An Analysis of Internet Traffic Flow in SANReN using Active and Passive Measurements

by

Luqmaan Salie

Minor Dissertation presented in partial fulfilment
of the requirements for the degree of Master of Science

in the
Department of Computer Science
University of Cape Town

Supervised by Dr Josiah Chavula

Cape Town
March 2021

Plagiarism Declaration

I know the meaning of plagiarism and declare that all of the work in the dissertation, save for that which is properly acknowledged, is my own.

Abstract

National research and education networks (NRENs) in developing regions such as Africa experience various performance issues due to inadequate infrastructure and resources. The South African National Research Network (SANReN) connects universities, research institutions, and oversees science projects such as the Square Kilometre Array. In this study, we conduct active and passive measurements to assess the performance of SANReN and to identify problem areas in the network. Active measurements were done to determine network performance when accessing SANReN internally (using PerfSONAR) and externally (using Speedchecker). We found that SANReN needs to be reinforced in and around Port Elizabeth, Cape Town, and Durban. Universities in these cities had the highest delays and page load times. We found that the network traffic flowing from PE uses circuitous routes to flow to universities in Johannesburg and Pretoria, causing high delays (medians of 25.26 ms to WITS, 25.47 ms to UJ, and 25.95 ms to UNISA) and high page load times (medians of 237.07 ms to WITS, 272.09 ms to UNISA, 280.47 ms to UJ transferring 31594 bytes of data). Using Cape Town as the traffic source resulted in a low median throughput of 5.47 Gbps for internal active measurements. Throughput from Durban to Cape Town was low as well (4.91 Gbps), causing high page load times between these two cities (medians of 350.32 and 305.22 ms from Durban to UCT and UWC respectively). SANReN's passive measurements results show us that there is a ratio of 11.16:1 for download speed to upload speed. We also observe a ratio of 2.29:1 for outbound flows (uploads) to inbound flows. Thus, majority of traffic flows experience low throughput amounts. Based on the test results, we design an SDN model and compare its performance to SANReN. The SDN model's results show that it would increase throughput while decreasing delays and page load times.

Acknowledgments

Firstly, I would like to thank my wife, Mariam Maneveld Salie, for all her support, for her constant encouragement, for her patience, and for all the sacrifices she has made. I would also like to thank my parents for everything they have done for me and everything they have given me. I express thanks to my employers for sponsoring me and allowing me to study while working. I would like to acknowledge and thank Speedchecker for giving me access to their API to conduct the external measurements. I would also like to thank SANReN for providing their network traffic data for me to analyse. Finally, I want to thank my supervisor, Dr Josiah Chavula, for his motivation, advice, and all the help he has given me.

Contents

List of Figures

List of Tables

List of Abbreviations

AARNET	Australian Academic and Research Network
AM	Anchoring Measurement
ARP	Address Resolution Protocol
AS	Autonomous System
ASN	Autonomous System Number
BWCTL	Bandwidth Test Controller
CDN	Content Delivery Network
CSIRT	Computer Security Incident Response Team
DNS	Domain Name System
DST	Department of Science and Technology
DUT	Durban University of Technology
ESP	Encapsulating Security Payload
GRP	Generic Routing Encapsulation
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IQR	Interquartile range

ISP	Internet Service Provider
IXP	Internet Exchange Point
NDT	Network Diagnostic Tool
NMU	Nelson Mandela University
NREN	National Research and Education Network
OWAMP	One-Way Ping
PERT	Performance Enhancement Response Team
PLT	Page Load Time
POP	Point of Presence
QoE	Quality of Experience
QoS	Quality of Service
RTT	Round-trip Time
SAFIRE	South African Identity Federation
SANReN	South African National Research Network
SDN	Software Defined Networking
SKA	Square Kilometre Array
SR	Segment Routing
SRN	Software Resolved Network
STP	Spanning Tree Protocol

TCP	Transmission Control Protocol
TENET	Tertiary Education and Research Network of South Africa
TLD	Top-level Domain
UCT	University of Cape Town
UDM	User-Defined Measurement
UDP	User Datagram Protocol
UJ	University of Johannesburg
UNISA	University of South Africa
UWC	University of the Western Cape
VPN	Virtual Private Network
Wits	University of the Witwatersrand, Johannesburg

Chapter 1

Introduction

The year 2020 has brought a pandemic which has forced scholars and the working population to carry out their work from home. The situation has exposed and magnified broadband access gaps that countries have in terms of network coverage, performance, and affordability. Some South African universities, such as University of Cape Town (UCT), have provided students with internet data bundles to aid online teaching and learning during the lockdown. In addition, universities have had to negotiate zero-rated access to several educational web servers. These zero-rated web servers were carefully selected by universities due to their importance for academic purposes.

With the pandemic changing the network traffic flow – increasing the amount of work being done from outside of the university and its internal network – traffic flow into the network would increase [1], which adds extra stress on the complex job of network upkeep and maintenance. Scalability, reliability, performance, and quality of experience (QoE) are among the most common issues faced by network administrators [2], [3]. These issues are amplified in the context of developing regions such as Africa.

Active network measurements or active network monitoring is a method of testing network performance which involves manually injecting packets into the network and monitoring their path and the network's performance. It produces extra traffic in the network as opposed to naturally occurring traffic created by end-users. Network administrators can control the volume of traffic used when conducting each test, to ensure the tests produce meaningful results [4]. Passive network measurements involves recording network traffic over prolonged periods of time. The network could use a specific device to record traffic or use routers that can record traffic. The recording device is polled for its recorded information and the data is collected by the network administrator [4]. Active and passive measurements have been used to conduct QoE studies and network performance studies such as: [5], [6], [7].

Software Defined Networking (SDN) is a networking paradigm that separates the forwarding plane and the control plane by having a central controller which calculates all routes and allows network switches to solely carry out forwarding of packets [2]. This means that changes to the network configuration can be made in a single location with software, then distributed to the routers in the network. This method is more convenient and less error prone than configuring each network switch individually [3]. The controller can also be programmed to automatically change routes when certain paths in the network undergo unexpected delays, congestion, or failures [3]. We use SDN to design a model of SANReN and compare the performance of the SDN model to the results from the measurements that were carried out.

1.1 Problem Statement

South Africa specifically has issues in terms of performance with both fixed and mobile internet connectivity. Chetty et al. [8] has shown South African broadband users regularly do not achieve the bandwidth speeds that are expected on their respective internet connections. There are often high delays between different Internet Service Providers (ISPs), which largely affect the performance of the network and QoE for users. Another factor hindering network performance is the lack of reliability and robustness of the network [9]. Some of South Africa's networks are inflexible and do not have high inter-connectivity [8], and such networks suffer when links or nodes go down. These problems stem from multiple factors, a key factor being ISPs having low inter-connectivity between each other, resulting in high delays and long round-trip times (RTT) [8]. In terms of National Research and Education Networks (NRENs), traffic flow between African universities has been shown to use primarily (over 75%) circuitous routes to reach destinations, causing high delays of up to twice the time of normal traffic transported in and around Africa [10]. In addition to the issues that exist in South African networks, the pandemic added extra stress on the networks, resulting in an increase in RTT as well as a user-perceived decrease in video streaming quality experienced by Facebook users [11]. We explore the issues faced by the South African National Research Network (SANReN) by using active and passive measurements.

1.2 Research Questions

In this study, we focus on the network architecture of SANReN, its performance and its trends. The main research question we try to answer is: What, if any, are the performance issues in SANReN? What are the usage trends of SANReN? Would an SDN architecture improve network performance? More specifically, we explore the following questions:

- Where are the performance gaps in the topology of the network, what is causing these issues and how can we improve the network resource utilisation?
- Which countries, ISPs, and protocols produce the highest volume of traffic and during which periods does the network experience the most traffic flow?
- To what extent, if any, would an SDN framework improve SANReN's network performance in terms of delays, page load times, and throughput?

1.3 Contribution

The main contribution of the study is to find the performance issues within SANReN and find the reasons for the issues that exist within SANReN. We also contribute an analysis of the traffic flow and its trends within SANReN during the Coronavirus pandemic of 2020. Once all the experiments and monitoring have been completed, we analyse the trends that emerge from both sets of results. From the resulting plots, we find patterns and weaknesses in SANReN. Finally, using the results, we propose an SDN model of SANReN and compare the performance found from the active measurements to our SDN model of SANReN. To the best of our knowledge, this is the first study conducted using active and passive measurements focusing on SANReN and comparing the performance to an SDN model of the network.

1.4 Outline

This study consists of three parts – starting with active network measurements, followed by passive network measurements, and finally measuring network performance with an SDN model based on our experiment results. Firstly, network performance tests are carried out to analyse the quality of service (QoS) when accessing the zero-rated web servers hosted by SANReN. More specifically, the active measurements focus on performance disparities when accessing these educational web servers from different locations and networks in South Africa. The active measurements also focus on the performance of accessing these educational web servers from within the South African National Research Network (SANReN) compared to from outside the network. One of the key metrics used for the comparison is page load time (PLT) – which is the average time taken from the time the user enters the URL in the browser, until the page is completely loaded. The other metrics used are end-to-end packet delay, throughput, traceroute, and packet loss. The location of web servers, inter-connection between network operators and SANReN, and consequently, the network paths followed by packets from source to destination are among the factors that are investigated.

We conduct passive network measurements to find the trends that are present in the network. These network trends are extracted from data of naturally occurring traffic inside SANReN during the 2020 South African lockdown. The usage trends of the network show how the network was used during the lockdown period in terms of where traffic is coming from, where it is flowing to, and the volume of data that was transferred between certain countries, ISPs and periods. In addition, to find the trends in performance, we measured the speed of data being transferred through the network. This gives us an understanding of the flow of traffic within SANReN and we are able to extrapolate the throughput in the network and the main sources of data transfer.

Finally, we propose an SDN model for SANReN by using the existing SANReN topology and implementing the delays found from the active performance experiments. The SDN model makes use of the Spanning Tree Protocol (STP) where links are given a cost based on the link capacities. This model serves as a basis for using SDN as a solution to SANReN's network issues. We conduct the active performance measurements on our proposed model and compare the results to the active and passive measurements conducted on the existing SANReN architecture.

The paper is organised as follows: the next chapter (2) outlines previous works related to network performance in South Africa and NRENs. Chapter 3 describes the methodology used to select the test sources and destinations, carry out the tests, gather results, mine through the SANReN data, and create our SDN model. In chapter 4, we present the results of the active and passive network monitoring that was carried out, followed by a comparison of the performance of our SDN model. Chapter 5 conducts a discussion of the results, analysing the causes and effects of the network performance. Finally, chapter 6 concludes the paper and outlines future work.

Chapter 2

Background and Related Work

Our research focuses on the network usage and patterns of SANReN during the coronavirus pandemic as well as the performance of SANReN with its current architecture. We use active and passive measurements to analyse the performance in various parts of the network to identify and explain any issues that are present. In addition, we explore an SDN solution as a possible improvement to the current network architecture used by SANReN. In this section, we discuss previous works that have used similar methodologies and technologies to analyse network usage and changes. We discuss previous works that have used active and passive monitoring methods, and finally, we explore implementations and uses of SDN.

2.1 Background

2.1.1 Applications of Active Measurements

Active network measurements are used to monitor networks and assess their performance. It consists of manually injecting packets into the network to conduct performance tests such as measuring the time it takes to traverse a network path, figuring out the network capacity, or finding response times within the network. Unlike passive measurements, active measurements do not require large amounts of storage space because only the traffic generated by the test is relevant and needs to be recorded. In addition, there are no privacy concerns with the data recorded because it is not actual user data [4]. Extra network traffic is created when conducting active measurements which could disrupt the natural flow of traffic in the network. To avoid the potential of skewing the results, researchers have combined active measurements with passive measurements, applying the method that would produce the most accurate results at various times when measurements need to be executed [12], [13], [14]. Active network measurements are applied to find the network performance, focusing on delay, packet loss and bandwidth capacity [15], [16], [17], [18].

Active Measurements applied to SDN

de Vergara et al. conduct a clearly detailed active measurement study [15] to dynamically find out if paths meet the requirements of the data that needs to be transferred in an SDN. The architecture for conducting the experiments consists of two parts – the transmitting side and a receiving side. Packets are timestamped at both points of their transfer, on the receiving side for calculating throughput and on the transmitting side to calculate round-trip time. At the request of a measurement, the transmitting side generates a packet, setting parameters such as VLAN, source and destination IP addresses and ports, and packet size. For each test, the number of packets sent and the inter-packet gap can be set as well. Moreover, the architecture allows for *fine* or *coarse* tests – for *fine* latency measurements, the minimum packet size will be used, whereas for *fine* throughput measurements, the maximum packet size is used. Once the packets have been generated, they are sent to the destination where the receiving side will filter by type such as Address Resolution Protocol (ARP) and User Datagram Protocol (UDP), the packets are verified and the information is extracted, compiled and sent to the controller of the SDN.

In this study, we see how active measurements can be used to find the performance of a network and whether a certain path has the capacity to transfer packets based on their requirements. The clearly defined parameters and test packet design is an example of how tests can be created for specific purposes. We conduct tests to determine the performance of SANReN in a similar manner, primarily focusing on latency.

Large-Scale Active Measurements

The study [16] conducted by van Rijswijk-Deij et al. attempts a large-scale active measurement experiment, targeting the largest top-level domain (TLD) and aiming to carry out the experiment without negatively affecting the natural traffic in the network. The goals of the study are: to measure all domains in a TLD with its main target being *.com*, measure relevant resource records which cover the most common DNS uses, measure each domain once a day for a year, be able to analyse the data collected efficiently, and be able to scale the measurement design to TLDs of different sizes. The challenges expected in performing the study are: the query volume (measuring every domain in *.com* is expected to require more than 1.85 billion queries per day), query pacing (the volume of queries might cause strain on the DNS), storage of data, robustness of the measurement system, and ease of operation.

The paper motivates the design choices made, explaining the choice of using off-the-shelf DNS software instead of implementing a DNS resolution from the ground up. The choice was made to use existing software because it mitigates the risk of bugs in a newly developed application by using existing and trusted software, it also avoids the large amount of work that developing a new application would necessitate. Although the novel solution would offer higher speeds due to it being devoid of excess software or features, this was not one of the goals or challenges that needed to be overcome. Thus, the solution to use existing software was chosen due to the robustness it offers, which is important when collecting data over long periods. The second design decision was to choose the best design for scalability of the system. From a previous study conducted by the researchers, a single system would be insufficient in supporting the amount of measurements that were going to be carried out. This is due to the high CPU utilisation that is caused by carrying out the measurements. Thus, a distributed virtual image approach was chosen, making sure the implementation can be easily converted to a cloud solution if the system needs to be scaled up. Our study carries out the active measurements using existing tools as well, as we were able to leverage the advantages of both Speedchecker [1] and perfSONAR [19] to target specific source and destination combinations. Although the distributed approach is more scalable and more flexible, we did not choose this approach due to our measurements having a smaller target network. We were able to complete our experiments using a single system.

