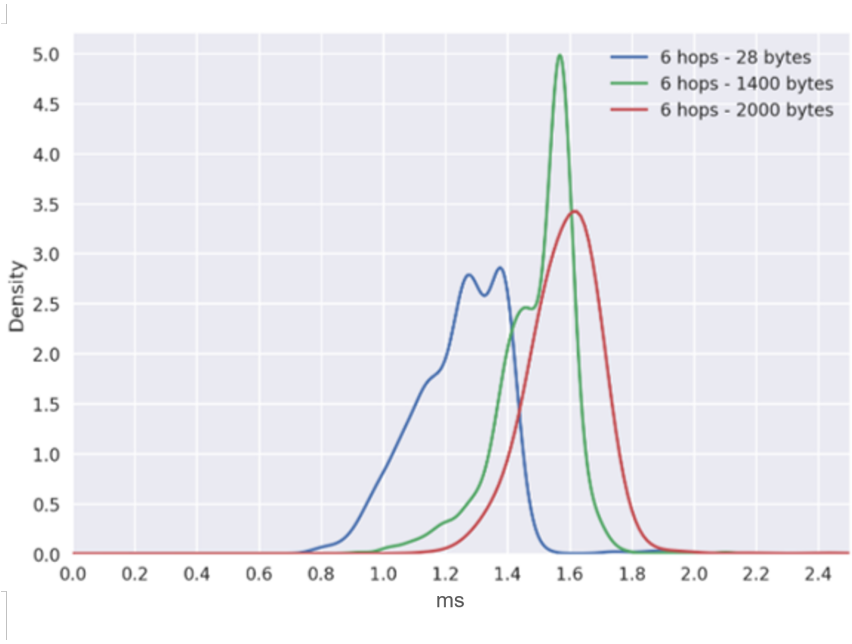


Figure 5.8: Pinging the server in the time period 06:00PM to 06:30 PM, six hop path

Packet size (in bytes)	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
28	1.25 and 1.38	0.146	1.244	±0.009
1400	1.57	0.123	1.492	±0.008
2000	1.60	0.242	1.591	±0.015

Table 5.2: Results from figure 5.8

Comparing observations from Figure 5.7 and 5.8

First, studying the RTT measurements we see that when the IP packet size is set to 1400 bytes or 2000 bytes, the RTT is the same in both periods. Due to the data made with 28-byte IP packets having two peaks, it is more difficult to conclude. However, both peaks have a RTT larger during 06:00 PM - 06:30 PM. The increase in RTT is respectively 0.016 and 0.12 times larger than during 09:00 AM - 09:30 AM.

Looking now at the accuracy measurements for each IP packet size used, the standard deviation is the smallest in both periods when using 1400-byte packets. This shows us that the spread in RTT values is the smallest when using a packet size of 1400 bytes. We also see that the differences in standard deviation when the period changes is the smallest when using 1400 bytes. From these results, using 1400-byte IP packets gives the most stable results.

When studying the 95%-confidence interval, we see that the values are similar in both periods when 28-byte and 1400-byte IP packets are used. It is only when the 2000-byte IP packets are used that the 95%-confidence interval changes. The change is a reduction of 0.002ms. Comparing how the 95%-confidence intervals are in comparison to each other, using 28-byte compared to 1400-byte IP packets, the interval is 0.2 times smaller when using 1400-byte packets. When comparing 28-byte to 2000-byte, the interval increases by approximately 0.5 times when using 2000-byte IP packets. When comparing 1400-byte to 2000-byte, the 95%-confidence interval increases by 0.5 times when using 2000-bytes

Looking at the 95%-confidence intervals we see that the interval is the smallest when using 1400-byte IP packets and that the size of the interval so not change when the period changes. This indicates that the expected value is the closest to the sample mean in the data set. In terms of detecting a changed network path or a change in the network path, it is an advantage that the data being compared have small confidence intervals and standard deviations. Thus, using 1400-byte IP packet is the preferred packet size to use given these requirements.

Pinging the server from Area 2

The upcoming figures, Figure 5.10 and Figure 5.9 are presenting data that has been collected from the path PC Area 2 => switch => server in Area 2. Thus, the host is a computer located in Area 2. The number of hops in the path is two. Similarly to the previous paths studied, the pinging sessions have been done in two different sessions at different times of the day to study how different traffic load during the day affects the RTT. Figure 5.10 the pinging sessions is done between 06:00 AM and 06:30 PM, while the pinging sessions presented in Figure 5.9 is done between 09:00 AM and 09:30 AM.

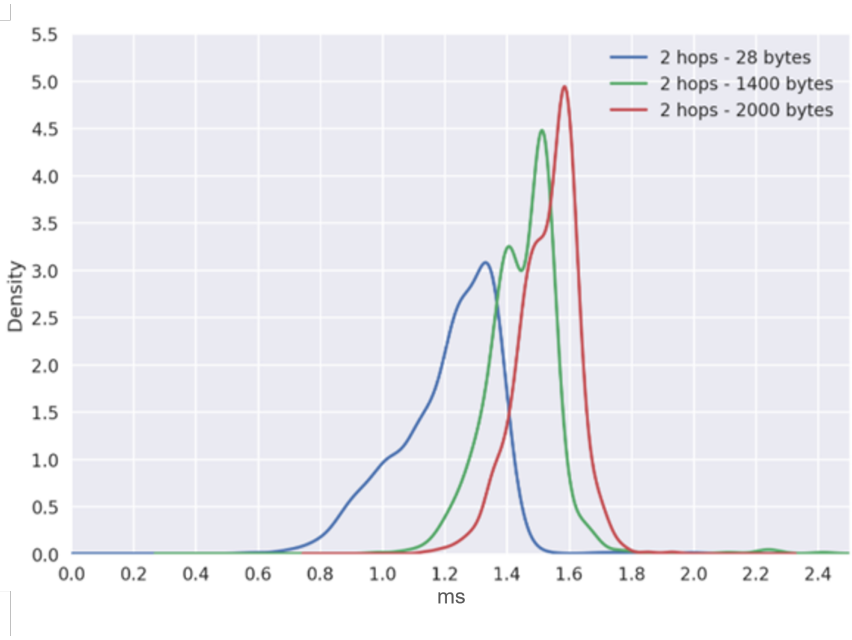


Figure 5.9: Pinging the server in the time period 09:00AM to 09:30 AM, two hop path

Packet size (in bytes)	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
28	1.35	0.157	1.210	±0.010
1400	1.50	0.120	1.449	±0.007
2000	1.58	0.095	1.531	±0.006

Table 5.3: Results from figure 5.9

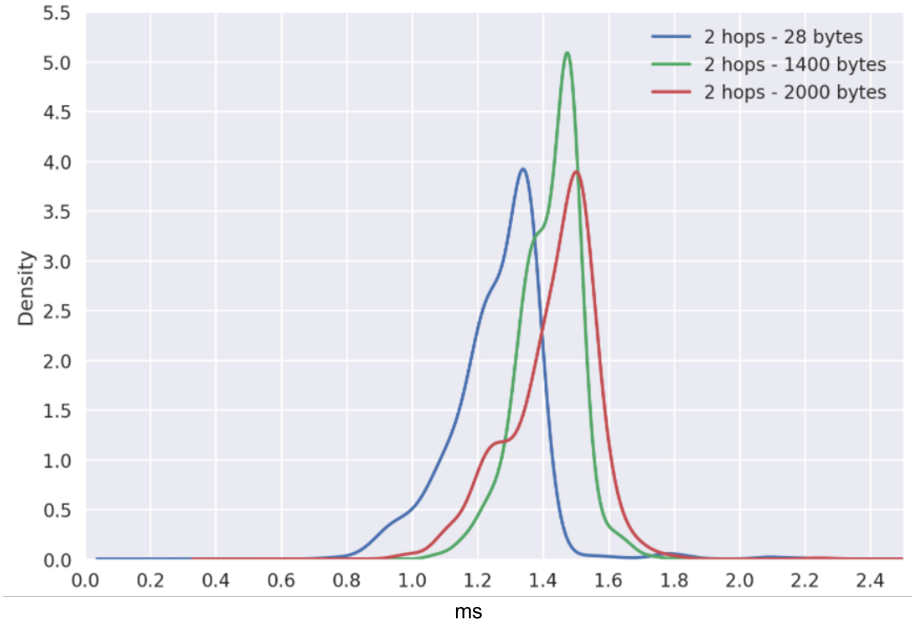


Figure 5.10: Pinging the server in the time period 06:00PM to 06:30 PM, two hop path

Packet size (in bytes)	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
28	1.35	0.138	1.259	±0.009
1400	1.43	0.094	1.259	±0.009
2000	1.55	0.132	1.430	±0.008

Table 5.4: Results from figure 5.10

Comparing observations from Figure 5.9 and Figure 5.10

The path now consists of two hops instead of six. When using 28-byte IP packets, the RTT with the highest density is the same in both periods. When the packet size increases to 1400, the difference in the RTT measurement increases by 0.04 times. When using 2000-byte IP packets the RTT is 0.02 times shorter during 06:00 PM - 06:30 PM than during the first period. Thus, here are the RTT most affected when using 1400-byte IP packets.

Studying the standard deviations for the two periods we see that the values varies significantly. In the first period the standard deviation is smallest when using 2000-byte IP packets, while in the second period using 1400-byte IP packets gives the smallest value. This gives an indication that the data set vary when the periods change. The standard deviation is the largest in both periods when using 28-byte packets.

Now considering the 95%-confidence intervals. The interval decreases 0.1 times when the period is changed to 06:00 - 06:30 PM when 28-byte IP packets are used. Similarly, when using 1400-byte IP packets, the interval increases by 0.2 times. When using 2000-byte IP packets, the interval increases by 0.25 times when the period changes to 09.00 - 09:30 AM. There is no pattern in whether the interval increases or decreases as the period changes. Comparing the size of the 95%-confidence interval in terms of each other, we see that the range is smallest when 2000-byte IP packets are used.

Pinging the router from Area 1

A similar experiment as presented in section 5.3.1 was done when pinging the router to see if any results were affected by the type of destination chosen. The results are illustrated in figures 5.12 and 5.11. The time of the pinging sessions is done due to the same reasoning as presented earlier. The data illustrated in Figure 5.11 were captured between 09:00 AM - 09:30 AM. The data presented in Figure 5.12 are made from data captured between 06:00 PM - 06:30 PM. The path used for collecting data is PC Area 1 => switch => area router in Area 1 => router connecting two areas => area router Area 2. Thus, the host is a computer located in Area 1, which is the same situation as the previous experiment. However, now the destination is the router located in Area 2.

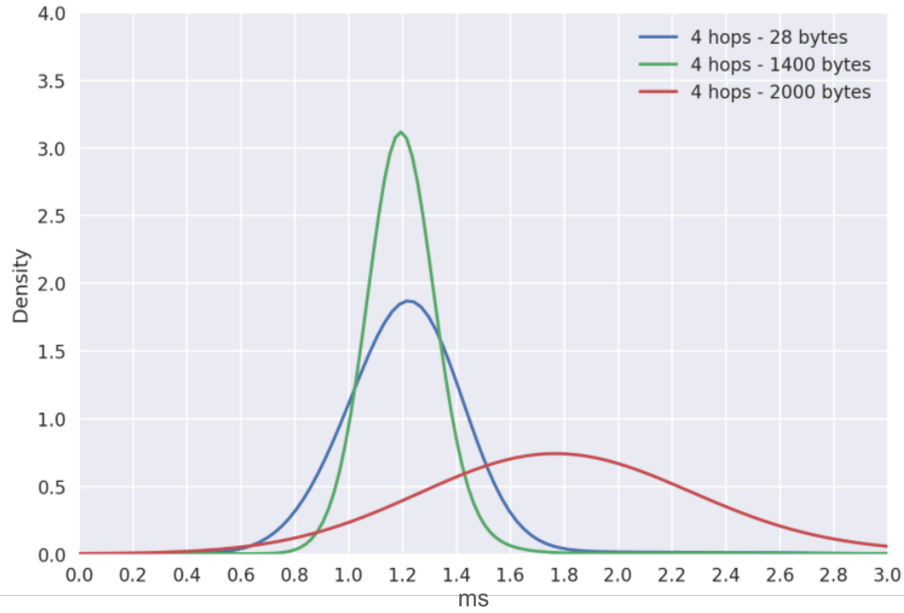


Figure 5.11: Pinging the router in the time period 09:00AM to 09:30 AM, four hop path

Packet size (in bytes)	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
28	1.22	0.599	1.219	±0.037
1400	1.20	0.336	1.213	±0.021
2000	1.65	1.609	1.209	±0.100

Table 5.5: Results from figure 5.11

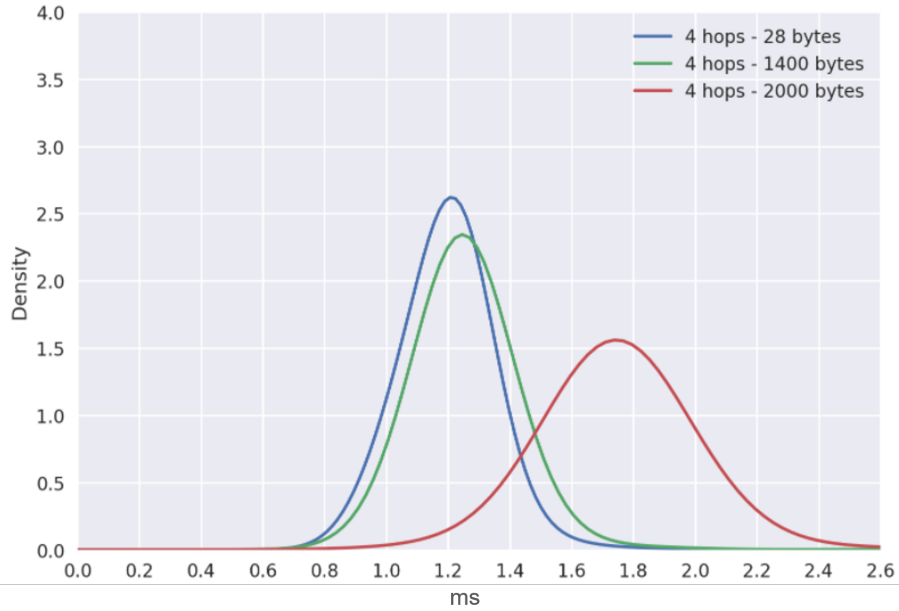


Figure 5.12: Pinging the router in the time period 06:00PM to 06:30 PM, four hop path

Packet size (in bytes)	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
28	1.20	0.414	1.219	±0.026
1400	1.25	0.572	1.300	±0.036
2000	1.75	0.864	1.859	±0.054

Table 5.6: Results from figure 5.12

Comparing observations from Figure 5.11 and 5.12

First, we see that the RTT with the highest density decreases by 0.02 times when the packet size is 28-bytes and the period changes from 09:00 AM - 09:30 AM to 06:00 PM - 06:30 PM. The RTT increases when the packet size is either 1400- or 2000-bytes by respectively 0.04 and 0.06 times from the first period to the second period.

As seen from the previous data sets the standard deviations fluctuates when the periods changes from 09:00 AM - 09:30 AM and 06:00 PM - 06:30 PM. We discover that the standard deviation decreases when using 28-byte and 2000-byte IP packets when the period changes from the first to the second period, while the standard deviation increases when using 1400-byte IP packets. Based on the values there are no packet sizes that creates data sets with a similar spread in the RTT values. The spread in values is twice the size when using 2000-byte IP packets and is therefore the least preferred packet size to use based on this data.

When studying the 95%-confidence intervals, the intervals are reduced in size when 28-byte and 2000-byte packets are used. The intervals increase when the 1400 bytes are used when the period changes to 06:00 PM - 06:30 PM. The difference in size is smallest when using 28-byte IP packets, which is a reduction of 0.018ms. In these data sets we see that the 2000-byte IP packets are most affected when the period changes.

Pinging the router from Area 2

The next figures, 5.13, 5.14, 5.9 and 5.10 are also illustrating how the time of day and the size of the packets can affect RTT. However, the location in the network has changed. The path used to collect the data illustrated in Figure 5.14 and Figure 5.13 is PC in Area 2 => switch => area router in Area 2. Here, the computer is located in Area 2 and the destination is the router. Now, the number of hops in the path is two hops, compared to earlier where the paths have consisted of four and six hops. Figure 5.13 illustrates the same path pinged between 09:00 AM - 09:30 AM, while Figure 5.14 illustrates the data collected in the period 06:00 PM - 06:30 PM.

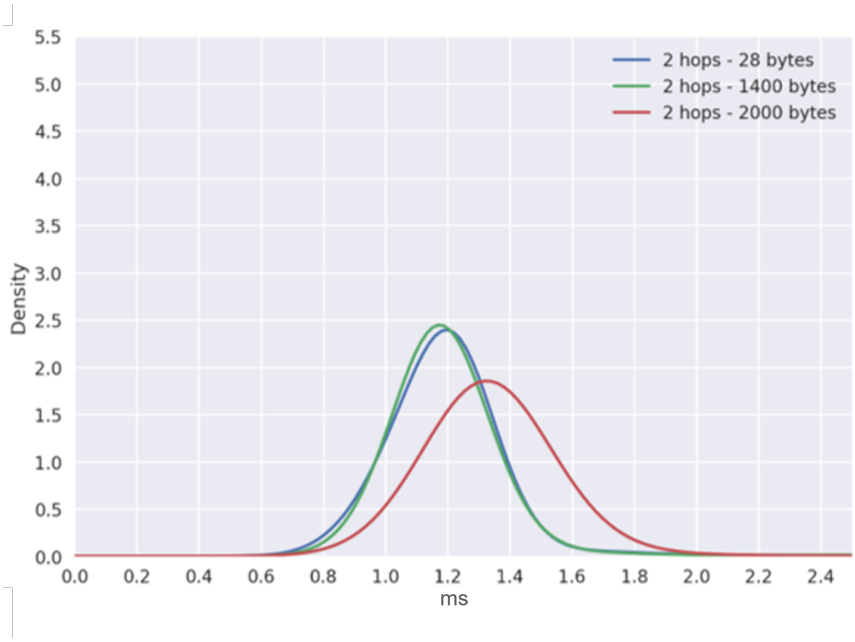


Figure 5.13: Pinging the router in the time period 09:00AM to 09:30 AM, two hop path

Packet size (in bytes)	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
28	1.20	0.486	1.221	±0.031
1400	1.18	0.539	1.248	±0.033
2000	1.32	0.745	1.427	±0.046

Table 5.7: Results from figure 5.13

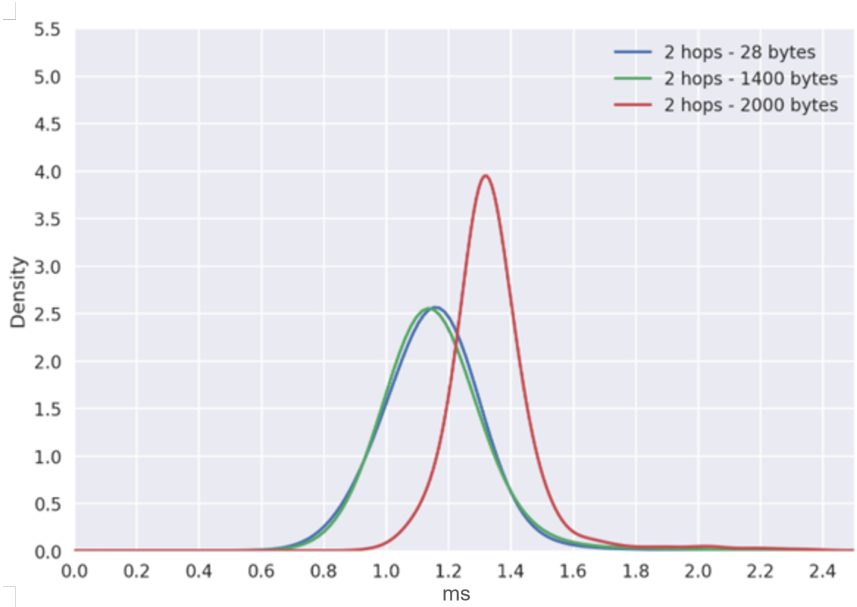


Figure 5.14: Pinging the router in the time period 06:00PM to 06:30 PM, two hop path

Packet size (in bytes)	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
28	1.18	0.437	1.179	±0.027
1400	1.15	0.496	1.197	±0.031
2000	1.33	0.215	1.350	±0.013

Table 5.8: Results from figure 5.14

Comparing observations from Figure 5.13 and 5.14

The RTT with the highest density is the same in both periods when the IP packet size is set to 2000-bytes. When using 28-byte and 1400-byte IP packets, the RTT decreases for the measurements in the second period. The RTT experience a decrease being respectively 0.02 and 0.03 times the RTT of the first period.