In terms of making sure the measurements do not negatively affect the natural traffic in the network, the method discussed but not chosen in the paper is limiting the amount of measurements carried out by the system. The researchers chose not to limit the number of measurements made because the software chosen to carry out the tests did not support this and it would have needed development to change the software to cater for limiting the requests. In addition, since servers that are higher on the DNS hierarchy will receive much more requests than the lower level domains, limiting the requests is not as simple as reducing the overall number of requests made by the system because the distribution of DNS entries is uneven. The higher-level DNS servers will also have a larger capacity to cater for the higher number of requests that they receive. The uneven distribution of capacity in the DNS complicates limiting the requests from the system. Therefore, the decision was made not to limit the requests, but instead to investigate and prove that the requests do not need to be limited because they are not negatively affecting the network.

To prove that the measurements were not negatively affecting the network, the researchers analysed the network statistics. The measurement system runs on the NREN in Netherlands

---

[1] https://www.speedchecker.com/

(SURFnet [2]) and the traffic for the top DNS entries on the network is shown in Figure 2.1. We can see that there is a significant amount of traffic caused by the measurements. However, a previous study has shown that over 900 million queries are made daily to a single top-level domain server. Therefore, the 2 billion daily active measurement requests account for less than 1.6% of the total requests in the network. The DNS management company confirmed to the researchers that although the volume of traffic generated by the measurement system is non-trivial, the network is not negatively impacted. This justification is made with a caveat that because the load is significant on the network, there is a warning to other researchers not to run similar measurements. Thus, the issue of not negatively affecting the traffic caused on the network was not addressed by the study. For this reason, designing a proper query limiting method would have been the ideal option, and would allow other researchers to carry out their own measurements with the same limiting methods in place.

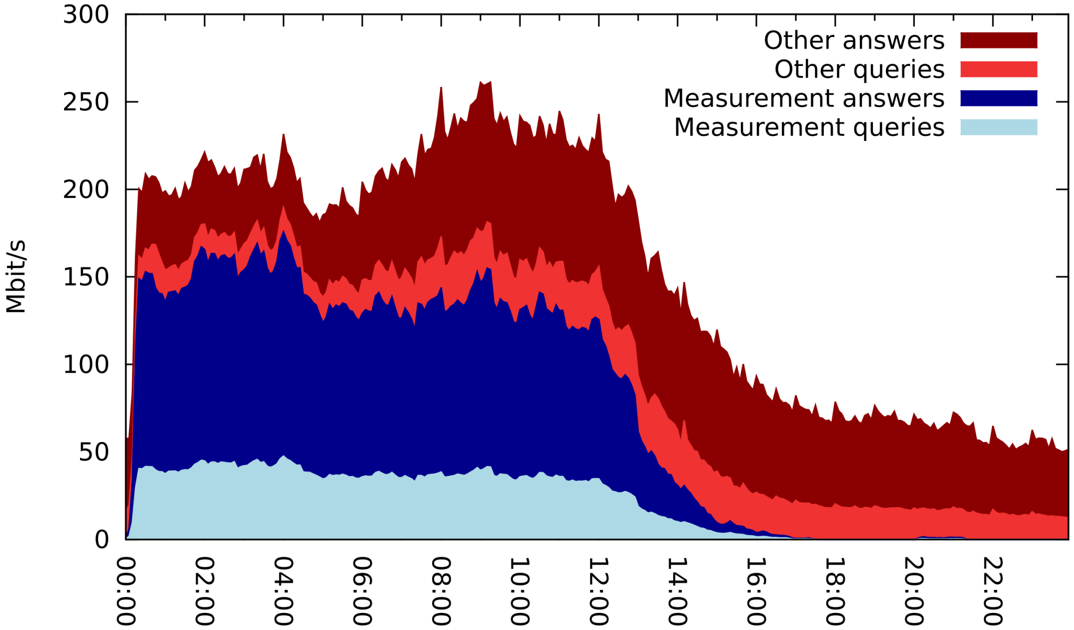


Figure 2.1: Measurement flows versus other flows from the SURFnet network [16].

---

[2] https://www.surf.nl/en

2.1.2 Applications of Passive Measurements

Passive network measurements collect data over time by recording natural network traffic by building software into the network links or by programming the network's routers to send the traffic received to a database. The data typically recorded by passive network monitoring are the number of bytes sent, percentage of packets lost, bandwidth usage, as well as the distribution of the protocols used in the network. The recorded data is generally high level and only records flows individually, which means that the state and behaviour of the network is not recorded. One of the potential issues with passive network measurements is the amount of data that is recorded. A gigabit link can send 250000 packets per second when the link utilisation is 60%. If we store only the IP and transport layer headers which consist of 40 bytes per packet, the system will generate 10 megabytes of data per second, or 36 gigabytes per hour. Consequently, another potential issue is the difficulty in analysing the data captured due to the large volume of data recorded. Since the data recorded is actual user data, there is an ethical concern raised about user privacy. Thus, when carrying out passive measurements, before the data is recorded, the payload data needs to be removed and the traffic needs to be anonymised by removing the IP addresses [4].

The uses of passive network measurements include identifying top ISPs and applications in the network making use of the most bandwidth, finding destination web servers with the most HTTP requests, and carrying out network engineering – finding bottlenecks, finding congestion prone paths for specific source and destinations, and finding alternative paths for these source and destination combinations. From an ISP perspective, passive network measurements can be used for recording the network usage by customers and charging them appropriately – as passive network measurements allow ISPs to charge based on traffic type or the number of bytes used [20].

Data Reduction for Passive Measurements

In order to reduce the amount of data recorded, methods such as sampling, flow aggregation, packet truncation, and filtering are applied. Sampling is the random or pseudo-random selection of data and recording only the selected data. For example, sampling can be carried out by selecting and recording the Nth packet that reaches the router, which will record certain packets in a burst of packets. Time-based sampling will record packets at regular time intervals which runs the risk of missing important packets that are transferred during the period when recording is not operational. Probabilistic sampling records a packet that has a probability of $1/N$ and probability distribution-based sampling records a packets at every X where X is a random variable governed by a specified law (geometric, exponential) and has a mean of N. Flow aggregation is combining and summarising a succession of packets into sessions or flows. This method is common because it is easy to classify network traffic in high-speed networks due to modern hardware geared towards ease of measurement and routers being able to implement flow aggregation. Packet truncation reduces the size of the packet by only recording a part of the packet, such as only a certain number of bytes. Since the number of bytes chosen to record is arbitrary, this might result in the recorded part of the packet including sensitive user information. Thus, common practice is to only record packet headers and ignore the payload. Filtering reduces the amount of data by only recording traffic that meets certain criteria, such as only specific port numbers, IP addresses, protocol, or any arbitrary criteria. This method is useful for clearly defined passive network measurements where only certain traffic is considered relevant. Filtering can be done by software and hardware. However, software filtering is constrained to processing capabilities of the network infrastructure [20], [21], [22]. For our research, the dataset from SANReN was created using flow aggregation and packet truncation.

Passive Measurements applied to SDN

Applying passive measurements is possible on various network infrastructures as shown in a study [23] performed by van Adrichem et al. They present a solution, OpenNetMon, for monitoring networks that use POX [3] – an OpenFlow [24] SDN controller. OpenNetMon allows users to specify the links and their destination switches in the network (link-destination pairs) for different types of traffic, and will constantly monitor throughput, packet loss, and delay on all flows between the specified pairs. Throughput is recorded by polling switches for new statistics, switches return the duration of the flow and the size of the flow as the number of bytes sent, which are then used to calculate per-flow throughput. Different paths can be taken for a link-destination pair due to the nature of SDN, thus, every path between every link-destination pair is polled for statistics. In addition, OpenNetMon increases and decreases the rate of polling as necessary – increasing the rate when flows change their usage trends and decreasing the rate when the performance stabilises. OpenNetMon estimates per-flow packet loss by querying the port statistics of each switch when there is a linear relation to link packet loss and per-flow throughput. When prioritisation occurs based on QoS specifications, the relation of packet loss to per-flow throughput is non-linear. Thus, the first and last switch of each path is polled, and per-flow packet loss is calculated by taking the packet count on the source node, less the packet count on the destination node. One of the challenges of passive network measurements is calculating delay since no packets can be injected and monitored while it is traversing the network. In addition, OpenFlow does not allow packets to be marked with a timestamp upon arrival at a switch nor send duplicate packets back to the controller for the controller to compare the time of sending with the time of arrival. OpenNetMon makes use of OpenFlow's available functions and constantly injects packets into the network at the first switch (switch 1). These probe packets travel the same

---

[3] https://github.com/noxrepo/pox

path as the actual traffic and are sent back to the controller by the last switch (switch 2) so that the controller can calculate the time difference between when the packet was sent and when it came back, and subtracting the estimated additional latency caused by switch-to-controller delay. The switch-to-controller delay is the RTT divided by 2 and is found by injected packets that return to the controller immediately. This will give an estimation of the delay experienced by the actual traffic and results in the formulaic representation for delay being: $t_{delay} = (t_{arrival} - t_{sent} - \frac{1}{2}(RTT_{switch1} + RTT_{switch2}))$. We will mostly focus on the throughput and volume of data sent in our passive measurements, where the throughput is calculated based on the data recorded by SANReN.

2.1.3 SANReN

SANReN is the South African nation-wide network that supplies universities, science councils, science projects such as the Square Kilometre Array (SKA), and various other projects and institutions with broadband internet connectivity. It is operated by the Tertiary Education and Research Network of South Africa (TENET) [4], and primarily funded by the Department of Science and Technology (DST) [5]. SANReN is a nation-wide network that spans across multiple universities around South Africa. Most universities are connected via either 100 Gbps, 10 Gbps, or 1 Gbps links across the land, supplied by telecommunication companies such as Telkom, Neotel, and DFA. There are multiple 10 Gbps undersea cables supplied by West Africa Cable System (WACS), SEACOM, and Eastern Africa Submarine Cable System (EASSy) that link universities via London and Amsterdam. Closer universities are linked wirelessly or share a metropolitan network. SANReN is responsible for overseeing services such as Eduroam [25], the Computer Security Incident Response Team (CSIRT) [6],

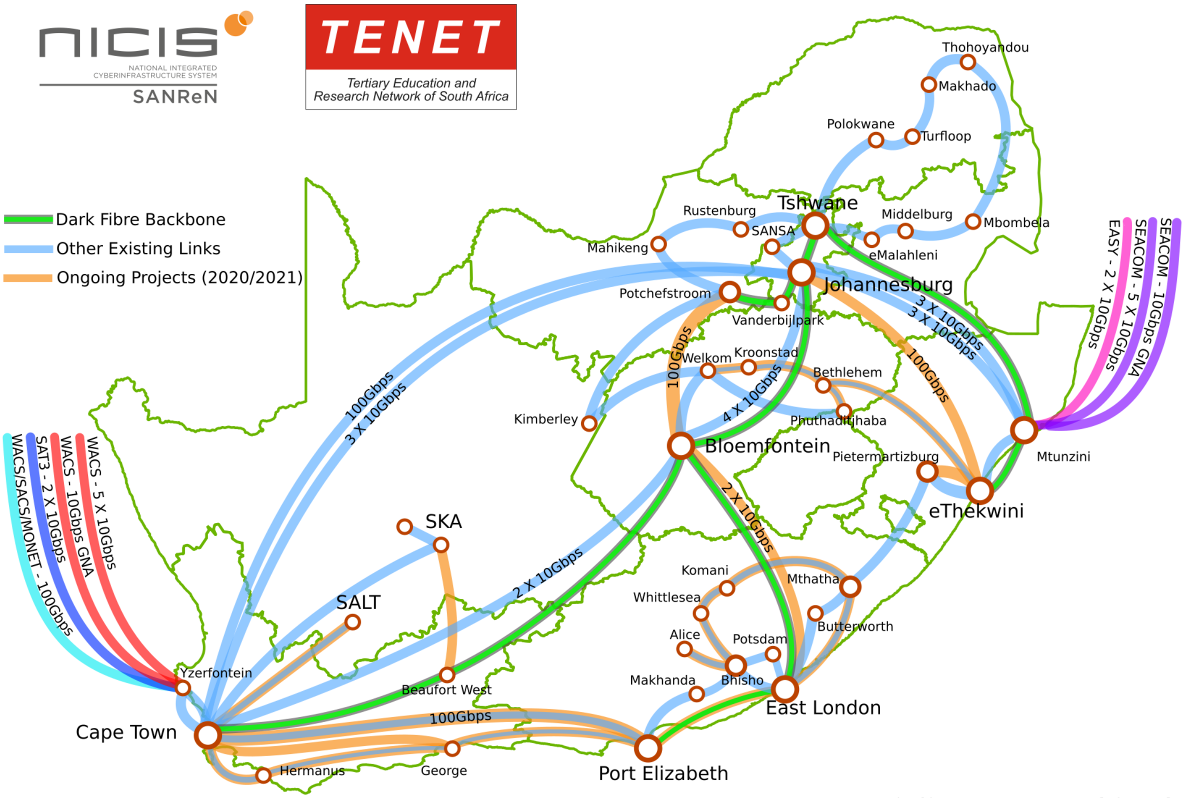
---

[4] https://www.tenet.ac.za/

[5] https://www.dst.gov.za/

[6] https://csirt.sanren.ac.za/

Filesender [7], perfSONAR [19], Performance Enhancement Response Team (PERT) [8], and South African Identity Federation (SAFIRE) [9]. As of 31 March 2019, SANReN has a Total Available Broadband Capacity (TABC) of 3557 Gbps.



South African NREN - 2020-09-10 [John Hay]

Figure 2.2: SANReN backbone map.

Figure 2.2 shows the topology of SANReN and we see that there are some ongoing projects currently taking place on the network. There is an upgrade inserting backbone links with a bandwidth capacity of 100 Gbps which is running across the core national dark fibre paths. This will greatly improve inter-connectivity in the network. Currently, there seems to be a deficiency in the links between Port Elizabeth [10] and the rest of the metropolitan area

---

[7] https://filesender.sanren.ac.za/filesender/

[8] https://sanren.ac.za/services/pert/

[9] https://safire.ac.za/

[10] Gqeberha was named Port Elizabeth during the time of this study

networks. Bottlenecks, operational failures, and load shedding are all factors that could cause the network to perform sub-optimally. These effects are experienced by the 1 million daily end-users that use the services hosted on SANReN.