Studying the standard deviation in these two data sets we see that the values decrease when the period changes from 09:00 AM - 09:30 AM to 06:00 PM - 06:30 PM. Thus, the values are closer in value when the load is assumed to be lower. An

interesting observation is that the highest and lowest standard deviation is when 2000-byte IP packets are used. However, this also means that the spread in values varies the most when using 2000-byte packets, meaning it is more challenging to predict the results when using this packet size for a pinging session.

Studying the 95%-confidence intervals we see that regardless of the packet size used, the interval decreases in size to the second period. The decrease in the range of the interval when using 28-byte, 1400-byte, and 200-byte IP packets are, respectively, 0.13, 0.06, and 0.72 times from the first period. When comparing the size of the interval with respect to each other, using 2000-byte packets is most affected by the period when the measurements are done.

What do the results mean?

We have seen that the standard deviation do not have a clear tendency in terms of decreasing and increasing in value when the period changes from 09:00 - 09:30 AM to 06:00 PM and 06:30 PM, nor does the size of the 95%-confidence interval. The reason for these observations may be due to the amount of load in the network. We do not know what the load is in the network. However, as stated earlier in this section, we assume that the load is less during the period 06:00 PM - 06:30 PM than in the period 09:00 AM - 09:30 AM. Therefore we assumed that the RTT would be lower during the period 06:00 PM - 06:30 PM. Thus, the results do not correspond to the assumptions made. The reason may be related to the load in the network being approximately the same in both periods, making the assumption wrong. There may be fewer students and employees present, but there may be running other processes causing load in the network. Thus, data packets experience approximately the same amount of delay even though the time of day is different.

When referring to 2000-byte IP, packets, we are talking about two different Ethernet frames having a payload of 1000 bytes each. This is due to the fragmentation that occurs as explained in Section 4.5. How packet fragmentation affects the RTT measurements is variable. When studying the two-hop path in the two periods pinging a server, the 95%-confidence intervals range between ± 0.006 and ± 0.008. Thus when having fragmented packets the range is approximately the same as when using 28-byte and 1400-byte IP packets. However, when the path consists of six hops when pinging the server, the 95%-confidence interval has doubled when using 2000-bytes compared to the other packet sizes studied. This indicates that fragmented Ethernet frames experience a varying delay the more hops the path consists of.

When pinging the router, the data made from 2000-byte IP packets the 95%-confidence interval has a range of approximately 0.026 to 0.200. Using 28- or 1400-byte IP packets, we have seen that the difference in the measurements is more similar, having a range of, respectively, 0.052 to 0.074 and 0.042 to 0.066. This indicates that when dealing with fragmented Ethernet frames, the expected RTT value in a path is more difficult to predict due to the large spread of values in the data set.

One of the main key findings from the results in Section 5.3.1 is the difference in the type of device chosen as the destination. Looking at the accuracy measurement for a data set having the router as the destination compared to the server, we see that the standard deviation and the 95%-confidence intervals for the data set are always smaller than when pinging the server compared to pinging the router. This means that when pinging a router, most cases will give RTT measurements with values further apart from each other, than a server. These observations may be caused by the router having more processes running than the server. Thus, the time it takes to send an ICMP echo response in return varies more compared to the server.

The varying time can occur because the other processes run by the router do not take the same amount of time every time they run, whereas the processes running on the server are more precise in comparison. When having a larger range where the values occur, it will most likely be difficult to compare the data when the network path changes. This is because the increase in RTT may not be outside the expected range of RTT in the original path, causing the change to not be detected.

The reason for assuming that the difference in accuracy is related to the type of destination pinged is based on that the same network devices are used in the other network paths pinged as well as when the server is pinged. Taking the six-hop path between Area 1 and the server located in Area 2 compared to the four-hop path pinging the router, the six-hop path consists of the same network devices as the four-hop path, in addition to two more network devices. Thus, when comparing how the accuracy measurements change in conjunction with the type of destination pinged and the change in the network path, it is reasonable to say that the destination is the reason for the accuracy measurements to get significantly different.

Six-hop path:

PC in Area 1 => switch => area router in Area 1 => router connecting two areas => area router in Area 2 => switch => server in Area 2

Four hop path:

PC at Area 1 => switch => area router in Area 1 => Router connecting two areas => area router in Area 2

Two-hop path:

PC in Area 2 => switch => area router in Area 2

Four hop path:

PC in Area 2 => switch => server in Area 2

5.3.2 What packet size reveals an anomaly in the network?

The following results look at how different packet sizes should be chosen when comparing RTTs to see if a network path has changed. The data collected are illustrated in Figures 5.15 - 5.20. The idea is to study how, respectively, 28-, 1400-, and 2000 bytes reveal a changed network path. Figures 5.15, 5.16 and 5.17 illustrate paths consisting of two and six hops pinging the server. The Figures 5.18, 5.19, and 5.20 are illustrating data sets were the router was pinged. The type of destination that should be chosen when detecting a changed network path is also analyzed. The data is collected during the same time of day to have as similar load in the network as possible. The reason is that we want the environment to be as similar as possible when comparing two data sets.

Following is the four network paths studied. The path consisting of two hops pinging a server: PC in Area 2 => switch => server in Area 2. The Path consisting of six hops pinging the server: PC in Area 1 => switch => area router in Area 1 => router connecting two areas => area router in Area 2 => switch => server in Area 2. The path consisting of two hops pinging the router: PC in Area 2 => switch => area router in Area 2. The path consisting of four hops pinging the router: PC in Area 1 => switch => area router in Area 1 => Router connecting two areas => area router in Area 2.

Pinging the server

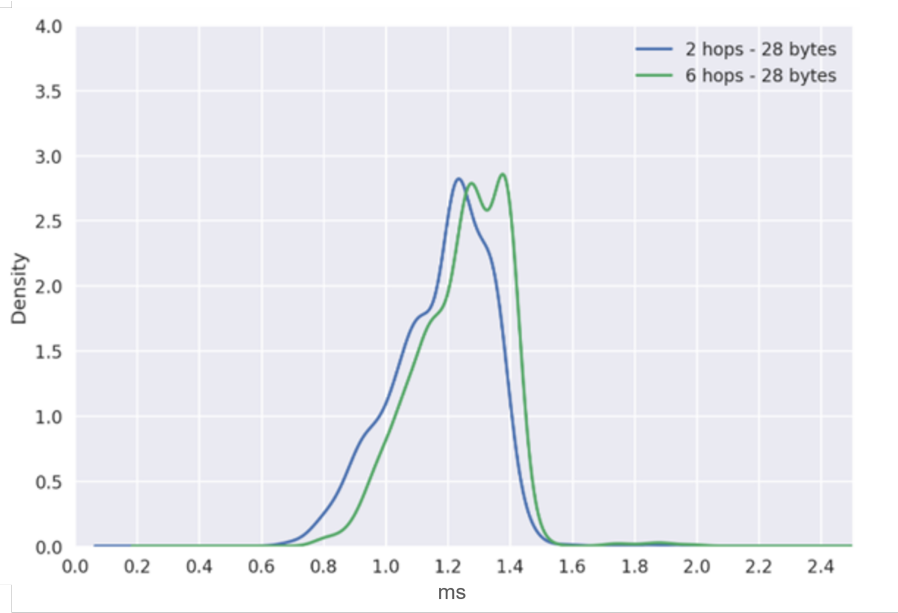


Figure 5.15: Comparing of RTT when using 28 bytes IP packets pinging the server

Number of hops	RTT (ms)	std. deviation	sample mean (ms)	95%-confidence interval size (ms)
2	1.23	0.156	1.179	±0.010
6	1.25 and 1.38	0.146	1.244	±0.009

Table 5.9: Results from figure 5.15

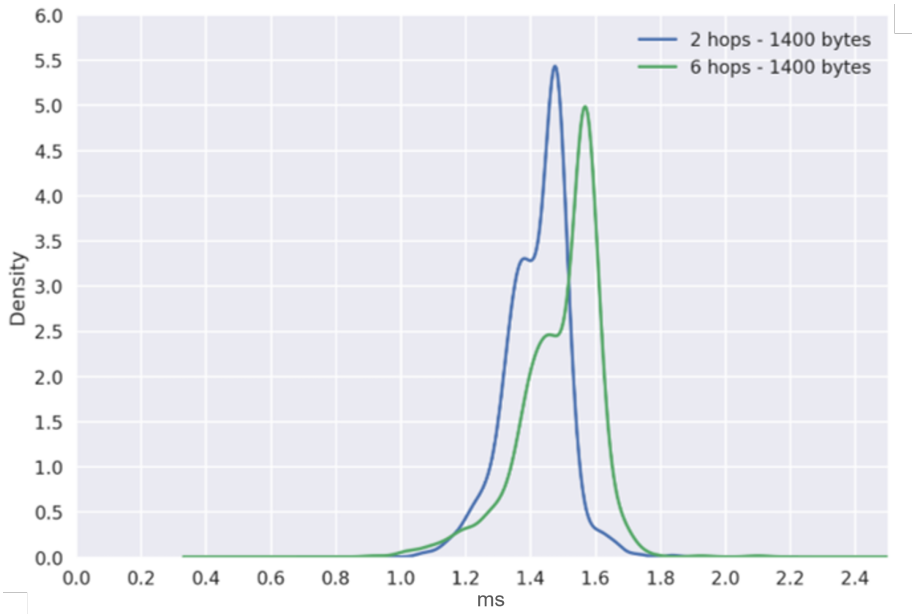


Figure 5.16: Comparing of RTT when using 1400 bytes IP packets pinging the server

Number of hops	RTT (ms)	std. deviation	sample mean	95%-confidence interval size (ms)
2	1.48	0.094	1.421	±0.006
6	1.58	0.123	1.492	±0.008