2.1.4 Summary of Background

In the sections above, we have shown that both active measurements and passive measurements are important methods in measuring network performance. Active measurements are advantageous because they do not require large amounts of additional storage space for their tests results. With SDN, active measurements are also useful in testing specific types of flows and specific parts of the network since tests can be created for specific purposes. Delay is easily measured with active measurements. Conversely, delay is not easily measured with passive measurements since no packets are injected into the network and monitored while they are traversing the network. However, passive measurements are advantageous because the results are naturally occurring flows from within the network that have been tracked. This gives the researcher insight into which types of traffic are most used, which ISPs create the most traffic, and which parts of the network have recurring congestion.

SANReN is the South African nation-wide network that supplies various research institutes. Most universities are connected via either 100 Gbps, 10 Gbps, or 1 Gbps links across the land, supplied by telecommunication companies such as Telkom, Neotel, and DFA. Closer universities are linked wirelessly or share a metropolitan network. As of 31 March 2019, SANReN has a Total Available Broadband Capacity (TABC) of 3557 Gbps which supplies SANReN's 1 million daily end-users. We investigate SANReN's performance using the methods above to find where the network can be improved.

2.2 Related Work

2.2.1 Network Performance Measurements using perfSONAR and Speedchecker

perfSONAR

A *big data* study done by Zurawski et al. [26] shows the potential of perfSONAR by applying it to science related use cases and other approaches to network modelling. As stated in the paper, perfSONAR can execute tests for throughput periodically, as well as uninterrupted checks for latency and packet loss. The inclusion of tools such as Bandwidth Test Controller (BWCTL) [11], One-Way Ping (OWAMP) [12], and Network Diagnostic Tool (NDT) [13] allows perfSONAR to conduct tests for specific types of traffic such as bulk data transfer and video transfer. The positioning of the nodes in the network that have perfSONAR installed is important in achieving optimal results. Good practice is installing perfSONAR on nodes that are located close to the source of the measurements and responsible for traffic exchange, i.e.: the edge of a university network. Using perfSONAR, the researchers conducted performance measurements transferring data between Brown University (in the United States) and the Large Hadron Collider at the European Organization for Nuclear Research (CERN). The data transferred between the two institutions is typically hundreds of gigabytes or several terabytes in size. The performance measurements showed that flows originating from CERN going to Brown University was significantly slower than the speed of outbound data from Brown University. Upon further investigation, the congestion was found to be created by the security measures of the university's network. The network was amended and dedicated

---

[11] https://software.internet2.edu/bwctl/

[12] https://software.internet2.edu/owamp/

[13] https://software.internet2.edu/ndt/

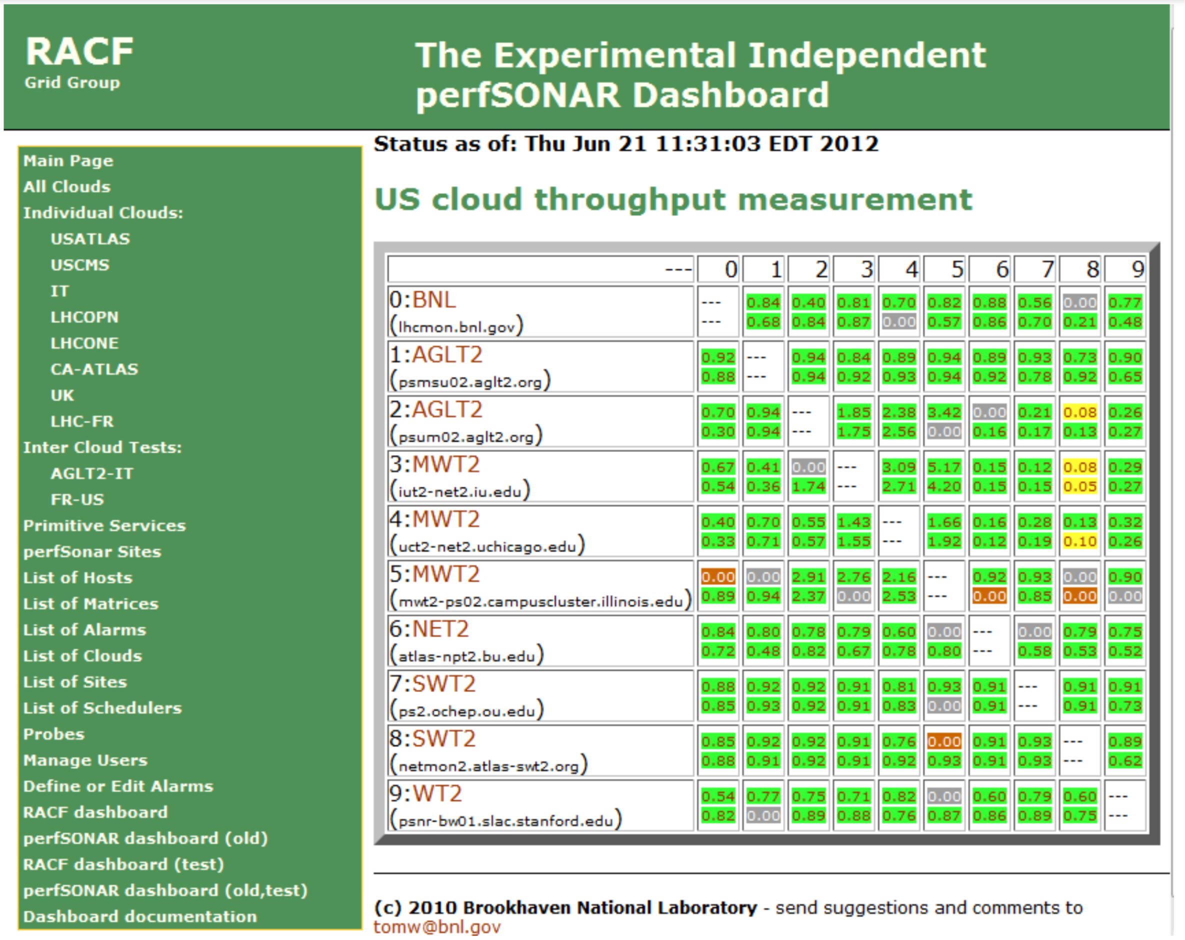
paths were created for these data-intensive traffic transfers. In our research, we conduct similar performance tests focusing on finding congestion prone paths in SANReN.

Another project making use of perfSONAR for network monitoring is the US ATLAS [14] project. McKee et al. [27] studied the perfSONAR implementation as the US ATLAS project was one of the first projects to make use of the perfSONAR tools. The measurements that are done are: available bandwidth, round-trip latency, one-way latency (and packet loss), and traceroute. The bandwidth test runs every 4 hours from each host to every other host in the network, running for 20 seconds using TCP. The round-trip latency is measured every 5 minutes and sends 10 packets of 1000 bytes every second. One-way latency is measured by sending 10 UDP packets per second of 20 bytes in size, whilst recording the time it is sent and the time it reaches the destination. Every 10 minutes a traceroute is carried out by sending 40-byte UDP packets from source to destination. The testing methodology advises each site to maintain two hosts that perform the network monitoring — a host for latency measurements and a host for bandwidth measurements. This allows the two tests to be carried out concurrently without interfering with one another. It will also mitigate the risk of tests being inaccurate due to external reasons. As perfSONAR was implemented in the federated US ATLAS network architecture, the researchers found that the performance results were hard to compare because all the sites were independently collecting results.

To get around this, a dedicated monitoring system was created, which collects the performance data and transfers it to a centralised database. The solution, shown in Figure 2.3, also included a dashboard which is capable of showing the performance statistics between all the nodes in the network at once. The shortcomings of perfSONAR mentioned in the paper are that the bandwidth tests would sporadically fail to run. In addition, long running tests and many concurrent tests may cause the perfSONAR services to experience reliability

---

[14] https://po.usatlas.bnl.gov/



RACF
Grid Group

The Experimental Independent perfSONAR Dashboard

Main Page
All Clouds
Individual Clouds:
- USATLAS
- USCMS
- IT
- LHCOPN
- LHCONE
- CA-ATLAS
- UK
- LHC-FR

Inter Cloud Tests:
- AGLT2-IT
- FR-US

Primitive Services
perfSonar Sites
List of Hosts
List of Matrices
List of Alarms
List of Clouds
List of Sites
List of Schedulers
Probes
Manage Users
Define or Edit Alarms
RACF dashboard
perfSONAR dashboard (old)
RACF dashboard (test)
perfSONAR dashboard (old,test)
Dashboard documentation

Status as of: Thu Jun 21 11:31:03 EDT 2012

US cloud throughput measurement

---	0	1	2	3	4	5	6	7	8	9
0:BNL (lhcmon.bnl.gov)	--- ---	0.84 0.68	0.40 0.84	0.81 0.87	0.70 0.00	0.82 0.57	0.88 0.86	0.56 0.70	0.00 0.21	0.77 0.48
1:AGLT2 (psmsu02.aglt2.org)	0.92 0.88	--- ---	0.94 0.94	0.84 0.92	0.89 0.93	0.94 0.94	0.89 0.92	0.93 0.78	0.73 0.92	0.90 0.65
2:AGLT2 (psum02.aglt2.org)	0.70 0.30	0.94 0.94	--- ---	1.85 1.75	2.38 2.56	3.42 0.00	0.00 0.16	0.21 0.17	0.08 0.13	0.26 0.27
3:MWT2 (iut2-net2.iu.edu)	0.67 0.54	0.41 0.36	0.00 1.74	--- ---	3.09 2.71	5.17 4.20	0.15 0.15	0.12 0.15	0.08 0.05	0.29 0.27
4:MWT2 (uct2-net2.uchicago.edu)	0.40 0.33	0.70 0.71	0.55 0.57	1.43 1.55	--- ---	1.66 1.92	0.16 0.12	0.28 0.19	0.13 0.10	0.32 0.26
5:MWT2 (mwt2-ps02.campuscluster.illinois.edu)	0.00 0.89	0.00 0.94	2.91 2.37	2.76 0.00	2.16 2.53	--- ---	0.92 0.00	0.93 0.85	0.00 0.00	0.90 0.00
6:NET2 (atlas-npt2.bu.edu)	0.84 0.72	0.80 0.48	0.78 0.82	0.79 0.67	0.60 0.78	0.00 0.80	--- ---	0.00 0.58	0.79 0.53	0.75 0.52
7:SWT2 (ps2.ochep.ou.edu)	0.88 0.85	0.92 0.93	0.92 0.92	0.91 0.91	0.81 0.83	0.93 0.00	0.91 0.91	--- ---	0.91 0.91	0.91 0.73
8:SWT2 (netmon2.atlas-swt2.org)	0.85 0.88	0.92 0.91	0.92 0.92	0.91 0.91	0.76 0.92	0.00 0.93	0.91 0.91	0.93 0.93	--- ---	0.89 0.62
9:WT2 (psnr-bw01.slac.stanford.edu)	0.54 0.82	0.77 0.00	0.75 0.89	0.71 0.88	0.82 0.76	0.00 0.87	0.60 0.86	0.79 0.89	0.60 0.75	--- ---

(c) 2010 Brookhaven National Laboratory - send suggestions and comments to tomw@bnl.gov

Figure 2.3: perfSONAR dashboard showing the throughput matrix in the US ATLAS network [27].

issues due to completed tests not being cleaned up. Regardless, the researchers claim that perfSONAR is a vital tool in network monitoring and the solution has been adopted by multiple other projects. In our research, we will not schedule too many tests on a single node to avoid fail over. We will not make use of a dashboard as we are not conducting live measurements, but rather analysing data based on the results from the experiments.

There are a number of studies using perfSONAR, implementing the framework in different ways: [28], [29], and [30]. Chen [28] creates a browser extension which measures performance of web-servers using perfSONAR. The extension allows network administrators to quickly find the performance of their web-servers. Zhang et al. [29] use perfSONAR to identify and distinguish bottlenecks within the network from anomalies of slow network performance. The network is monitored and possible problematic areas are identified. The reseachers then conduct perfSONAR one-way latency tests to confirm whether the path has a bottleneck. Campana et al. [30] implement perfSONAR conventionally to monitor the performance of sites within the Worldwide LHC Computing Grid (WLCG). While these are all implementations of perfSONAR, to the best of our knowledge, this is the first study using perfSONAR to conduct active performance measurements on SANReN.

Speedchecker

Arnold et al. conducted a study [31] using Speedchecker to find out the impact of a private WAN on cloud performance. Speedchecker was chosen as the primary tool for conducting measurements in this study because its API allows users to run measurements such as ping, traceroute and HTTP GET (page load time). Moreover, it allows users to conduct measurements from ASes that host an estimated 91% of internet users. The study shows that Speedchecker has the most coverage compared to other popular network measurement platforms. Ripe Atlas [32] and PlanetLab [33] are the other two platforms considered and offer significantly smaller coverage than Speedchecker. At the time of this study (July 2020), PlanetLab had 42 active sites and Ripe Atlas had 10 thousand, whereas Speedchecker had over 1 million probes, and more than 56 thousand available on demand. Speedchecker covers 4 times more ASes than Ripe Atlas and has 35 times more probes. Since the number of tests permitted per day with a Speedchecker account is limited, the researchers select different city and AS combinations every day to get as much coverage as possible, then restarted the selection process once all the vantage points have been used. Measurements are issued 10 times per day, from each city and AS combination to 8 destinations. Five pings tests, one traceroute, and one HTTP GET (of a 10 Mb file) are requested for each destination. The large file was chosen for the page load time test to get a better idea of performance when a sustained file transfer occurs. The requests are made to IP addresses to remove the DNS resolution overhead and load balancing, so that the routing performance is the main focus. We do not resolve the DNS websites because we are conducting a QoE experiment and trying to simulate the end-user's experience. In addition to Speedchecker having the most coverage of the tools available, it also allows us to conduct measurements from outside of SANReN whilst specifying the source city.

Other studies that have used Speedchecker for performance measurements are [34] and [35]. Formoso et al. [34] and Chavula et al. [35] use Speedchecker to conduct performance tests between countries within Africa, and found that certain countries within the continent (Benin, Egypt, and South Africa for example) have reached levels of delay similar to developed countries. The researchers found that there are clusters within Africa with low delay inter-connectivity, however, between the clusters, there are high delays. Similar to our study, these have been done in the African context. However, to the best of our knowledge, our research is the first to focus on SANReN using Speedchecker for active measurements.

2.2.2 Effects of the Covid-19 Pandemic on Network Traffic

Since the Covid-19 pandemic altered the lives of billions of people all over the globe, and there was a big expectation of a shift in internet usage worldwide, a few studies have been conducted focusing on the effects on network performance, the effects on e-learning, the response shown by ISPs, social media usage, and traffic volume and type. These studies have been conducted in different contexts and countries that have different network infrastructures compared to South Africa.

Effects of the Covid-19 Pandemic on European Networks

The change in network usage during the pandemic has been looked at by Feldman et al. [1]. This study focused on three main areas. Firstly, a large ISP in Central Europe with over 15 million fixed line subscribers. Secondly, three major Internet Exchange Points (IXPs) across Europe and the US (with peak traffic of 8 Tbs, 500 Gbs, and 600 Gbps each). Finally, an educational network in Madrid with 290,000 users and a total of 5.2B flows of inbound and outbound traffic. The aim of the paper by Feldman et al. was to analyse the effect of the lockdown across Europe on the internet traffic. The analysis in the paper is broken down in various ways — namely by network, transport layer protocol, type of traffic, VPN traffic, and educational network traffic. The comparisons carried out are by week; showing the volume of traffic from before the lockdown period, during the lockdown, and after. The results shown in Figure 2.4 show an increase in traffic of 15 — 20% within days of the beginning of the lockdown.

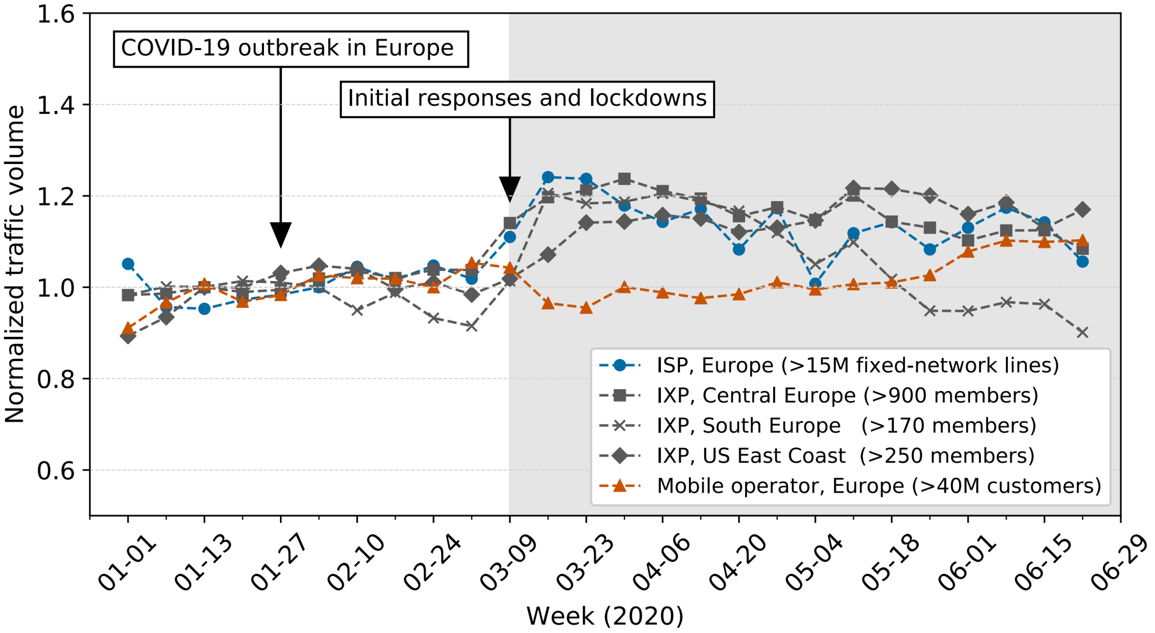


Figure 2.4: Normalised traffic changes during 2020 [1].

Under normal conditions, ISPs and Content Delivery Networks (CDNs) expect a traffic increase of 30% in a year, meaning the 15 — 20% increase after lockdown is substantial. ISPs that had not provisioned for unexpected increases beforehand had to quickly add capacity. Educational traffic and remote working traffic such as VPN traffic and traffic using video conferencing software increased by over 200%. However, the study also found that the difference in the peak amount of traffic was small due to the pandemic increasing traffic during work hours, over a sustained period of time, which is easily handled by networks that have provisions for unanticipated traffic surges by 30%. For network operators, some port capacities needed upgrading to be able to handle the increase in traffic. Overall, networks managed to handle the increase in traffic over the lockdown and pandemic periods. The study conducts a thorough analysis of the network traffic in Europe over the pandemic period. It shows the network patterns with clearly described methodologies. The researchers normalise the data for smoothing, however, since our data comes from a single dataset, we do not normalise the data. We also breakdown the data by finding which countries, ISPs and protocols produce the highest volume of incoming and outgoing traffic.

Candela et al. conducted a study [36] focusing on latency changes in Italian networks during the pandemic. The researchers made use of Ripe Atlas to carry out their latency measurements. Ripe Atlas is capable of carrying out two types of measurements – Anchoring Measurements (AMs) and User-Defined Measurements (UDMs). AMs are carried out automatically from Ripe Atlas probes to their pre-defined destination nodes, called anchors, which are located in IXPs, data centres, or ISP operation centres, allowing users to monitor the network infrastructure. UDMs allow users to choose their own destinations for measurements to be carried out. Ripe Atlas UDM results of all UDMs are public and can be seen by anyone, it is not restricted to the user who initiated the measurement. The latency is measured by conducting a conventional network ping (with ICMP) to the specified destinations.

The researchers decided to classify the minimum, average, and maximum delay individually, meaning that they would be able to get a deeper analysis of the network performance. They also consider path changes as a potential factor that would cause inaccuracies in the results. To circumvent this issue, they factor in a margin of error by using the minimum time between two nodes as a baseline and adding the minimum delay to the delay experienced between the nodes. Finally, in order to assess the effects of the lifestyle changes at the country or city level, the adjusted delay values are grouped into buckets of 30 minutes in duration and averaged out. The results of the study showed that the latency gradually grew over time as Italy phased in the lockdown across the country. When schools began to close, higher delay peaks were observed, and on the day that the complete lockdown started, there was a 30% increase in the minimum delay compared to before the lockdown. In addition to the increase in peak latency, there was an increase in variability in the latency values. In terms of packet loss, it also increased by $2-3$ times compared to before the pandemic. The dataset we received from SANReN has also been grouped into buckets of $30-35$ minutes in duration. However, the only data extracted from the buckets are summaries showing the total flows, packets, and averages of bits per second, packets per second, and bytes per packet. This does not give us a detailed view of the network's performance.

Facebook's Worldwide Edge Network

For a broader perspective on the effects of the pandemic on the internet, Böttger et al. conducted a study [11] reviewing the network usage on Facebook's Edge Network – edge traffic is traffic flowing between end-users, ignoring traffic flowing between and inside data-centres. There are 2.5 billion monthly active users around the world on Facebook's Edge Network. In terms of the usage, the study shows (Figure 2.5) that there is a gradual increase in traffic in the first half of March, a considerable increase in traffic in the second half of

March, followed by a plateau through April until the end of July as people adapted to a new way of life, and at the end of the study period, a decrease in traffic volume is recorded. This shows that some of the traffic growth was due to people trying to adapt to life in isolation.

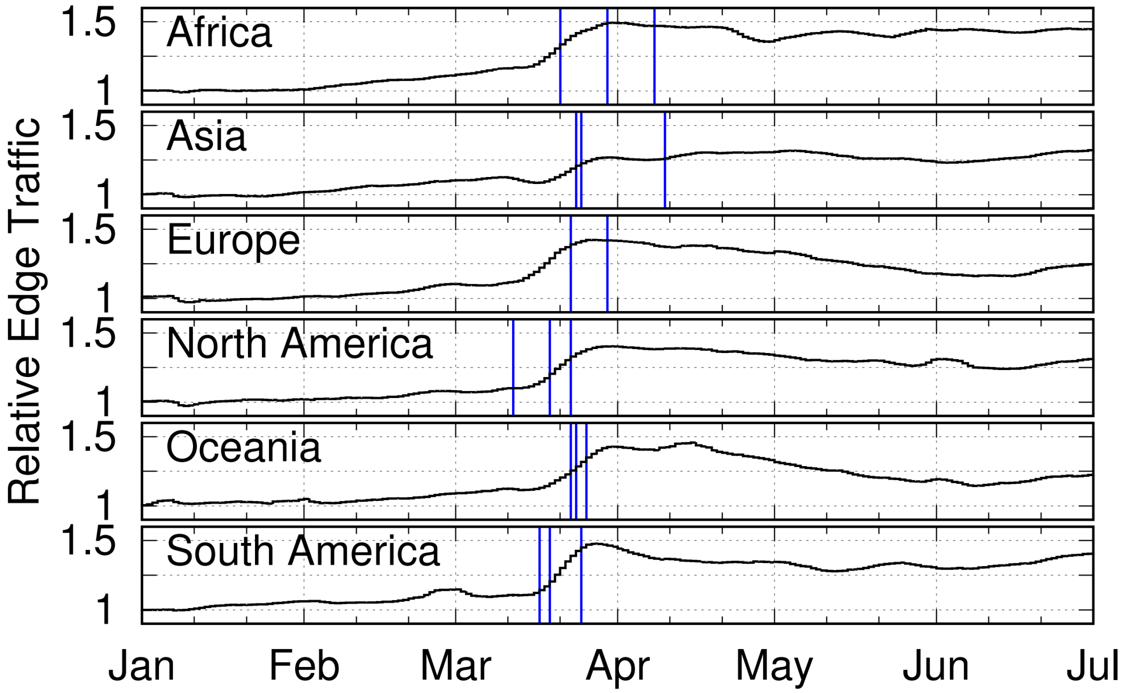


Figure 2.5: Traffic growth per continent, the start of lockdown indicated by vertical blue lines [11].

In terms of types of traffic that users consumed, there was an exponential surge in live-streaming products, however, without largely affecting the overall traffic. Messaging services traffic grew globally, with the highest growth rates in Europe and South America, and with India and Sub-Saharan Africa experiencing the biggest percentage increase in overall observed traffic. There was a global 5% increase in video traffic, with a higher rate of increase in regions like North America, Europe, and India, and lower increase rate in Africa. The increase in traffic volume was mostly experienced by broadband networks, whereas mobile networks remained stable besides experiencing a small increase in the second half of March. Since video accounts for a large portion of total global traffic, the study finds out its QoE during the pandemic. The findings were that while networks in some developed regions such as North America and Europe were stable, networks in developing regions like India, South America and Sub-Saharan Africa experienced a degradation in QoE. With respect to the RTT, similar results are found, as there is an increase in average RTT globally in the second half of March, and a decrease from April onwards. Once again developed countries experience similar RTT compared to before the pandemic, whereas South Africa and India experience much higher RTTs. The increase in RTT and the degradation in video QoE can be caused by the limited capacity in the network which, in turn, causes congestion.

2.2.3 Summary of Related Work

In the studies reviewed, we noted previous applications of perfSONAR and Speedchecker, and discussed their strengths and limitations. We found that for previous studies, measurement tests conducted with perfSONAR have failed sporadically, and running many tests concurrently could cause issues with garbage collection. We will avoid the scheduling issues by refraining from overloading any perfSONAR nodes. With Speedchecker, we are limited in the number of requests we are allowed to carry out per day. As such, we will have to calculate the number of tests that are conducted for each source and destination and for each type of test.

The current climate of the Coronavirus Pandemic has shifted the way we work and use the internet. We reviewed studies that focused on internet usage during the pandemic from different perspectives. In terms of volume, European networks experienced a 15 - 20% increase in total volume withing days after lockdown started. This is a substantial amount given that under normal circumstances ISPs expect an increase of 30% a year. Globally, Facebook found that there was an exponential surge in usage of live-streaming products, with large increases in usage of messaging services as well. RTT and video streaming QoE were negatively affected by the pandemic. There was a global increase in RTT and a degradation of QoE in video streaming which was caused by congestion due to the limited network capacity. We are conducting our research during the pandemic as well, and will broadly consider the impact of the Coronavirus on internet usage.

Chapter 3

Methodology

The research consists of three parts – conducting active network measurements to assess QoS, using passive network measurements to assess flow data, and testing an SDN solution that could potentially address performance issues found within SANReN. Firstly, we try to answer the first research question by conducting experiments to measure performance of SANReN in terms of packet delay, page load time, and throughput. We also consider location of web servers, inter-connection between network operators and SANReN, as well as the routes and number of hops taken for packets to travel from source to destination.

To tackle the second research question, we conduct passive network monitoring by analysing a netflow dataset collected from SANReN gateways. We look for trends of how the network is used in terms of volume, speed, Internet Service Providers (ISP), protocols, and countries. These trends help to identify which facets of the network are most used, how the volume of traffic changes over time, and what are the busy periods and the lower usage periods of the network. The SANReN dataset also gives us the throughput (bits per second) of each flow, which we use to find correlations between the volume and performance.

Finally, we propose an SDN model of SANReN to answer our third research question. We create a virtual implementation of the model, using parameters from the performance results of the active and passive measurements. Thereafter, we conduct the performance tests on the virtual network and compare its performance to that of the SANReN architecture.

3.1 Active Measurements Methodology

Active measurement experiments were carried out from sources outside of SANReN (external) and sources within SANReN (internal) to find how the network performs for its users. We outline the methodology used for the active measurements and the types of tests in this section. We begin by describing the types of tests carried out, followed by the tools used to conduct the experiments, then the procedure used to carry out the tests.

3.1.1 Types of Performance Tests

The independent variable for our tests is the chosen destination web server and the dependent variables are the respective metrics. The control variables are the source city and the platform used to conduct the test.

Delay

The network delay to each web server from different source locations tells us how long it takes for packets to reach the destination from different vantage points. Thus, delay is one of the key metrics used to test network performance. Delay also influences traffic engineering decisions such as routing based on application or bandwidth requirements [37], [38], [39]. We conduct delay (ping) tests to various zero-rated web servers in South Africa that reside inside SANReN to determine packet delay. The delay to each web server shows us the performance in terms of average travel time for packets to be transported across the network. The dependent variables for the delay tests are the round-trip time (RTT), and the percentage of packets lost. This determines the reliability of the network in different paths, and informs us of congestion issues that exist in the network. In addition to the time it takes to reach the web server, we also keep track of the percentage of packets lost for each city.