Table 5.10: Results from figure 5.16

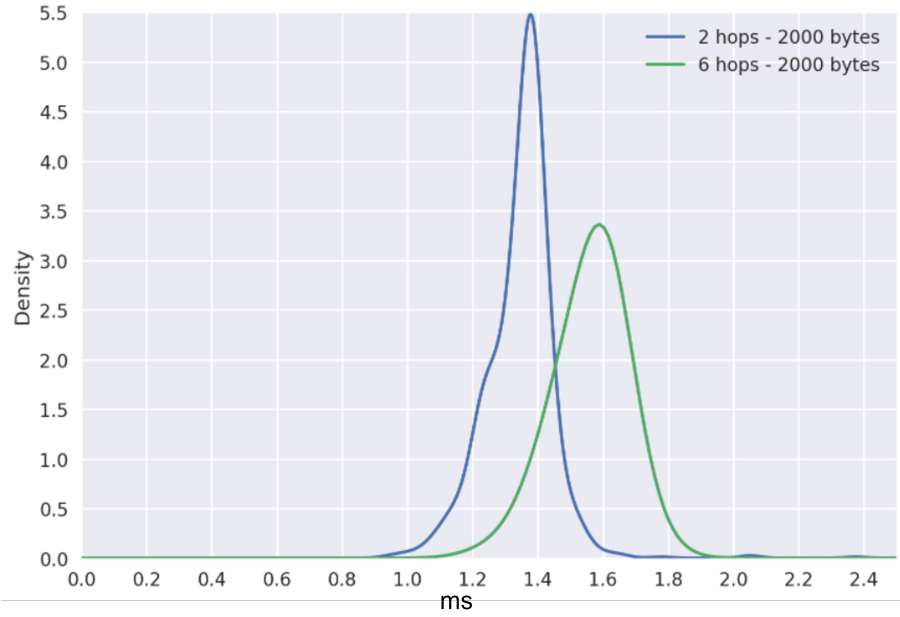


Figure 5.17: Comparing of RTT when using 2000 bytes IP packets pinging the server

Number of hops	RTT (ms)	std. deviation	sample mean	95%-confidence interval size (ms)
2	1.38	0.106	1.346	±0.007
6	1.60	0.242	1.591	±0.015

Table 5.11: Results from figure 5.17

Comparing observations from figures 5.15, 5.16 and 5.17

From the data set studied here, we see that when using 28-byte IP packets, we have two peeks in the data making the RTT measurements 0.02 and 0.11 times larger when pinging the path consisting of two hops compared to the path consisting of two hops. When using 1400 byte IP packets, the RTT is 0.06 times larger in the six-hop path. When the IP packets are 2000-bytes, the RTT increases by 0.14 times when the path increases from two to six hops. Since we want to have the greatest difference in RTTs to get the greatest difference between two paths, IP packets of 2000 bytes would be preferred on the basis of these results.

The six-hop path has an interval 0.1 times smaller than the two-hop path when using 28-byte IP packets. When using 1400- and 2000-byte IP packets, the 95%-confidence interval increases by, respectively, 0.25 and 0.53 times. We want the smallest confidence interval as possible to have the most accurate measurements possible. When using 1400-byte IP packets, the interval has the smallest range in the two situations presented. Comparing how the confidence intervals are with respect to each other, going from 1400-byte IP packets to 28-byte IP packets, the intervals increase by 0.6 and 0.11 times. When going from 1400-byte IP packets to 2000-byte IP packets the intervals increase by 0.14 times and 0.46.

An interesting observation is that the sample mean is significantly lower than the RTT with the highest density when using 1400-byte IP packets. This can indicate that there are a period in the data set were the packets were experiencing a lot less delay, but that the values were not too similar to becoming a clearly visible extreme point. This applies to both data sets. The reasoning can be justified that there is a lower peak on both PDFs, and the area beneath the graph is significantly wider to the left than the right from the maximum point.

Looking at the standard deviations we see that when using 1400-byte IP packets we get the most preferred results since the value is the lowest. We want the RTT values being as close as possible meaning that we know what RTT to expect when doing a pinging session trying to detect a delay anomaly. Based on this information, 1400-bytes would be preferred due to having the shortest 95%-confidence intervals and the lowest standard deviations, regardless of the path studied.

Pinging the router

The same experiment is now presented. However, the router is now the chosen destination. This is because we want to see how the measurements are affected by the type of destination chosen. In Figure 5.18, Figure 5.19 and 5.20 the paths consist of two and four hops.

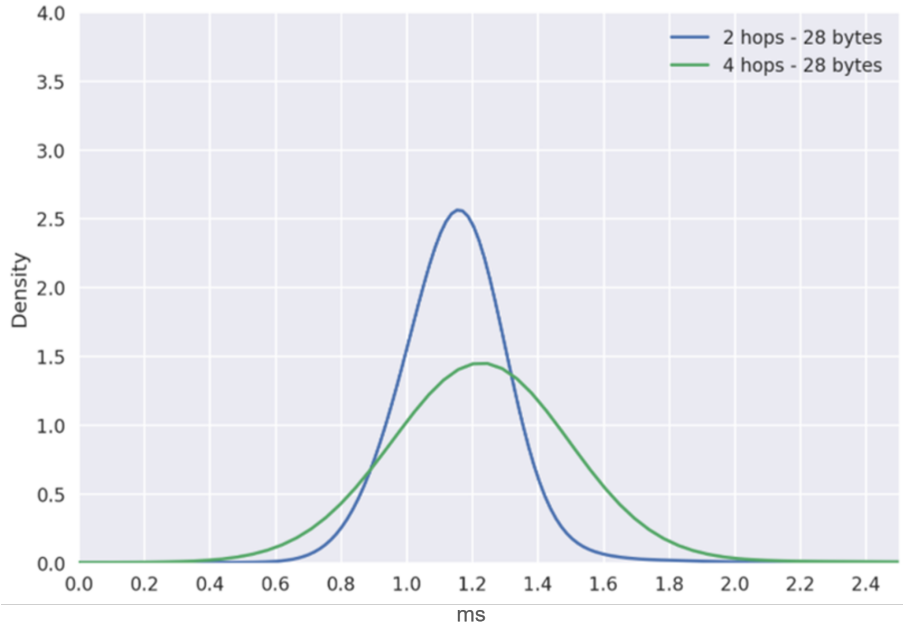


Figure 5.18: Comparing of RTT when using 28 bytes IP packets pinging the router

Number of hops	RTT (ms)	std. deviation	sample mean	95%-confidence interval size (ms)
2	1.18	0.437	1.179	±0.027
4	1.22	0.998	1.304	0.062

Table 5.12: Results from Figure 5.18

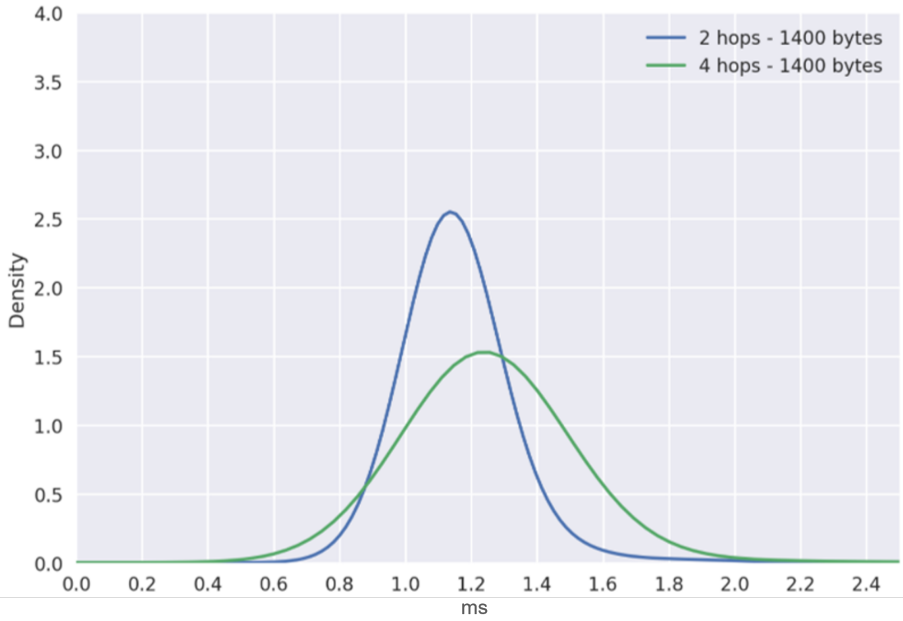


Figure 5.19: Comparing of RTT when using 1400 bytes IP packets pinging the router

Number of hops	RTT (ms)	std. deviation	sample mean	95%-confidence interval size (ms)
2	1.17	0.495	1.197	±0.031
4	1.25	0.933	1.342	±0.056

Table 5.13: Results from Figure 5.19

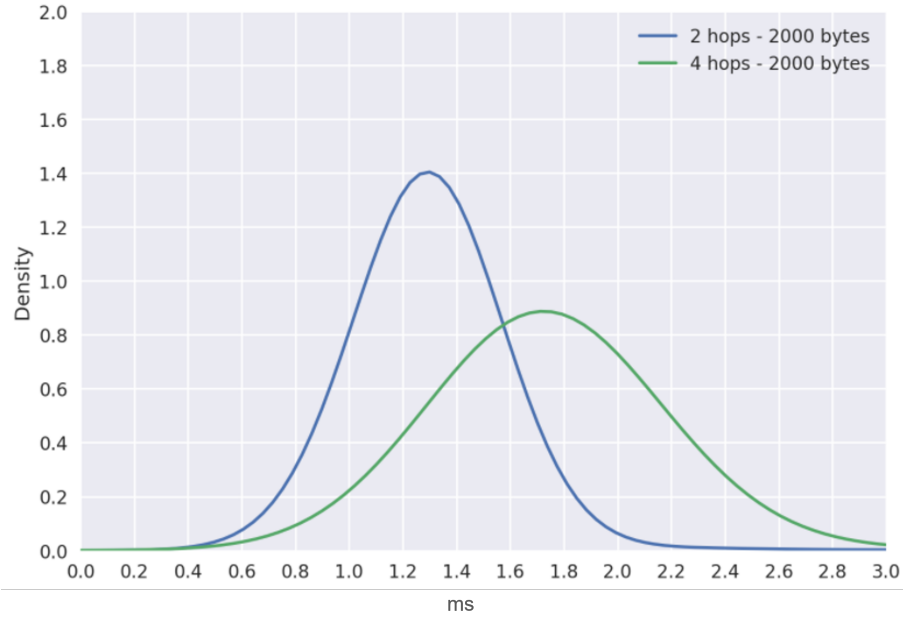


Figure 5.20: Comparing of RTT when using 2000 bytes IP packets pinging the router

Number of hops	RTT (ms)	std. deviation	sample mean	95%-confidence interval size (ms)
2	1.30	0.832	1.362	±0.052
4	1.70	1.392	1.900	±0.090

Table 5.14: Results from Figure 5.20

Comparing observations from Figures 5.18, 5.19 and 5.20

Looking at the RTT with the highest density we see that when using 28-byte IP packets, the RTT increases by 0.03 times as the path increases to four hops. For the 1400-byte IP packets, the four-hop path has a RTT 0.06 time larger than the two-hop path. At last, RTT measurement is 0.31 times larger in the four-hop path than in the two-hop path when using 2000-byte IP packets. As when the server was pinged, the larger the packet size, the greater the difference in RTT when comparing the values in two different network paths.