Page Load Time

Page load time (PLT) is the total time it takes for a web server to load from when the user enters the URL in a browser, until the page is loaded completely. PLT is a key metric in finding the QoS of web servers because it shows us the network performance when actual data is requested by an end-user (it tells us how long the user has to wait for the network to load a page) [40]. Moreover, we use page load times because studies have shown that user tolerance of page load time is an important driving factor for web server usage [41]. We access the same web servers from each city to get the performance from different cities, as well as how the web servers perform when accessing them from outside SANReN and from within [42]. The dependent variable for the page load time tests is the total page load time. We also make use of the total downloaded bytes (TDB) and the time-to-first-byte (TTFB) in our analysis.

Path Analysis (Traceroute)

We conduct traceroute tests to the web servers and use the results to find possible reasons for high delays and high PLTs observed from the previous tests. By analysing traceroute results, we find gaps in the network and identify congestion prone paths. The traceroute also tells us whether routes are being chosen by the network that are circuitous or non-optimal. The dependent variable for these tests is the number of IP hops it takes to travel from each source to the specified destination. We define a successful traceroute as a traceroute that reaches the target web server's ASN. Using MaxMind's ASN database, we look up the target web server's ASN as well as the last IP that the traceroute reached. We check if the two ASNs are the same and save the traceroute as successful or unsuccessful. We plot our traceroute results using only the successful tests.

Throughput

Throughput is used to tells us how fast the network transfers data between two points and where the network needs reinforcing due to insufficient bandwidth capacity [43], [44]. We use different methods to measure throughput for internal and external tests due to limitations in the measuring platforms available. For external tests, we calculate throughput from the page load time tests by using the total number of bytes downloaded for each test. We define throughput for these tests as the number of total downloaded bytes (TDB) divided by the result of total page load time minus time-to-first-byte (TTFB), i.e.: $Throughput = \frac{TDB}{PLT-TTFB}$. For the PLT tests, the median amount of data being transferred is 31594 bytes, thus, the throughput results are based on 31594 bytes of data. For the internal tests, we use Iperf3 [45] to measure the throughput.

3.1.2 Performance Measurement Platforms

Two platforms are used to conduct the active performance measurements: Speedchecker and perfSONAR. These platforms are outlined below.

Speedchecker — External Measurement Platform

Speedchecker is a global network measurement platform that operates in over 170 countries. There are three different types of devices in the Speedchecker network: PC probes — Windows computers; Android probes — mobile phones and tablets; and Router probes — DD-WRT routers in customer homes. Speedchecker probes are placed on end-user devices and this allows a more accurate measurement of end-user experience. Speedchecker allows a researcher to specify the source probes from which tests are run, allowing us to run tests from different cities using probes that reside outside of SANReN, which means that the measurements will indicate the performance when accessing SANReN from an external network. Speedchecker supports tests for ping, page load time, and traceroute, but does not have a dedicated test for throughput. Hence, we calculate throughput from the PLT tests by dividing each test's values for TDB by the total page load time minus the TTFB.

PerfSONAR — Internal Measurement Platform

PerfSONAR is also a global network measurement framework that operates within NRENs. We thus use it to conduct internal performance tests from sources within SANReN. This gives us a representation of how the network performs when transporting packets between destinations that are within the network. We use these measurements to find gaps in the network as well as compare the performance of data transfer, both internally and externally. PerfSONAR has dedicated tests for delay (ping), page load time, traceroute, and throughput. However, perfSONAR does not provide the value of total downloaded bytes for each PLT test, thus we measure throughput using the built-in throughput test that uses Iperf3. Moreover, we were not able to get successful tests from perfSONAR nodes to destinations that are not also perfSONAR nodes. This means that we could not gather data for throughput to the zero-rated web servers from internal sources. We thus gathered our throughput data based on source and destination nodes that are hosted by perfSONAR. We use the internal (PerfSONAR) and external (Speedchecker) tests to compare the network performance from different vantage points.

3.1.3 Performance Experiments Outline

Using both Speedchecker and perfSONAR, each test is run from Cape Town, Johannesburg, Durban, Port Elizabeth, and Pretoria. These are the cities with the top five population sizes in South Africa. They are also spread across the country, and this gives us a better indication of performance for different parts of the country. In addition, the web servers that we used as destinations are chosen based on where they are hosted — web servers from each of the source cities are used, and their locations were checked using IP-API and IPWHOIS. Using multiple source locations gives us a good representation of performance to and from different parts of the country. The universities that were used as destinations (shown in Table 3.1) were chosen because they are seven of the biggest universities in South Africa and are spread out across the country. The web servers used as destinations are hosted by these universities as part of SANReN.

Table 3.1: Destination universities used in this study.

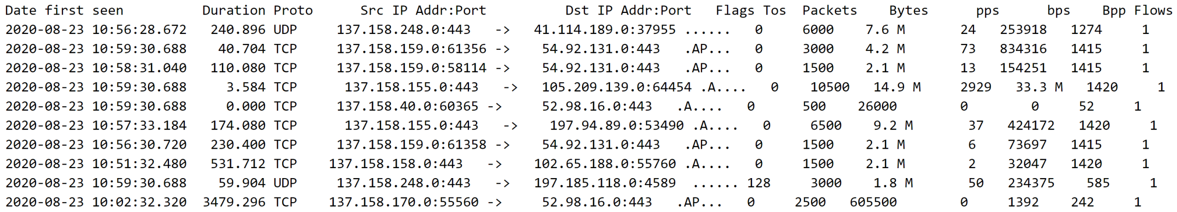
Abbreviation	Name	Location
UCT	University of Cape Town	Cape Town
UWC	University of the Western Cape	Cape Town
WITS	University of the Witwatersrand, Johannesburg	Johannesburg
UJ	University of Johannesburg	Johannesburg
DUT	Durban University of Technology	Durban
UNISA	University of South Africa	Pretoria
NMU	Nelson Mandela University	Port Elizabeth

We ran daily tests from each source city to each destination university. In order to comply with Speedchecker's fair-usage policy, we limited our tests to 14 per day per destination university. Since five source cities and seven destination universities were used for this study, 14 tests per university meant 490 tests per day. In addition, since the network has different usage levels at different times of the day, tests were conducted in the morning (9AM), afternoon (3PM), and night (9PM) to get a measure of the network performance.

Speedchecker initialises tests and returns a status and a test ID, where the status states whether the test started successfully or not. We fetch the results using the test ID and write a JSON string for each test. We parse the JSON files and collect the data into CSV files. For each type of test, we use a python script to analyse the CSV files. PerfSONAR runs the test and returns results immediately as plain text. Thus, we use a script to extract the results from the text and input them into CSV files for analysis. For all the experiments, the final processed CSV files contain the city used as the source, the target web server, the university in which the web server resides, and the result for the measurement — delay (ms), PLT (ms), number of IP hops, or throughput (bytes/ms). A failed test results in an empty value for the metric that was measured. For traceroute experiments, the final result files also contain the target web server ASN as well as the ASN that the IP of the final hop resides in. We compare these ASNs to find out whether the traceroute was successful or not. For external page load time tests, the final processed files also have columns for TTFB, TDB, and calculated throughput.

3.2 Passive Measurements Methodology

We received a netflow dataset collected by SANReN to analyse usage of the network. The data from SANReN consists of inbound and outbound flow data for UCT, with the following information recorded: date first seen, duration, protocol, source IP with port, destination IP with port, flags, 'tos', packets, bytes, packets per second, bits per second, bytes per packet, and flows. The data is also grouped into buckets and summarised after each bucket. The duration of each bucket is 30 – 35 minutes and the summary given shows the time period, total flows, total packets, average bits per second, average packet per second, and average bytes per packet. We cleaned the dataset using python scripts and inserted the data into a database. We then queried the database summarising the data in various meaningful ways, to create plots and visualise the network usage patterns.



```
Date first seen          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Flags Tos  Packets    Bytes      pps      bps    Bpp Flows
2020-08-23 10:56:28.672   240.896 UDP     137.158.248.0:443   ->   41.114.189.0:37955 ......   0     6000    7.6 M      24   253918   1274     1
2020-08-23 10:59:30.688    40.704 TCP     137.158.159.0:61356 ->    54.92.131.0:443   .AP...   0     3000    4.2 M      73   834316   1415     1
2020-08-23 10:58:31.040   110.080 TCP     137.158.159.0:58114 ->    54.92.131.0:443   .AP...   0     1500    2.1 M      13   154251   1415     1
2020-08-23 10:59:30.688     3.584 TCP      137.158.155.0:443    ->  105.209.139.0:64454 .A....   0     10500   14.9 M    2929   33.3 M   1420     1
2020-08-23 10:59:30.688     0.000 TCP     137.158.40.0:60365 ->     52.98.16.0:443   .A....   0      500    26000       0        0     52     1
2020-08-23 10:57:33.184   174.080 TCP      137.158.155.0:443    ->   197.94.89.0:53490 .A....   0     6500    9.2 M      37   424172   1420     1
2020-08-23 10:56:30.720   230.400 TCP     137.158.159.0:61358 ->    54.92.131.0:443   .AP...   0     1500    2.1 M       6    73697   1415     1
2020-08-23 10:51:32.480   531.712 TCP    137.158.158.0:443   ->   102.65.188.0:55760 .A....   0     1500    2.1 M       2    32047   1420     1
2020-08-23 10:59:30.688    59.904 UDP     137.158.248.0:443   ->  197.185.118.0:4589  ...... 128     3000    1.8 M      50   234375    585     1
2020-08-23 10:02:32.320  3479.296 TCP    137.158.170.0:55560 ->     52.98.16.0:443   .AP...   0     2500   605500       0     1392    242     1
```

Figure 3.1: Excerpt of raw SANReN flow dataset.

SANReN provided the data for traffic from 1 July 2020 — 31 August 2020. Therefore, our study is restricted to this two-month period. To clean the data, we ignore the unwanted columns such as flags, 'tos', and ports because these are not relevant to our research. In the raw data, some columns denote megabyte values and gigabyte values with an *M* or a *G* respectively, which stops us from querying the column as a numeric field, and thus, we multiply the value of by 1000000 or 1000000000 to get the bytes value accordingly. Once the data is cleaned, it is stored in a SQL database table with the structure shown in Table 3.2 below.

In addition to cleaning the data, we perform IP lookups for each source and destination IP to find the ASN, ISP, and geo-location — country, city, latitude, and longitude. With this information, we can extract the trends of the network by ISP and country, as well as extract the volume per periods, countries, and ISPs. To lookup the IP, we use MaxMind [1] as our primary tool, making use of their ASN, city, and country databases. For IPs [2] that are not found in MaxMind's databases, we conduct an online lookup using IPWHOIS [3] and IP-API [4]. Since each flow either originates from UCT or flows to UCT, we do not store the geo-location information about the UCT side of the traffic because it is known and would be

---

[1] https://www.maxmind.com/en/geoip2-services-and-databases

[2] Anonymised and only resolved to their subnets

[3] https://ipwhois.io/documentation

[4] https://ip-api.com/docs

Table 3.2: Database table structure of cleaned data.

Column	Type	Length	Precision	Scale
Time	datetime	8		
Duration	float	8	53	
Protocol	varchar	4		
SourceIP	varchar	50		
DestIP	varchar	50		
Packets	varchar	50		
Bytes	float	8	53	
PacketsPS	varchar	50		
BitsPS	decimal	9	15	3
BytesPP	varchar	50		
Inbound	bit	1		
Country	varchar	50		
CountryCode	varchar	2		
Latitude	varchar	20		
Longitude	varchar	20		
ASN	varchar	6		
ISP	varchar	250		

repeated data since the IPs are anonymised. Instead, we mark flows as inbound or outbound, and store the non-UCT IP's details only. By marking the traffic as inbound or outbound, we are able to compare traffic that comes into UCT with traffic that goes out of UCT.

We performed our analysis using the columns in Table 3.2. We used the sum of the bytes column to calculate the volume of data transferred over the two-month period. We present our analysis of the results grouped by criteria such as ISP, country, and protocol. We also use the bits per second column to find the throughput experienced grouped by the same criteria. We use the volume analysis to make deductions since it shows the trends of the network usage over the time period studied. We also compare traffic volumes against the throughput to show where the network is experiencing performance issues. With passive monitoring, we can also see what the QoE is like for end-users of the network as the data being analysed is actual end-user data. With these trends, we can highlight the QoS for SANReN and we can compare these findings to the active measurements.

3.3 SDN Solution Setup

We use the results from our active measurements and the SANReN backbone map (Figure 2.2) to configure a virtual network using an SDN design. We conduct active measurements on this virtual network and compare the performance of the virtual network to that of SANReN performance results. This allows us to determine whether the network can be improved by SDN.

3.3.1 Virtual SDN Configuration

To create the SDN topology, we use Mininet [46] and Ryu [47], where Mininet creates the virtual network and Ryu acts as its controller. We configure our network to use a single controller with OpenFlow 1.3 [24] as the underlying protocol. We use the SANReN backbone map to model our virtual topology.

Topology

Our virtual SDN topology consists of all the nodes and links shown on the SANReN backbone map (Figure 2.2). Due to limitations of simulation, we scale down the bandwidth of SANReN by 1000, so that each network link has a capacity equal to 0.001% of their real-world capacity. We simulate the universities and cities that were used in our active measurements by attaching hosts to the CPT, PE, JHB, eThekwini (Durban), and Tshwane (Pretoria) switches — where 2 hosts each are attached to CPT and JHB. Multiple nodes in the network have multiple paths between them, resulting in loops in the network. Therefore, under normal circumstances, some hosts would be unreachable in the network. To manage the loops, we use make use of a Spanning Tree Protocol (STP) implementation of the Ryu controller. STP is a protocol that removes loops in a topology by closing selected ports in the network that are causing loops [48]. STP also provides fault tolerance by having redundant links that can be used as backup when a link failure occurs. The policy for opening and closing ports in the SDN model is based on the cost of the path. We adopt the STP implementation in Ryu, which allocates costs to the paths based on link capacities. Low bandwidth capacity means higher cost of traversing a path [47].

Link delays

In terms of delays between nodes, we configure the network to use the delays obtained from our internal active measurements where applicable. For nodes that we did not conducted measurements on, we use the latitude and longitude between the two points and calculate the delay based on the distance between them. From previous research, 5 ms is an accurate measure of delay between two points using single-mode optical fibre links [49]. We adjust this baseline based on our active measurements results. We modelled the delay between these nodes assuming a 6.5 ms delay for a distance of 1000 km between two nodes. We calculate the delays between each host and switch by using the results from our internal active measurements. We use the minimum delay achieved from the active measurement experiments between two points to model the delay in our virtual network. Link delays in Mininet are one-way, thus, we set the link delay between a host and a switch as the round-trip time between a university and the city it resides in, divided by two. Where we have two options to calculate the delay, we use the minimum of the options available. For example, to set the delay between Cape Town and Johannesburg, we have results for the delay between WITS and Cape Town as well as UCT and Johannesburg. Thus, we use the minimum of the options minus the delay between the university and its source city. We use the minimum between each node because it is what the network is capable of in an ideal case.

SDN Model

We created the SDN model shown in Figure 3.2. Mininet does not allow multiple links between the same pair of nodes, thus, we combine any multiple links running between two nodes and add the capacities to get the total capacity between the two nodes. The link capacities in the figure is shown by colour according to the legend. Hosts for each university tested in this study are attached to the respective city that they reside in.

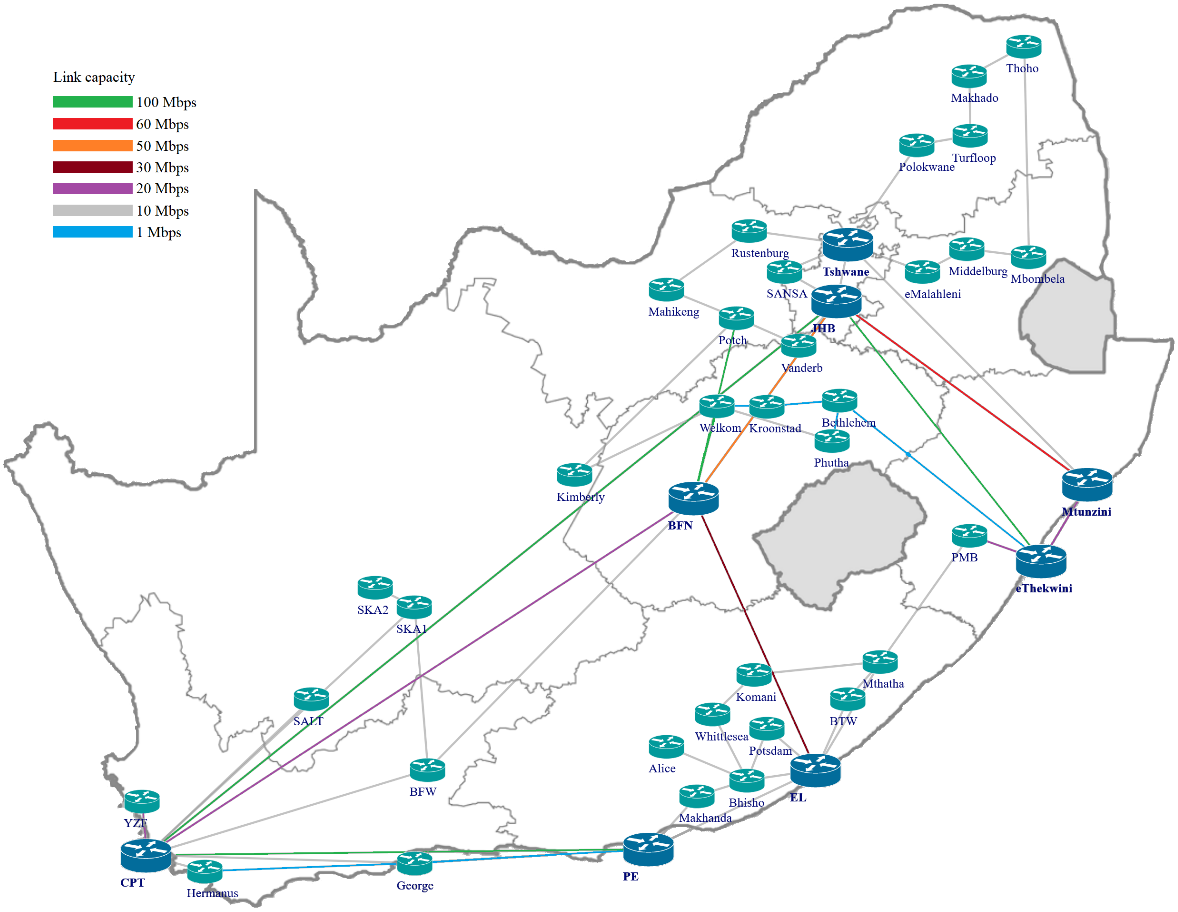


Figure 3.2: SDN model topology transposed onto a map of South Africa.

3.3.2 Virtual Network Performance Experiments

To evaluated performance in the SDN, we conduct the same tests as in the active measurements. Mininet allows for delay tests, page load time tests via *Wget*, throughput via *iperf*, as well as traceroute. However, we do not use the traceroute tests in this experiment due to our network consisting of the SANReN backbone only. From each host, we conduct the delay and throughput tests to the rest of the hosts and gather results from the tests using the same methodology outlined in 3.1.1. For page load time tests, we arrange for each host to be set up as a web server by using python's SimpleHTTPServer. We start the web server on the host, then use *Wget* in every host to download the file. We emulate lossless connection between all the nodes in the network and therefore do not measure packet loss.

Once we complete the performance tests on our SDN model, we compare the results to the results of the active measurements carried out on SANReN. Thereafter, we can tell if the SDN model would improve performance of the existing infrastructure.

Chapter 4

Results

In this section we present results from the tests that were conducted as well as the trends that were compiled from SANReN's netflow dataset. We present an analysis of the results from the active measurements experiments conducted. We then present our analysis of the netflow dataset focusing on usage in terms of volume and throughput. Lastly, we propose a potential solution to address the performance issues currently experienced by SANReN.

4.1 Active Measurements Performance Analysis

We present our active measurement results by experiment type and split them into external and internal results. We used Speedchecker to conduct experiments from outside of SANReN flowing into SANReN (external) and perfSONAR to conduct experiments from within SANReN flowing inside SANReN (internal). We present the data as box plots to visualise the results from the experiments and show the distribution of the data for each city and university. We used scatter plots to display the relationship between page load time and delay. We also show the number of IP hops taken by the traceroute to reach each destination from the source cities. After presenting the internal and external results, we present a comparison thereof.

4.1.1 Packet Delay Results

Packet Delay from Outside of SANReN

The results below show the delay when trying to access web servers from outside of SANReN. Figure 4.1 presents a box plot of the delay from the five cities that were used as sources for testing. From the figure, if we focus on overall results from each source city to all the universities, we see that the cities with the lowest delay are Johannesburg with a median packet delay of 29 ms, and Pretoria with a median delay of 35 ms. Durban had a median delay of 47 ms, Cape Town: 48 ms, and PE: 53 ms. In terms of the destination web server, the overall results from all source cities to targeted universities display similar results with WITS (residing in Johannesburg) having the lowest median delay: 37 ms, followed by UNISA (residing in Pretoria): 37.5 ms. NMU, which is located in PE, has the highest median delay with 57 ms, followed by UCT and UWC with 46 ms and 43 ms respectively. Both UCT and UWC are located in Cape Town.

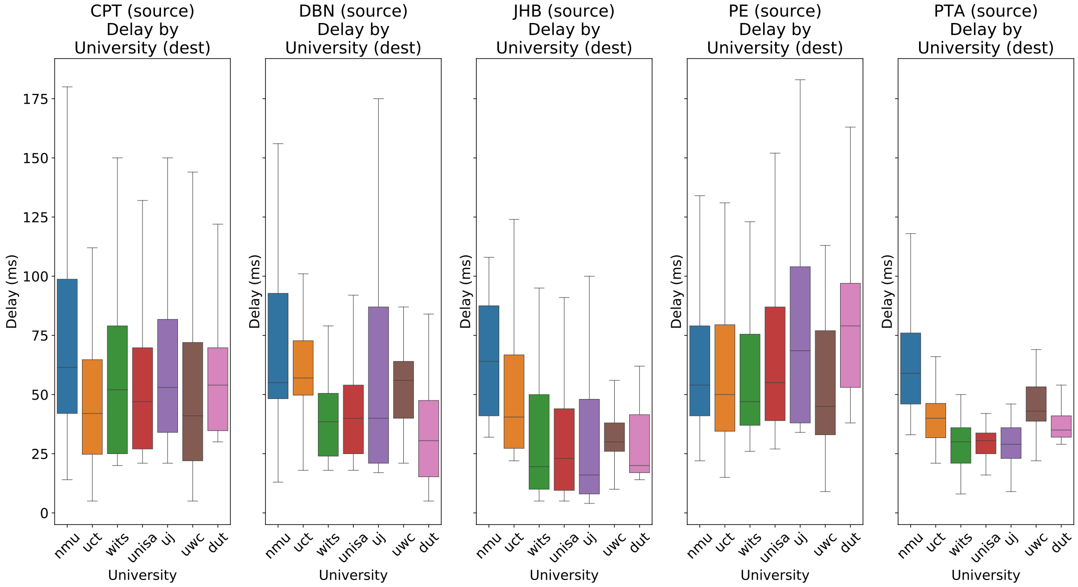


Figure 4.1: External delay by destination university.

We analysed the results in isolation focusing on specific source and target pairs. We found that the delay from Durban to Cape Town universities is higher than the delay from Durban to universities in Johannesburg, Pretoria, and Durban. The median delay from Durban to UWC is 56 ms and Durban to UCT is 57 ms. Delay between Durban and the universities in Johannesburg, Pretoria and Durban is lower, with Durban to UJ: 40 ms, Durban to UNISA: 40 ms, Durban to WITS: 38.5 ms, and Durban to DUT: 30.5 ms. We observe a similar pattern when looking at the delay between Cape Town universities and Johannesburg, as the median delay from Johannesburg to UWC is 30 ms and Johannesburg to UCT has a median delay of 40.5 ms. Johannesburg to UJ has the lowest median delay of any source and target pair (16 ms), followed by Johannesburg to WITS: 19.5 ms, Johannesburg to DUT: 20 ms, and Johannesburg to UNISA: 23 ms. By analysing the tests using Cape Town as the source city, we observe that the results are grouped closer together, resulting in a smaller difference in delay when targeting Johannesburg, Pretoria and Durban universities compared to targeting universities in Cape Town. We observed that the median delay from Cape Town to UCT and Cape Town to UWC is 42 ms and 41 ms respectively. From Cape town to UJ, the median delay is 53 ms, Cape Town to UNISA: 47 ms, Cape Town to WITS: 52 ms, and Cape Town to DUT: 54 ms. The delay between Cape Town and the cities more to the north-east of South Africa is high compared to the delays between cities that are closer to the north of the country.

Packet Delay from within SANReN

Results from the internal delay tests are presented below. From Figure 4.2, we see that the data is grouped close together in the results to all the target universities — the biggest interquartile range (IQR) existing between PE and Pretoria, with 2.93 from PE to UNISA and 2.9 from Pretoria to NMU. The small IQRs could be due to both the source and destination nodes being hosted inside SANReN. One of the main factors causing delays seems to be the distance between the source and target because universities that are far from the source city have higher delays. For example, the median delay when targeting DUT from Cape Town is 27.05 ms, while targeting UCT and UWC from Durban results in a median delay of 26.82 ms and 27.35 ms respectively. PE experiences its highest median delay when targeting UNISA (25.95 ms), and Pretoria experiences its highest median delay when targeting NMU (26.24 ms).

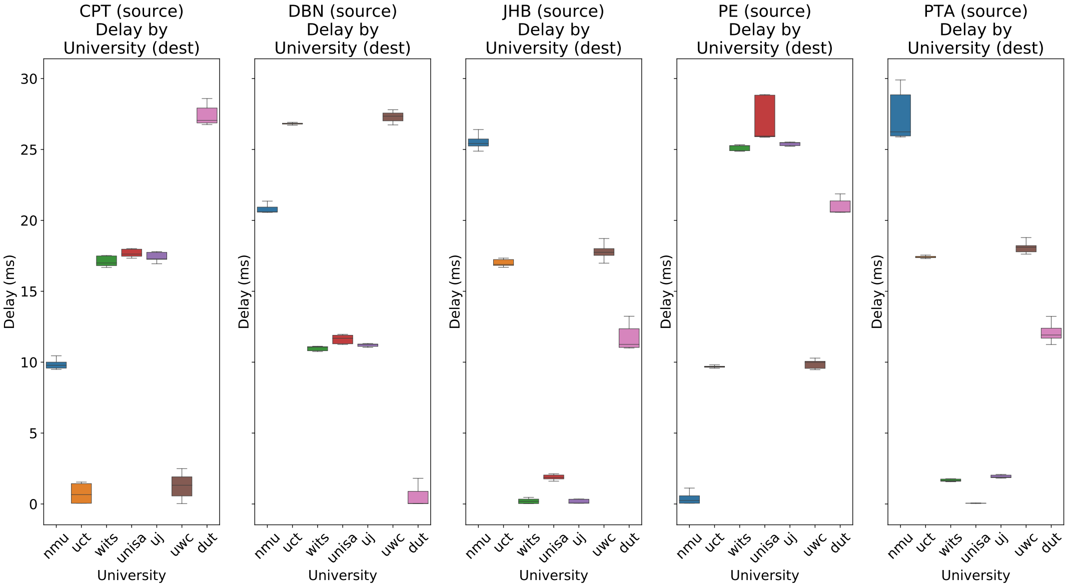


Figure 4.2: Internal delay by destination university.

When we use PE to target UJ and WITS, we observe high median delays of 25.47 ms and 25.26 ms respectively. In return, we observe that the university with the highest median delay from Johannesburg is NMU with 25.42 ms. Cape Town is the only city that reaches PE (and vice-versa) with low delays of around 10 ms, whereas Durban, Johannesburg and Pretoria experience median delays of 20.6 ms, 25.42 ms, and 26.24 ms respectively when targeting NMU. Delay from a source city to a target university located in the same city is very low, the highest median delay in this respect being Cape Town to UWC with 1.32 ms.

External vs Internal Packet Delay

We combined and compared the results from the external and internal delay experiments. Figure 4.3 shows the overall delay to each university and Table 4.1 shows of the median for each source and target pair with notable high and low values highlighted in bold. We observe that the internal tests have lower delays when targeting each of the universities. We also observe that NMU has the highest median delay from both external and internal sources, while WITS has the lowest median delay from both external and internal sources. UJ has the biggest IQR (53) and the highest value for the maximum delay from external sources (154 ms), which means that packets cannot regularly reach their destination timeously. The biggest difference in external and internal median is experienced by NMU with the difference being 36.4 ms (57 ms - 20.6 ms).