In the data sets analyzed, the 95%-confidence intervals increase regardless of the packet size used. When using 28-byte IP packets, the four-hop path has an interval

2.3 times larger than the two-hop path. The interval increases by 1.8 times when the IP packets consist of 1400 bytes, and the path goes from the two-hop path to the four-hop path. Using 2000-byte IP packets, the interval increases by 1.7 times. Thus, the increase in the 95%-confidence interval is the smallest when using 2000-byte packets. This means that the spread in the data values is the most similar. However, the intervals when using 2000-byte IP packets are also the widest.

Looking at the standard deviations we see that when using 28-byte and 1400-byte packets the values are approximately the same regardless of the path studied. When using 2000-byte IP packets the standard deviation is almost twice the size in the path consisting of two hops than when using either 28- or 1400-byte packets. Thus, using 2000-byte IP packets is not the optimal choice when we want the interval and standard deviation to be as narrow as possible. Thus, choosing 28-byte or 1400-byte packets is preferred.

What do these results mean?

In Section 5.3.2, we have studied both the type of destination pinged packet and the size of the packet as in Section 5.3.1. However, the focus has been on the conditions that give the most significant difference in RTT when pinging two different paths in the network. As discovered, the server gives more accurate RTT measurements based on the 95%-confidence intervals. Throughout Section 5.3.2, we have studied which packet size should be used to obtain the RTT measurements with the greatest difference. We have also studied how the type of destination pinged affects the difference in RTT.

We have seen that the greatest difference in RTT, looking at the RTT with the highest density, occurs when a router is being pinged using 2000-byte IP packets. As discussed earlier, when using 2000-byte IP packets for pinging, the packets are fragmented into two Ethernet frames consisting of 1000 bytes of payload each. This may be part of the reason why the difference in RTT is most significant when the fragmented Ethernet frames travel through two more hops. Two Ethernet frames will have a longer processing time than one Ethernet frame. For each hop, two Ethernet frames must be processed, making the difference in RTT significantly longer than when only one Ethernet frame is processed. Although the difference in RTT is most significant when using 2000-byte IP packets, the 95%-confidence interval has increased significantly from the two-hop to six-hop path. This may be an indication that for each hop added, the spread in RTT values also increases. This may be because the two fragmented packets are victims of greater delay variations than when only sending one Ethernet frame is sent through the network.

In general looking at the 95%-confidence intervals we see that the intervals are more than doubled in the two-hop path when the router is pinged compared to

pinging the server. When the path consists of six hops, the intervals are more than four times the size when the router is pinged compared to pinging the server. The observations also applies to the values of the standard deviations. The standard deviations has always a larger value when pinging a router compared to a server, indicating that the spread in the RTT values is larger when pinging the router. These observations indicate that when we want to identify if a path has changed, we should ping a server because of the accuracy of the measurements. When the expected value from the RTT values is larger than the time added by a new device in the network, it makes detection of delay anomalies impossible. This is because the "new delay" would blend into the expected RTT values of the original path.

5.3.3 Key findings from the results presented in section 5.3.1 and 5.3.2

There are several key findings to be drawn from this experiment. First, we have seen the differences by pinging the router and the server. When the server is pinged, the spread in the RTT values in every data set is significantly smaller than when pinging the router. This observation is made regardless of the size of the IP packet used for pinging. This is observed both in Section 5.3.1 and Section 5.3.2.

We have studied how the packets are affected by the load in the network. Based on the assumption that the load is less during the period 06:00 PM - 06:30 PM compared to the period 09:00 AM - 09:30 AM, the influence of the load is negligible. This is observed in Section 5.3.1. The reason for this theory is because we do not find any pattern in the measurements between the periods, regardless of the packet size used and type of destination. Theoretically, we know that the load affects the measurements. However, the difference in load may not be significant enough to be detected by the computer because of the accuracy of the measurements. A reason for this finding may be due to the assumption being wrong, as explained in Section 5.3.1.

When choosing the packet size to detect a changed packet size, the 2000-byte IP packets give the largest difference in RTT which is desired. However, they can only be used in a path that does not consist of more than six hops. This is due to the 95%-confidence intervals and standard deviations doubles when the path increases from two to six hops. This can indicate that the 95%-confidence intervals and standard deviations are increasing for each hop added. This key finding is based on the results in Section 5.3.2 when pinging the server. Based on the results from Section 5.3.2 when the server is pinged, using IP packets of 1400 bytes are found to give the data sets with the smallest standard deviation, and the sizes of the 95%-confidence intervals are most stable when comparing the network paths in relation to the number of hops.

5.4 Patterns in the RTT measurements

An interesting observation made is that the data packets often were victims of considerable and varying delays when the router was pinged. Approximately every third time a pinging session was done toward the router, every fourth ping had a significantly higher RTT value than the following three pings. Below is an observed RTT sequence when pinging the router.

RTT sequence:

```
18.944, 1.114, 1.022, 1.088, 9.238, 1.038, 1.195, 1.011, 16.785
```

As seen in the sequence, every third delay is significantly longer than the following three RTT measurements. Also, every third delay also has quite different values, ranging from approximately $9ms$ to approximately $19ms$. Such observations are most likely due to internal processes in the pinged router. It can be caused by the router updating its routing tables or other types of software, which causes such variations in RTT. When other processes are running, the ICMP echo response has to wait longer to be sent from the pinged destination and back to the host. The assumption of the reason being due to other processes running is supported by the varying added delay in every third ping because the time it takes before the router responds with the ICMP echo response is varying as well.

The variations were observed during an entire pinging session, but were sometimes only partially observed through a pinging session. Therefore, close inspection of the RTT measurements was necessary when pinging the router. These incidents caused the data collected not to be usable for analysis because of too high standard variations. Similar observations were never made when the server was pinged. Thus, the assumption that the router is running processes that makes the RTT less accurate still applies.

Chapter 6
Discussion

In this section, we discuss the results presented and try to indicate whether an added hop would be detected in relation to the accuracy of the RTT measurements. This section also addresses the limitations of the proposed work. Additionally, the research questions presented in Chapter 1 are being answered.

6.1 Comparing the RTT in two network paths

In Section 5.3.2, we have seen that the computer can detect a difference between two paths consisting of two and six hops. Thus, when the path increases by four hops, there is a clear difference in the RTT values. Now we want to do a calculation to see whether it is possible to detect a change if only one hop is added or if a link has been extended. Since we have two paths being compared, we can use the difference in RTT as a basis. To do such a calculation, we have to do some simplifications. First, we have to assume that for each hop added, the RTT increases by the same amount.

The results presented in Chapter 5 indicate that the preferred accuracy of the results occurs when using 1400-byte IP packets when pinging the server. Given a 95%-confidence interval has a size of, respectively, $\pm 6\mu s$ and $\pm 8\mu s$ in the two and six hop paths. This means that there is a 95%-chance that the expected RTT through the path is within these intervals when doing many samples. To be able to detect a link extension or an added hop in the network, the RTT values must be increasing by at least $6\mu s$ in the two-hop path and by $8\mu s$ in the six-hop path. Table 6.1 shows that the RTT increases by $0.1ms = 0.100\mu s$ when four hops are added to the path. Assuming that each hop adds the same amount of delay, each hop added would contribute with $0.025ms$ to the RTT.

When can a link extension be detected?

We are now asking when a link extension can be detected based on the assumptions made. We assume that the path is rerouted for this calculation, but still consists of

Packet size (in bytes)	Hops	RTT (ms)
1400	2	1.48
1400	6	1.58

Table 6.1: The RTT having the highest density pinging the server using 1400-byte payload

six hops. The payload is set to 1400-bytes, and we are pinging the server because the data seem to give the most preferred accuracy.

When doing this calculation, we only have to consider the propagation delay.

Propagation delay:

$$\text{Propagation delay} = \frac{distance}{\text{propagation speed}} \tag{6.1}$$

The packet length does not affect the propagation delay. The propagation speed is the speed of light in the medium used. This is in the range of $2 * 10^8 \rightarrow 3 * 10^8 m/s$. Here, we use $2 * 10^8$.

$$\frac{x}{200000000 m/s} > 0.016 ms = 0.000016 s$$

$$x > 200000000 m/s \times 0.000016 s$$

$$x > 3200 m$$

The calculations are only valid when the 95%-confidence interval has the size of $\pm 8 \mu s$. Given the same conditions of the path used in the calculations, a new link of more than 3200 meters can be detected. This i because the added propagation delay is larger than the 95%-confidence interval, thus being a value outside the expected value for the path. We must remember that the increase in delay has to be consistent. This is because the interval only indicates that the expected RTT value is within the interval of a 95% chance. Therefore, there is a chance to get a RTT measurement outside the confidence interval, although a change in the network path is not causing it. However, the increase in delay is not large enough to be detected by the proposed

visualizations used in Section 5.3.1 and Section 5.3.2. The visualization requires a higher resolution to detect changes in the range of μs.

How many paths must be added to the path to be detected using this methodology?

We now want to consider whether an added hop will be detected by the computer considering the same situation as in the previous calculation, namely a 1400-byte payload pinging the server in the two- and six-hop path pinging a server. Still the simplifications and assumptions apply. We must consider the other types of delay for this calculation, including transmission-, processing-, and queuing delays. Propagation and transmission delays are consistent for every added hop, but the queuing and processing delay will affect the RTT measurements the most. For this calculation, we assume that the new hop is connected to the network using links of $200m$. When summarizing all the delays added to the RTT, the value must be greater than the range of 95%-confidence interval to be detected as a delay anomaly. Since each hop adds a delay of $0.025ms$, which is a value larger than the 95%-confidence interval, the RTT in the "new" path can be detected as a delay anomaly.