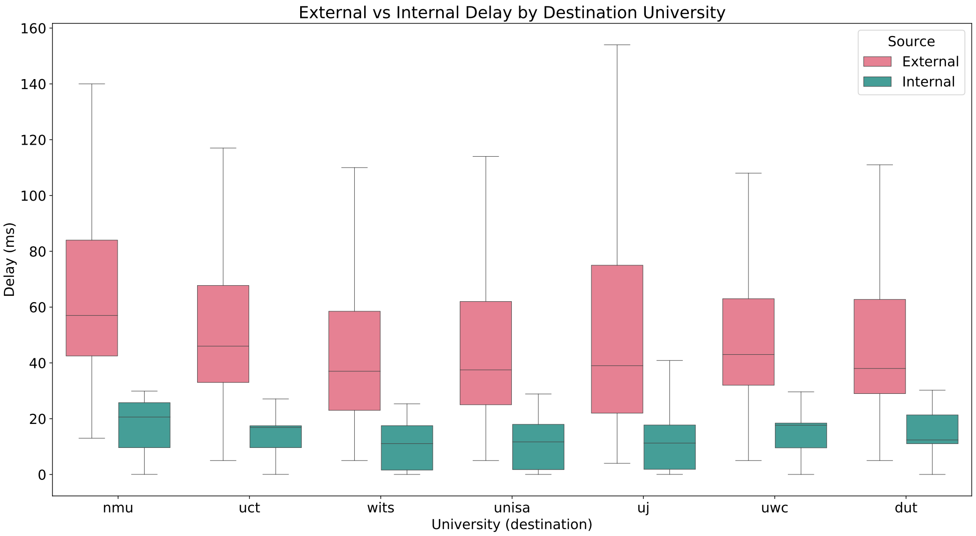


Figure 4.3: External vs internal delay by destination university.

Table 4.1: Summary of median external, internal, and overall packet delays (ms) for each source and target pair.

Source / Dest.	CPT		DBN		JHB		PE		PTA		Overall	
	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.
NMU	61.5	9.78	55	20.6	64	25.42	54	0.24	59	26.24	**57**	**20.6**
UCT	42	0.66	57	26.82	40.5	16.89	50	9.67	40	17.41	46	16.88
WITS	52	16.99	38.5	11.01	19.5	0.19	47	25.26	30	1.64	37	11.07
UNISA	47	17.62	40	11.69	23	1.86	55	25.95	30.5	0.06	37.5	11.69
UJ	53	17.28	40	11.26	16	0.01	68.5	25.47	29	1.89	39	11.26
UWC	41	1.32	56	27.35	30	17.75	45	9.98	43	18.11	43	17.66
DUT	54	**27.05**	30.5	0.03	20	11.25	79	20.59	35	11.92	38	12.38
Overall	48	16.9	47	11.7	29	11.01	**53**	**24.89**	35	7.15	42	**11.99**

4.1.2 Page Load Delay Results

Page Load Delay from outside of SANReN

Results from the page load time tests conducted by Speedchecker (external source nodes) to target web servers that reside inside SANReN are presented below. In Figure 4.4, we observe that results are similar across four of the different source cities, with PE being the outlier. PE experiences higher PLTs to load SANReN web servers than the rest of the cities with an overall median PLT of 1727 ms. Johannesburg has the lowest median PLT with 315 ms, followed by Durban with 472 ms, Cape Town with 496 ms, and Pretoria with 534 ms.

In terms of target web servers, NMU has the highest PLTs with an overall median of 899 ms across all the source cities. DUT, UWC, and UCT have median PLTs of 594.5 ms, 527 ms, and 516 ms respectively. WITS, UNISA, and UJ are the universities with the lowest overall median PLTs with 387 ms, 404 ms, and 426 ms respectively. When focusing on specific pairs of source city and target university, we see that although NMU resides in PE, the highest median PLT of any source and target pair is experienced when trying to reach NMU from PE, with 2112.91 ms. Moreover, PE has an overall IQR of 1244.5, showing that the PLTs are

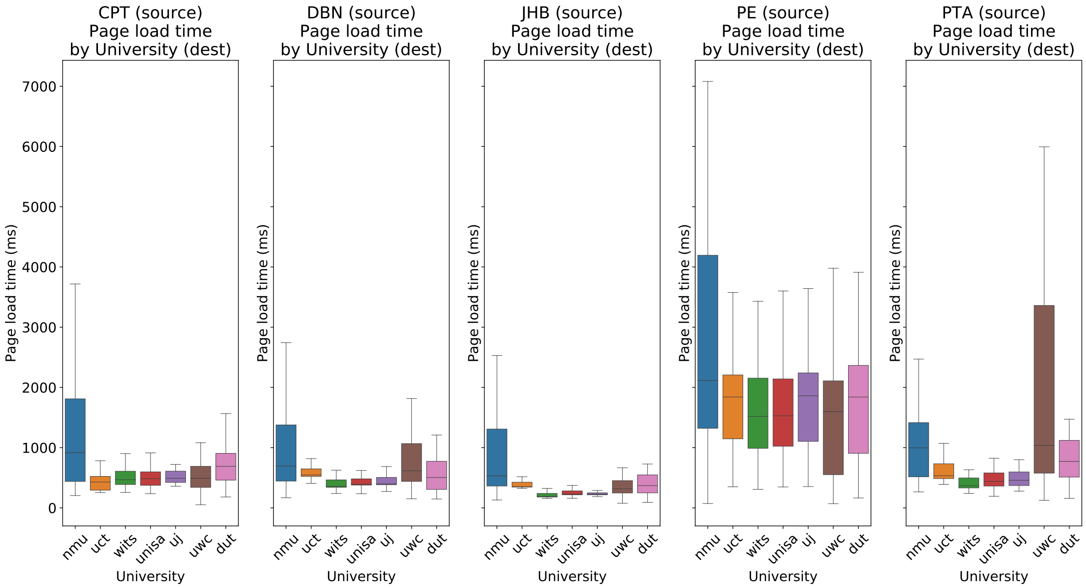


Figure 4.4: External page load times by destination university.

inconsistent to all the universities targeted. For comparison, Cape Town has an IQR of 322, Durban: 312, Johannesburg: 193, and Pretoria 526. In the case of Pretoria, the overall IQR is affected by the IQR when targeting UWC is 2853, showing that PLTs were inconsistent.

Page Load Delay from within SANReN

Figure 4.5 presents the results with respect to page load times from within SANReN. We see that NMU and DUT have the highest median PLTs across all the source cities with NMU having a PLT of 329.75 ms and DUT having a PLT of 341.82 ms. The rest of the universities experience lower PLTs, with UCT having an overall median PLT of 300.8 ms, UWC: 275.48 ms, UJ: 223.11 ms, UNISA: 215.68, and WITS: 179.55 ms. Similar to the delay results, targeting DUT from Cape Town poses an issue as the box plot to DUT from Cape Town has a median of 510.43 ms, while the next highest median PLT to DUT is 362.75 from PE. UWC has a median PLT of 275.48 ms, which is not as high as DUT and NMU but there is still a gap in PLT between UWC and the previously mentioned universities with lower PLTs

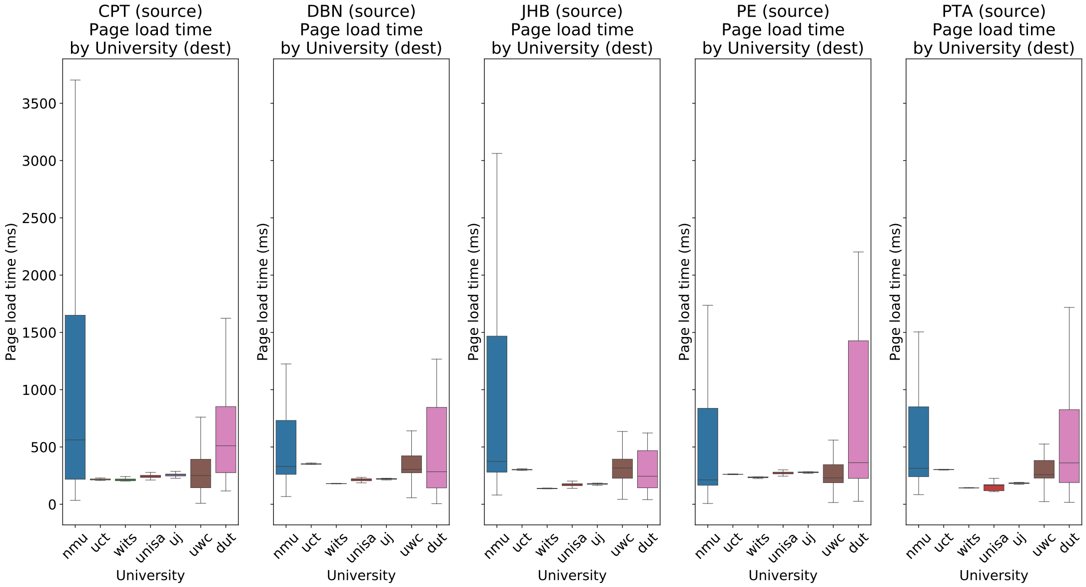


Figure 4.5: Internal page load time by destination university.

(WITS, UNISA, UJ). Although there is a small amount of variance in the PLTs targeting UCT, the median PLTs to UCT from Durban, Johannesburg, and Pretoria is still high with values of 350.32 ms, 301.88 ms, and 301.81 ms. However, when focusing on the tests using Cape Town as the source, the median PLTs are not as high with values of 256.71 ms to UJ, 240.77 ms to UNISA, and 216.72 ms to WITS.

External vs Internal Page Load Delay

We combine the results from our external and internal page load time experiments and present them in Figure 4.6. This shows a comparison between the overall page load times to each university from across South Africa. We observe that unlike the external and internal delay comparison, where there was no overlap, parts of the external and internal quartiles for NMU, DUT, and UWC overlapping. Between these universities and the cities in which they are located, we see high page load times, indicating congestion in the paths used to transfer data.

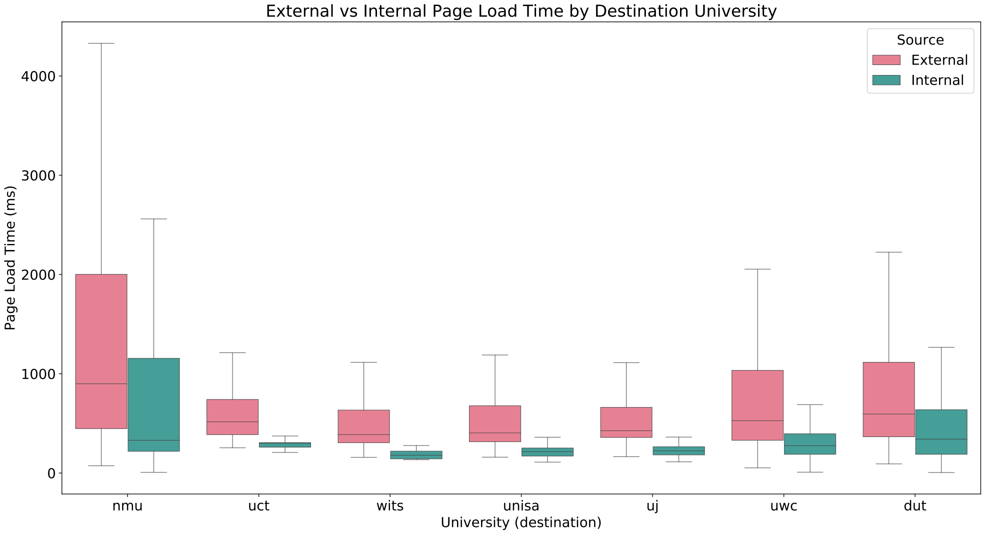


Figure 4.6: External vs internal delay by destination university.

More specifically, in Table 4.2, we see a difference of 569.25 ms when comparing the external and internal median PLTs to NMU, a difference of 252.68 ms when comparing DUT's external and internal PLTs, and a difference of 251.52 ms when comparing UWC. We observe a difference of 215.2 ms between UCT's median PLTs, 207.45 ms between WITS', 188.32 ms between UNISA's, and 202.89 ms between UJ's median PLTs. With the exception of UNISA, all the universities have a difference of over 200 ms between their external and internal PLTs.

Table 4.2: Summary of median external, internal, and overall page load times (ms) for each source and target pair.

Source / Dest.	CPT		DBN		JHB		PE		PTA		Overall	
	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.
NMU	916	561.49	694.5	329.75	531.5	374.75	**2113**	212.48	996	314.61	**899**	329.75
UCT	429	216.39	**546**	350.32	352	301.88	1841	261.63	534	301.81	516	300.8
WITS	467.5	216.72	357	179.51	190	137.71	1519	237.07	366.5	143.11	387	179.55
UNISA	484.5	240.77	392.5	215.63	231	173.55	1528	272.09	439	164.49	404	215.68
UJ	491	256.71	399.5	222.75	227	178.66	1860.5	280.47	458.5	183.31	426	223.11
UWC	491	250.73	**616**	305.22	316.5	316.56	1598.5	229.99	**1038**	258.65	527	275.48
DUT	**690.5**	510.43	**504**	283.69	370	244.75	1840	362.75	772	361	**594**	**341.8**
Overall	496	236.55	472	226.51	315	188.2	**1727**	262.93	534	189.55	491	237.5

4.1.3 Packet Delay vs Page Load Time

Packet Delay vs Page Load Time from outside of SANReN

After combining the results of the delay and PLT tests from all the source cities, we get the statistics shown in Table 4.3. We see that the median delay to WITS, UNISA, DUT, and UJ is 37 ms, 37.5 ms, 38 ms, and 39 ms respectively — these are the lowest median delays of the universities tested. Among these universities, WITS, UNISA, and UJ have similar median values for time-to-first-byte (TTFB) at 322.5 ms, 353 ms, and 371 ms respectively, whereas DUT has a median TTFB of 524.5 ms. This suggests that DUT processes HTML requests at a lower speed. NMU has the highest median delay (57 ms), PLT (899 ms), and TTFB (774 ms). The universities experience similar percentages of packet loss.

Table 4.3: External delay, page load time, and packet loss statistics by destination university.