How much delay does each type of delay contribute with in the network paths?

As stated in Section 2.1, the processing delay is said to be negligible. Now we want to do calculations studying where the delays are occurring based on the assumptions stated at the beginning of Section 6.1. We are still using the same example, the 1400-byte payload, using the two-hop and six-hop paths as a base for the calculations.

Transmission delay:

$$\text{Transmission delay} = \frac{bits}{capacity} \tag{6.2}$$

The Ethernet frame now consists of $1400bytes + 18bytes = 1418bytes$, which is below the maximum packet size. Thus, there is no fragmentation of the Ethernet frame. Using the same formula for transmitting a packet of $1418bytes = 11344bits$

$$\frac{11344bits}{1000000000bps} = 0.000011344s \approx 11.3\mu s$$

The propagation delay when assuming a link is 200 meters, using the speed $2 * 10^8 m/s$:

$$\frac{200m}{200000000m/s} = 0.000001s = 1\mu s$$

The rest of the delay is the processing and queuing delay. When using the calculations above, the processing and queuing delay are in this particular situation:

Six hop path:

$$\text{Total delay} - 2 \times (\text{propagation delay} + \text{transmission delay})$$

$$\text{Transmission delay} = 11.3\mu s \times 6 = 67.8\mu s = 0.0678ms$$

$$\text{Propagation delay} = 1\mu s \times 6 = 6\mu s = 0.006ms$$

$$= 1.580ms - 2 \times (0.0678 + 0.006) = 1.4342ms$$

We have to remember the difference between end-to-end delay and RTT. Thus, the need for multiplying the delay by two is needed to get the total delay in the RTT measurement.

Two hop path:

$$\text{Transmission delay} = 11.3\mu s \times 2 = 22.6\mu s = 0.0226ms$$

$$\text{Propagation delay} = 1\mu s \times 2 = 2\mu s = 0.002ms$$

$$1.480ms - 2 \times (0.002 + 0.0226)ms = 1.4308ms$$

Thus, when the path increases by four hops, the difference in queuing and processing delay increases by

$$1.4342ms - 1.4308ms = 0.0034ms$$

Assuming that every hop adds the same processing and queuing delay, for each hop added to the network, the queuing and processing delay will increase by:

$$\frac{0.0034}{4}ms = 0.00085ms = 0.85\mu s$$

. The total queuing and processing delay, given that each hop contributes with the same amount is therefore

$$0.85\mu s \times 7 = 5.95\mu s \approx 6\mu s$$

When studying the results, we see that the total delay does not add up to the RTT values measured. This is an indication that the computer measuring the RTT

is adding additional delay. This can be caused by the timestamping at the computer. However, since the accuracy measurements are in a range, making it possible to detect changes in the RTT when there is a change in the network. Thus, the delay added by the timestamping at the computer takes approximately the same amount of time, making the added delay acceptable. Furthermore, it also shows that the state of the computer is crucial. The computer must be in the same state when comparing the RTT in a path with an earlier pinging session. This is to avoid significant changes in the RTT, which can be suspected to be caused by a network anomaly, when they are not.

6.2 Research questions

For this section, the research questions presented in Chapter 1 are now being discussed on the basis of the gathered results. First, is a presentation of the accuracy of the RTT measurements followed by the most fundamental research question, being about whether non-designated hardware and software can detect significant changes in the RTT measurements. This question was also presented as a hypothesis as a foundation for this study. Furthermore, a presentation of the conditions that must be present to detect delay anomalies follows.

6.2.1 What is the accuracy of delay measurement when using a computer connected to the network?

Now we look at the accuracy of the RTT measurements. It is essential to note that the accuracy of the delay measurements is not a specified number. This is because we do not know the actual accuracy of the clock inside the computer, which timestamps the data packets. The clock embedded in the computer most likely does not recognize an incoming packet exactly when it is arriving. We do not know how long this time slot is from a packet that arrives at the computer before the computer timestamps it. Therefore, we use standard deviation and 95%-confidence intervals as a measure of accuracy.

In general we have seen that the accuracy of the measurements are significantly different when pinging the server and the router. This is reflected in Table 6.2, 6.3, 6.4 and 6.5 which are summarizing the accuracy measurements calculated from Section 5.3.2. When studying the values keeping in mind we want the data set containing as similar values possible, we see that when pinging the server we get the most preferred accuracy measurements. We also seek as small standard deviation possible to ensure that when measuring RTTs in a path, the expected value is known. Thus, we also want a narrow confidence interval which indicates that the mean of the sample is within the interval with a high certainty. Considering the accuracy measurements in Table 6.2 and Table 6.3, based on the results from Section 5.3.2, the accuracy

of the measurements has the smallest range when using a 1400-byte payload in the Ethernet frames, regardless of the path pinged, given that the destination pinged is the server.

Here is a summary of the accuracy measurements made in Section 5.3.2:

Server:

Packet size (in bytes)	Standard deviation	95%-confidence interval
28	0.156	1.179 ± 0.009
1400	0.094	1.421 ± 0.006
2000	0.106	1.346 ± 0.007

Table 6.2: Network path consisting of two hops, pinging the server

Packet size (in bytes)	Standard deviation	95%-confidence interval
28	0.146	1.244 ± 0.009
1400	0.123	1.492 ± 0.008
2000	0.242	1.591 ± 0.015

Table 6.3: Network path consisting of six hops, pinging a server

Router:

Packet size (in bytes)	Standard deviation	95%-confidence interval
28	0.437	1.179 ± 0.027
1400	0.495	1.197 ± 0.031
2000	0.832	1.362 ± 0.052

Table 6.4: Network path consisting of two hops, pinging the router

Packet size (in bytes)	Standard deviation	95%-confidence interval
28	0.998	1.304 ± 0.062
1400	0.933	1.342 ± 0.059
2000	1.392	1.900 ± 0.087

Table 6.5: Network path consisting of six hops, pinging the router

6.2.2 Is it possible to use non-designated hardware and software to detect delay-anomalies in the network?

In Section 5.3.2, we have studied whether the non-designated hardware and software can differentiate between two paths. However, to see that two different network paths are being analyzed, the accuracy of the measurements needs to be in the range less than the RTT added by one additional hop or by a link extension.

In Section 6.1, the calculations are based on the values collected when pinging the server in a two-hop and a six-hop path using a 1400-byte payload in the Ethernet frames. The results indicate that when a link is extended to 3200 meters, the delay added will increase the RTT to be outside the range of the expected RTT in 95% of the cases. Another calculation also shows that one extra hop added to the network will be possible to detect due to the same reasoning. However, these calculations are only valid in that specific scenario given those accuracy measurements and differences in RTT. Nevertheless, the calculations indicate that non-designated hardware and software have the potential to discover delay anomalies given certain circumstances. Section 6.2.3 gives an explanation of these conditions.

6.2.3 What conditions must be present to make a computer recognize a change in delay in a network path?

The conditions studied in this study are how packet size, time of day, and the type of destination affect the RTT measurements. In Section 5.3.1 we see that pinging sessions using a larger packet size, the longer the RTT. Thus, it is necessary to recognize the packet size used for pinging so that the same packet size is always used. If two pinging sessions are compared, the use of different packet sizes will give a wrong impression of the current situation. We have also seen that when using fragmented Ethernet frames, the 95%-confidence interval seems to increase as the number of hops increases. However, the larger the packet size used, the greater difference is the RTT when comparing the values in two different paths. Thus, the Ethernet frame should be close to the maximum payload allowed in an Ethernet frame. Based on the packet sizes used in the experiment, 1400-byte IP packets are the preferred choice

given that a server is being pinged. We have seen that 2000-byte packets can give better accuracy measurements. However, the variations in the delay measurements are greater for both 28- and 2000-byte IP packets. When using 1400-byte IP packets, the accuracy needed is trustworthy.

The results in Section 5.3.1 and Section 5.3.2 present data where the router and the server have been pinged. The spread in RTT values is always greater when pinging the router than when pinging the server, regardless of the packet size used. The 95%-confidence intervals and standard deviations calculated illustrate this observation. Due to the desire to get the smallest spread in RTT values possible, it is recommended to use a server as the type of destination based on the accuracy measurements.

We have studied how the time of day affects the results. On the basis of the results, measurements should be performed when the load in the network is similar. This is because the load in the network is affecting the results, but we do not know how much. From the results here, based on the assumptions made, the load has a negligible impact. However, the assumption may be wrong, meaning that the load in the network was the same in both periods when the pinging sessions were performed.

The last condition that we want to address is that the computer must perform the least amount of processes simultaneously as the pinging is happening. As stated in Section 5.1, this makes the timestamping of the arrival time of the packets less accurate, which leads to higher standard deviations in the data, and a larger 95%-confidence interval. In addition, having a full RAM slows the processor. Therefore, the computer should be restarted before any RTT measurements are done. This condition is based on observations done during the execution of the experimental work. During the experimental work, the computer sometimes had other processes running, which made the standard deviation to increase from 0.400 to 1.300, because other processes were running, making the timestamping of packets less accurate. This shows that the RTT measurements are fragile and should be exposed to the same conditions in the network when being compared to each other.

6.3 Requirements for detecting a change in a network path or a changed network path

As stated in the problem description, this thesis aims to present a detection method to identify a changed network path or a change in the network path using RTT. A list of requirements is presented containing the main findings of this study. The list presented the requirements necessary to get accurate measurements to be able to detect such changes in a network. The list includes finding from this study and requirements based on well known knowledge.

1. Connect to the network using wired Ethernet.
2. To get the most preferred accuracy for the RTT measurements, the type of destination pinged must be a server.
3. Ensure that the computer is running only necessary processes and that the RAM has enough memory to not slow down the processor. If the RAM is full, restart the computer.
4. Based on the packet sizes used in this study, the Ethernet frames should have a payload of 1400 bytes, making the ICMP packets consist of 1372 bytes.
5. Measurements must be performed at the same time of day to ensure that the data packets experience as similar amount of delay possible.
6. For each pinging session, perform at least 1000 RTT measurements.

6.4 Limitations in the proposed work

A limitation of the proposed work is that due to the fact that the network is a real-life network used by employees and students at NTNU, the amount of data in the network not controllable. Thus, knowing whether there is much or little data flowing through the network is based on assumptions about when few or many students and employees are present on campus. Also, we do not know anything about the different network devices, e.g., how they prioritize ICMP packets, which may affect the results. The devices may sometimes require an update of routing tables or different types of software. Due to not being the ones administering the network, we do not know when this happens and may lead to wrong RTTs.

Due to the network used in the study is operated by NTNU, we were unable to perform RTT measurements from every location in the network. We had to use the available connection points. The network's topology made it challenging to find two network paths that consisted of the same destination, where one of the paths

was one hop longer than the other, e.g., one path consisting of two hops and one consisting of three hops. It would be possible if changing the destination pinged. However, we wanted to study how the destination affected the RTT when the path changed, making that solution not desirable.

Chapter 7
Conclusion and future work

For this section, future work will be presented, followed by a conclusion summarizing the key findings of this study. Future work will focus on using different types of hardware and software in similar studies to find the best type of equipment for such a detection method. Future work also focuses on that the time it takes to analyze the RTT measurements should be shortened by creating fewer manual processes. Finally, machine learning is introduced before a conclusion is drawn.

7.1 Future work

For this study, only one type of non-designated hardware was used. Since different types of computers have different types of abilities, future work should focus on testing different types of non-designated hardware to find an easily accessible computer that gives RTT with higher accuracy than the results presented in this study. Future work should also investigate how the type of pinging tool affects RTT measurement and investigate whether new technology has been developed and can be tested in a similar study. Future work should focus on whether there are other types of pinging tools using different protocols to measure RTT, e.g. TCP that can get more accurate measurements than ICMP-ping. Nevertheless, keeping in mind that the pinging tool should be easily accessible.

In future work, the network paths compared should only have a difference of one hop. The presented methodology aims to detect a changed network path, including detecting whether one hop has been added. In this study, we were only able to study paths with a difference of multiple hops. To be able to study network paths with the desired amount of hops in a real-life network, an operator must be willing to let researchers use its network for exploration while the network runs as expected for its users. The reason is to be able to see how the RTT measurements behave in an actual network, while at the same time being able to study different network paths as desired.

Furthermore, future work should focus on making the method less of a manual process. There are now three steps to create the density plots. The steps consist of doing RTT measurements, then filtering the RTTs from the other data collected using a Python script. Then the density plots are made using a second Python script. In future work, the density plots should be made directly after having gathered the data to quickly get a result of whether there are any delay anomalies that could be caused by a changed network path or a change in the network path. In some cases, delay anomalies in the network may be due to network attacks, and the process needs to be as quick as possible. Thus, to make the process consist of fewer steps, a more effective way of visualizing the data gathered should be designed.

Visualization is excellent when analyzing the RTT using density as a measurement. However, the visualization used in this experiment is not as accurate in distinguishing between small changes in delay. Using algorithms to detect changes in the RTTs that are smaller than a human will be able to detect using visualization is part of future work. A machine learning algorithm can detect patterns humans are most likely to overlook. It is due to either minimal changes in delay or patterns that do not frequently appear in the data sets, making it not seem like a pattern for a human, while the machine learning algorithm can detect the pattern. Therefore, by using machine learning instead of manually analyzing the density of RTTs, smaller changes in the network path and a minor change in the network path that increases the RTTs in the network can be detected.

7.2 Conclusion

Due to the digitization of services, more data travel through networks. The networks have to be secure to ensure that the data is protected. Therefore, a method is necessary to detect if anything has happened in the network. Here, we have presented a methodology that requires only easily accessible hardware and software which can detect delay anomalies that can be caused by a changed network path or a change in the network path.

The experiment presented shows that non-designated hardware and software can be used to detect changes in a network. We have seen that there is a clear difference in the RTT values when a two-hop path is pinged compared to a six-hop path. To be able to detect a changed network path or a change in a network path there are several requirements to be followed. We have seen how the RTT data are affected by different payloads used in the Ethernet frames and the type of destination being pinged. The differences between pinging the router and the server are found to be significant in terms of the accuracy of the RTT measurements. The accuracy of the measurements is a critical factor in whether the hardware and software can detect path changes. The spread of expected RTT values indicates how much the network

anomaly must add to the RTT to be detected. The larger the expected value, the more difficult it is to detect a network anomaly. Thus, using a server as a destination, using IP packets of 1400 bytes, is preferred. When comparing two paths the data compared should have been performed in the same time period. This is due to the experiment not being able to conclude on how much the load in the network were affecting the results. Data packets in two pinging sessions should have as similar prerequisites possible to make the comparisons valid and trustworthy.

References

[16] «Ieee std 802.3-2015 (revision of ieee std 802.3-2012) : Ieee standard for ethernet.», 2016.

[22] «Iso/iec 27002:2022(en), information security, cybersecurity and privacy protection — information security controls», Feb. 2022. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:27002:ed-3:v2:en.

[ABF+00] A. Adams, T. Bu, *et al.*, «The use of end-to-end multicast measurements for characterizing internal network behavior», *IEEE Communications Magazine*, vol. 38, pp. 152–158, 5 May 2000.

[AGR03] K. G. Anagnostakis, M. Greenwald, and R. S. Ryger, «Cing: Measuring network-internal delays using only existing infrastructure», *Proceedings - IEEE INFOCOM*, vol. 3, pp. 2112–2121, 2003.

[AH09] B. Aziz and G. Hamilton, «Detecting man-in-the-middle attacks by precise timing», 2009. [Online]. Available: http://www.schneier.com/blog/archives/.

[Ame22] Amedia, «Kort om dataangrepet mot amedia», Feb. 2022. [Online]. Available: https://www.amedia.no/kort-om-dataangrepet-mot-amedia.

[BAČ+18] D. Brahneborg, W. Afzal, *et al.*, «Round-trip time anomaly detection», *ICPE 2018 - Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering*, vol. 2018-March, pp. 107–114, Mar. 2018.

[BGS07] Ø. Borgan, I. K. Glad, and A. R. Swensen, «Konfidensintervall», Aug. 2007. [Online]. Available: https://www.uio.no/studier/emner/matnat/math/STK1110/h07/annet/konfidensint.pdf.

[Bjø22] J. Bjørnstad, «Konfidensintervall», Apr. 2022. [Online]. Available: https://snl.no/konfidensintervall.

[cFo22] cFos, «Hrping - high-precision ping utility», 2022. [Online]. Available: https://www.cfos.de/en/ping/ping.htm.

[Cis06] Cisco, *Understanding the ping and traceroute commands - cisco*, Nov. 2006. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html.

[DF07] B. Donnet and T. Friedman, «Internet topology discovery: A survey», *IEEE Communications Surveys & Tutorials*, vol. 9, 4 2007. [Online]. Available: https://inl.info.ucl.ac.be/system/files/DONNET+LAYOUT.pdf.

[FJ18] J. Farmer and D. Jacobs, «High throughput nonparametric probability density estimation», 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0196937.

[Gez19] A. Gezer, «Large-scale round-trip delay time analysis of ipv4 hosts around the globe», *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, pp. 1998–2009, 3 2019.

[Har00] L. Harkins, «Icmp», vol. 9, Jul. 2000. [Online]. Available: https://www.proquest.com/docview/191116132/fulltextPDF/19A4CBFE8784A2CPQ/1?accountid=12870.

[HHZ+21] B. Hou, C. Hou, *et al.*, «Detection and characterization of network anomalies in large-scale rtt time series», *IEEE Transactions on Network and Service Management*, vol. 18, pp. 793–806, 1 Mar. 2021.

[Insa] N. Insitutt for matematiske fag, *Standardavvik*. [Online]. Available: https://tma4245.math.ntnu.no/forventing-og-varians/standardavvik/.

[Insb] ——, *Utlede konfidensintervall for my i normalfordeling, kjent varians*. [Online]. Available: https://tma4245.math.ntnu.no/konfidens-og-prediksjonsintervall/konfidensintervall/regneprosedyre-for-%5C%C3%5C%A5-utlede-konfidensintervall/utlede-konfidensintervall-mu-i-normalfordeling-kjent-varians/.

[JKA+16] H.-D. J. Jeong, H. Kim, *et al.*, «Analysis and detection of anomalous network traffic», in *2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2016, pp. 403–408.

[KR17] J. F. Kurose and K. W. Ross, «Computer networking a top-down approach seventh edition», 2017. [Online]. Available: www.pearsoned.com/permissions/..

[Kum16] G. Kumar, «Denial of service attacks-an updated perspective», *SYSTEMS SCIENCE & CONTROL ENGINEERING: AN OPEN ACCESS JOURNAL*, vol. 4, pp. 285–294, 2016. [Online]. Available: https://www.tandfonline.com/action/journalInformation?journalCode=tssc20http://dx.doi.org/10.1080/21642583.2016.1241193.

[LLCZ11] Y. Li, D. Li, *et al.*, «Research based on osi model», in *2011 IEEE 3rd International Conference on Communication Software and Networks*, May 2011, pp. 554–557.

[MAB18] D. Mirkovic, G. Armitage, and P. Branch, «A survey of round trip time prediction systems», *IEEE Communications Surveys and Tutorials*, vol. 20, pp. 1758–1776, 3 Jul. 2018.

[mat22] matplotlib.org, «Matplotlib: Visualization with python», 2022. [Online]. Available: https://matplotlib.org/.

[MC08] X. Ma and X. Chen, «Performance analysis of ieee 802.11 broadcast scheme in ad hoc wireless lans; performance analysis of ieee 802.11 broadcast scheme in ad hoc wireless lans», *IEEE Transactions on Vehicular Technology*, vol. 57, p. 3757, 6 2008.

[MC18] L. /. Man and S. Committee, «Ieee standard for ethernet», *IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015)*, pp. 1–5600, 2018. [Online]. Available: http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html..

[Mer22] Merriam-Webster, «Definition of freeware», 2022. [Online]. Available: https://www.merriam-webster.com/dictionary/freeware.

[MKH11] M. Merabti, M. Kennedy, and W. Hurst, «Critical infrastructure protection: A 21st century challenge», *2011 International Conference on Communications and Information Technology, ICCIT 2011*, pp. 1–6, 2011.

[MZK11] C. Malone, M. Zahran, and R. Karri, «Are hardware performance counters a cost effective way for integrity checking of programs», Oct. 2011.

[NAGY03] M. Naghedolfeizi, S. Arora, *et al.*, «Effect of ram amount on the thermal behavior of cpu operating under a heavy computational load», 2003.

[NGSB20] I. Nedyalkov, G. P. Georgiev, *et al.*, «Ways to measure the delay in ip networks», *28th National Conference with International Participation, TELECOM 2020 - Proceedings*, pp. 33–36, Oct. 2020.

[Num22] NumPy, «Numpy», 2022. [Online]. Available: https://numpy.org/.

[pan22] pandas.pydata.org, «Pandas», 2022. [Online]. Available: https://pandas.pydata.org/.

[PG] J. Paparrizos and L. Gravano, «K-shape: Efficient and accurate clustering of time series», [Online]. Available: http://dx.doi.org/10.1145/2723372.2737793..

[PJM+] J. Postel, D. Johnson, *et al.*, *Internet control message protocol (icmp) parameters*. [Online]. Available: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml.

[Pos81] J. Postel, «Internet control message protocol», Sep. 1981. [Online]. Available: https://www.rfc-editor.org/info/rfc0792.

[Pre02] O. U. Press, «Statistical tables», 2002, C. Dougherty 2001, 2002, These tables have been computed to accompany the text C. Dougherty Introduction to Econometrics (second edition 2002, Oxford University Press, Oxford), They may be reproduced freely provided that this attribution is retained.

[PRO81] D. I. PROGRAM, *Rfc 791 - internet protocol*, Sep. 1981. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc791.

[Pyt22] Python software foundation, 2022. [Online]. Available: https://www.python.org/.

[RLS21] K. Roenneberg, M. Lysberg, and K. M. Soerenes, «Stortingstopp fikk frastjålet minst 4000 e-poster. mistanken rettes mot kinesiske hackere.», Sep. 2021. [Online]. Available: https://www.aftenposten.no/verden/i/nWG8RB/stortingstopp-fikk-frastjaalet-minst-4000-e-poster-mistanken-rettes-mot-kinesiske-hackere.

[RWW04] R. Ramaswamy, N. Weng, and T. Wolf, «Characterizing network processing delay», *GLOBECOM - IEEE Global Telecommunications Conference*, vol. 3, pp. 1629–1634, 2004.

[Sil86] B. W. Silverman, «Density estimation for statistics and data analysis», *Monographs on Statistics and Applied Probability*, 1986.

[SJ95] D. L. Stone and K. Jeffay, «An empirical study of delay jitter management policies», 1995, pp. 267–279.

[TRKF04] Z. Trabelsi, H. Rahmani, *et al.*, «Malicious sniffing systems detection platform», *Proceedings - International Symposium on Applications and the Internet*, pp. 201–207, 2004.

[TRLC15] F. Tarissan, É. Rotenberg, *et al.*, «Udp ping: A dedicated tool for improving measurements of the internet topology», *Proceedings - IEEE Computer Society's Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, MASCOTS*, vol. 2015-February, pp. 506–509, February Feb. 2015.

[VR02] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. Springer, 2002.

[WDJG07] L. Wenwei, Z. Dafang, *et al.*, «On evaluating the differences of tcp and icmp in network measurement», *Computer Communications*, vol. 30, pp. 428–439, 2 Jan. 2007.

[WS] B. Williams and S. Sawyer, *Using information technology 9e complete edition 9th edition*. [Online]. Available: https://www.amazon.com/Using-Information-Technology-9e-Complete/dp/0073516775.

[Węg18] S. Węglarczyk, «Kernel density estimation and its application», 2018. [Online]. Available: https://doi.org/10.1051/itmconf/20182300037.

[ZW09] J. Zhang and X. Wang, «Robust normal reference bandwidth for kernel density estimation», *Statistica Neerlandica*, vol. 63, pp. 13–23, 1 2009.

[Aar14] H. Aarnes, «Sannsynlighetsfordelinger», 2014. [Online]. Available: https://www.mn.uio.no/ibv/tjenester/kunnskap/plantefys/tall/statfordeling.pdf.

Appendix

Appendix A

A.1 Code

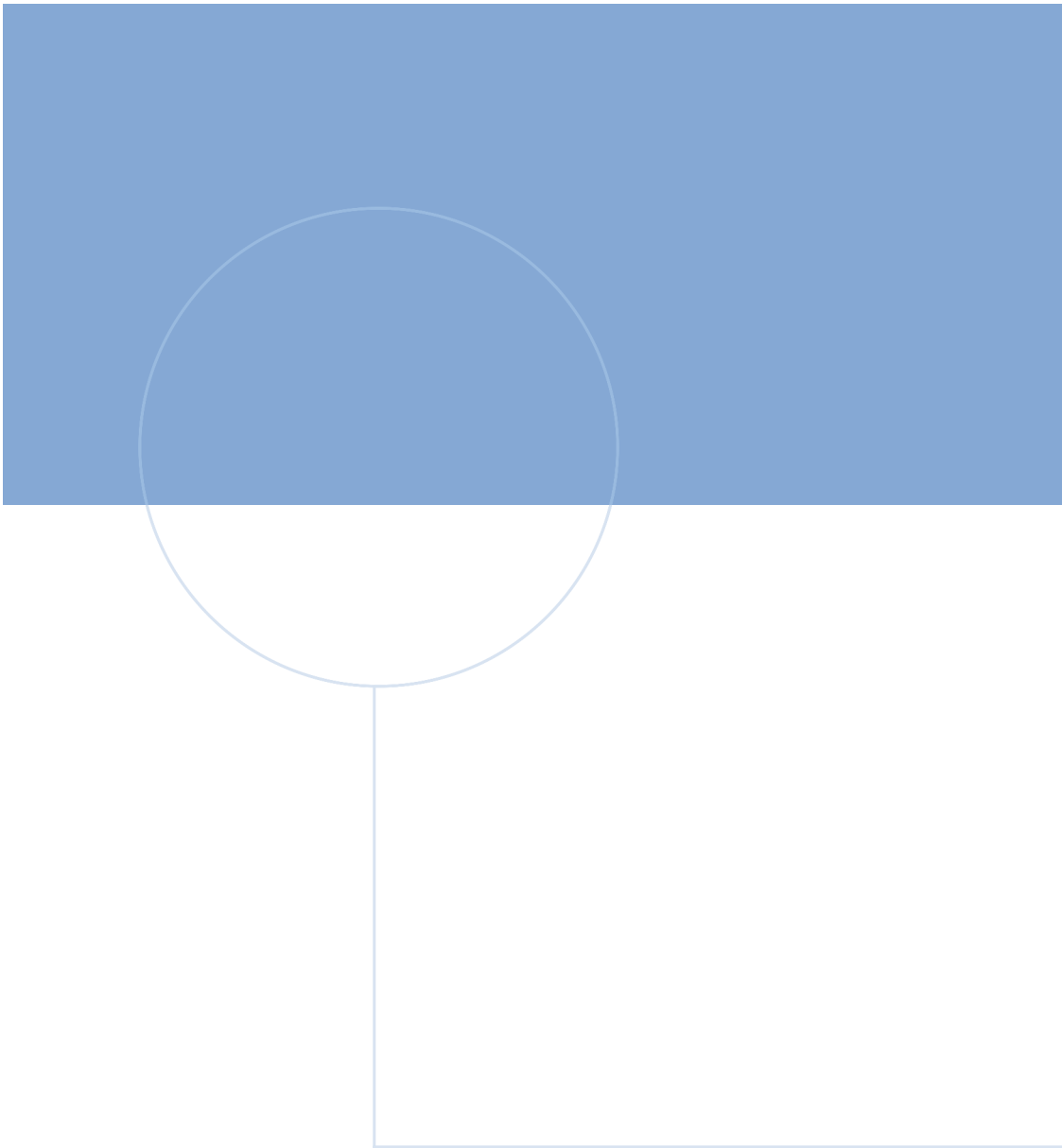
A.1.1 Code used to create a csv-file containing only RTT from ping results

```
import re
pingfile = open("path")
string = pingfile.read()
values = re.findall(r'time=(\d+\.\d+)', string)
with open('path', 'w') as csvfile:
csvfile.write("\n".join(values))
```

A.1.2 Code used to create plots, find standard deviation and sample mean

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
import numpy as np


#Reads the csv-files, and naming the column 'RTT'
pings = pd.read_csv('parsed_2504_hovedbygget_2.csv', names=['RTT'])
pings2 = pd.read_csv('parsed_2504_hovedbygget_4.csv', names=['RTT'])
pings3 = pd.read_csv('parsed_2504_hovedbygget_5.csv', names=['RTT'])

#Using Pandas to create Dataframes
df1 = pd.DataFrame(pings)
df1.columns=['4 hops - 28 bytes']

df2 = pd.DataFrame(pings2)
df2.columns=['4 hops - 1400 bytes']

df3 = pd.DataFrame(pings3)
df3.columns=['4 hops - 2000 bytes']


#Using NumPy to find the standard deviation and sample mean

sample_std1 = np.std(df1)
sample_mean1 = np.mean(df1)

sample_std2 = np.std(df2)
sample_mean2 = np.mean(df2)

sample_std3= np.std(df3)
sample_mean3 = np.mean(df3)


#Printing information (Example)
print('Connection: Ethernet, #pings: 1000')
print('Source Hovedbygget \n Destination: el-dsw.nettel.ntnu.no')
\n Destination red: el-dsw.nettel.ntnu.no')
print('Time 09:00 - 09:30')
```

```
print('Blue: Standard Deviation=%.3f' % (sample_std1))
print('Green: Standard Deviation=%.3f' % (sample_std2))
print('Red: Standard Deviation=%.3f' % (sample_std3))

print('Blue: sample mean=%.3f' % (sample_mean1))
print('Green: sample mean=%.3f' % (sample_mean2))
print('Red: sample mean=%.3f' % (sample_mean3))



#Makes you able to increase and decrease the image of the plot
plt.rcParams['figure.dpi'] = 200

#Using 'style' from matplotlib to get a grid
style.use('seaborn')

#Creating the x-axis and y-axis
ax1 = plt.gca()
ax1.set_xlim([0, 3])
ax1.set_ylim([0, 4])
plt.locator_params(axis="x", nbins=12)
plt.locator_params(axis='y', nbins=10)



#The graphs are being plotted using Pandas, and the bandwidth is set
choosing the 'smoothness' of the plots.
df1.plot(kind='density',ax = ax1, bw_method=0.30)
df2.plot(kind='density', ax = ax1, secondary_y=False, bw_method=0.30)
df3.plot(kind='density', ax = ax1, secondary_y=False, bw_method =0.30)

plt.show()
```



NTNU

Norwegian University of
Science and Technology