University	Median delay (ms)	Median PLT (ms)	Median TTFB (ms)	Packet loss (%)
NMU	57	899	774	0.02
UCT	46	516	439.5	0.01
WITS	37	387	322.5	0.02
UNISA	37.5	404	353	0.02
UJ	39	426	371	0.02
UWC	43	527	464	0.03
DUT	38	594.5	524.5	0.02

From Figure 4.7 we see that the data is distributed along both axes. This means that we experience high delays as well as high PLTs. The average of the delay results (rounded to the nearest integer) is 50 ms and the average of the PLT results (rounded to the nearest integer) is 783 ms. We use the averages to inspect specific areas of the plot to find where data points are concentrated. We observe that 47.54% of the data points are less than or equal to both averages. Increasing the limit for delay to 150 ms (three times the average) reveals that 73.32% of the data points are found in this area of the plot (along the x-axis).

Increasing the limit for PLT to 2349 ms (three times the average), we see that 59.08% of the data points are concentrated in this area of the plot (along the y-axis). This means that the majority of the tests experienced high delays as opposed to high PLTs.

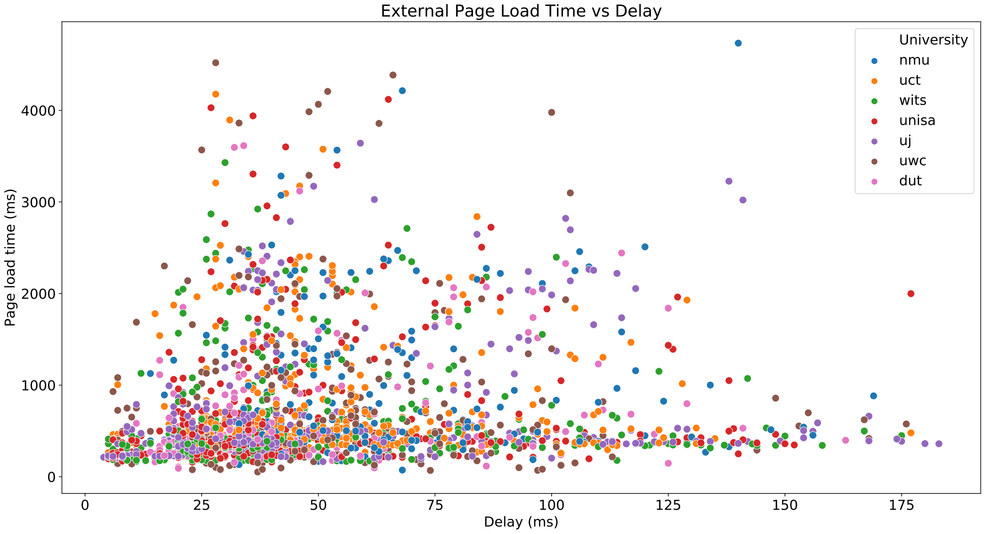


Figure 4.7: External page load times vs delay by destination university.

Packet Delay vs Page Load Time from within SANReN

Figure 4.8 shows the distribution of the combined results. The data points are grouped along the x-axis in the scatter plot in the same manner that they were grouped along the y-axis of Figure 4.2 (due to distance between the source and target). NMU is the only university with data points having PLTs higher than 1500 ms. A closer look at the data conveyed that these data points occur from Cape Town, Johannesburg, and PE. Overall, the groups of data are spread between universities and cities.

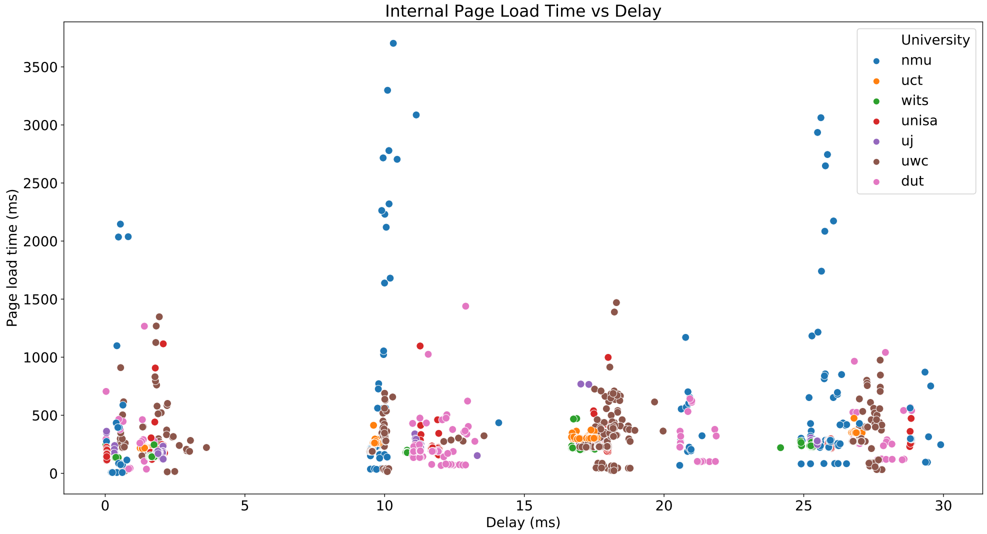


Figure 4.8: Internal page load times vs delay by destination university.

External and Internal Page Load Delay vs Packet Delay

We combined and compared the results from our external and internal page load time experiments. Figure 4.9 shows the overall results compared to one another. We notice that high delays are common when using external sources, with 20.53% of the data points having delays higher than 75 ms. We also observe that the page load times are similar for a large concentration of tests, with 77.42% of external tests having PLTs under 1000 ms and 93% of internal tests having PLTs under 1000 ms. This suggests that delay from external sources could be improved, and that throughput in the network is sufficient as page load times from inside the network are similar to that of external sources.

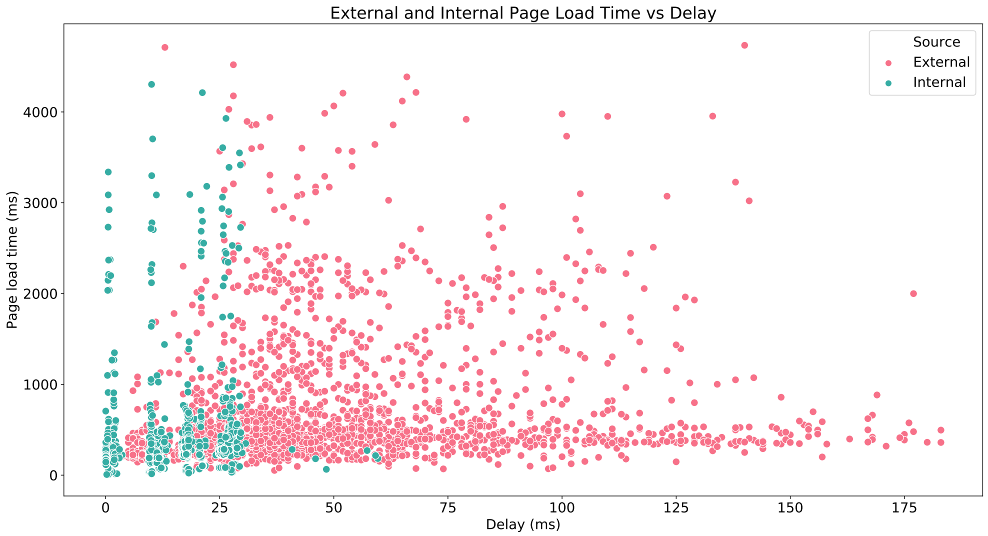


Figure 4.9: External and internal page load time vs delay.

4.1.4 Traceroute Results

Traceroute from Outside of SANReN

We conducted traceroute tests from the source cities to each target web server. Figure 4.10 shows the results of these tests and we make the following observations. When targeting DUT, we see a large variance from four of the source cities, with PE being the exception. When targeting DUT, we found an IQR of 9 from Cape Town and an IQR of 12 from Pretoria, Durban, and Johannesburg. NMU experiences some variance in the number of IP hops as well. We observe that the overall median number of hops to universities from PE is 14. The highest median among the target universities when using PE as the source is 17 from PE to NMU. PE is the only city where the university with the highest median number of hops is located in the source city itself. WITS has the lowest median number of hops from Durban, Johannesburg, and Pretoria. WITS also has the second lowest number of hops from Cape Town (11), with only UCT (residing in Cape Town) having a lower median (10).

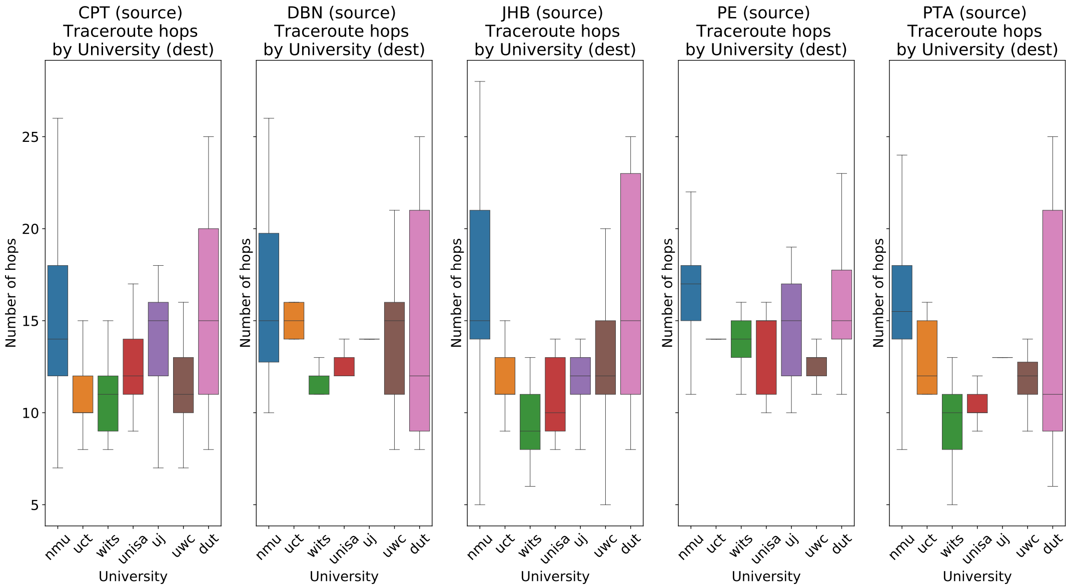


Figure 4.10: Number of IP hops for traceroute by destination university — external sources.

Table 4.4: External traceroute statistics by destination university.

University	Median no. of hops	Successful traceroute (%)	Unsuccessful traceroute (%)
NMU	15	85	8.1
UCT	12	77.86	19.05
WITS	11	80.24	14.76
UNISA	12	83.33	11.19
UJ	13.5	86.67	7.14
UWC	12	80.95	11.43
DUT	14	66.9	21.9

Table 4.4 shows the statistics of the results from the traceroute tests to each university. The percentages of successful and unsuccessful tests do not add up to 100 because the rest of the tests failed to initialise due to a lack of probe availability in Speedchecker. UJ is the university with the highest percentage of successful traceroutes with 86.67%. Although it has a median of 15 hops — the highest median number of hops across the universities — NMU has the second highest percentage of successful traceroutes with 85%. Traceroute tests to WITS, UNISA and UWC had success rates of over 80% as well. However, these universities

experience lower values for the median number of hops, with the results showing a median of 11 hops to reach WITS, and 12 to reach UNISA and UWC. A median of 12 hops were needed to reach UCT as well, but UCT has a higher percentage of failed tests (19.05%) and a lower percentage of successful tests (77.86%). DUT has a high median of 14 hops, the lowest success rate (66.9%) and the highest failure rate (21.43%).

Table 4.5: External traceroute statistics by source city.

City	Median no. of hops	Successful traceroute (%)	Unsuccessful traceroute (%)
CPT	12	87.93	11.9
DBN	14	87.24	9.86
JHB	12	80.95	5.78
PE	14	81.46	11.56
PTA	12	63.1	27.72

Analysing the results from the perspective of each source city, shown in Table 4.5, we see that Pretoria has only 63.1% successful traceroutes, and 27.72% unsuccessful. From PE and Durban, there is a median of 14 hops, while Pretoria, Cape Town and Johannesburg take a median of 12 hops. Cape Town and Durban have the highest percentages of successful tests with 87.93% and 87.24% respectively. PE and Johannesburg have success rates of 81.46% and 80.95% respectively.

Traceroute from within SANReN

Figure 4.11 shows the results of the number of hops from the traceroute tests conducted. We see that there is only one hop from a source city to a university hosted in the same city because these tests were conducted from the perfSONAR node to itself. UWC and UJ are the exceptions in this case regard as the Cape Town tests were done from UCT and the Johannesburg tests were done from WITS to get consistent representative results for these cities.

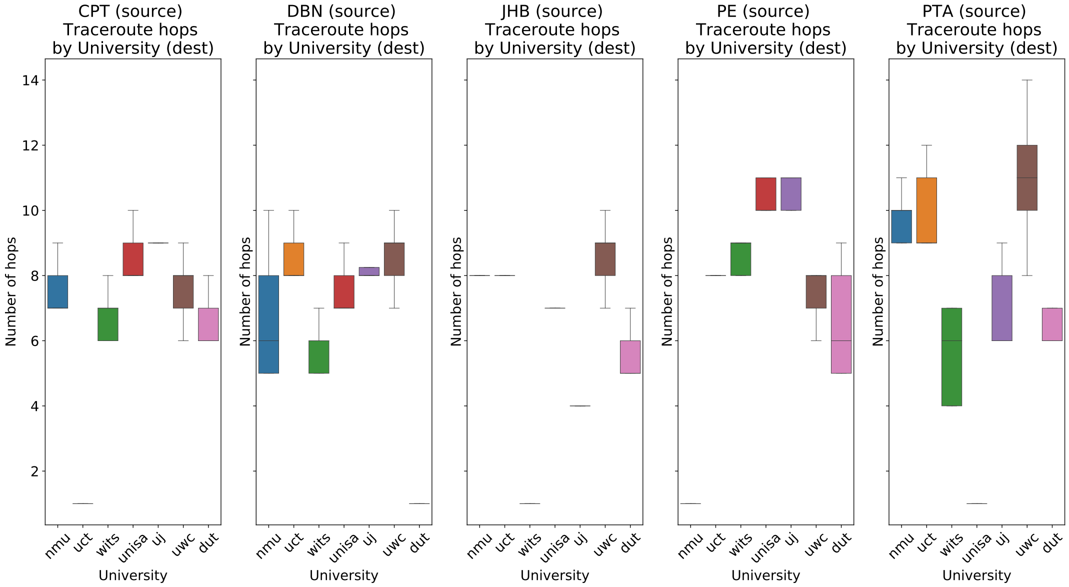


Figure 4.11: Number of IP hops for traceroute by destination university — internal sources.

From Johannesburg, UWC and DUT are the only universities that have an IQR of 1. We see that there is no variance from Johannesburg to the rest of the universities targeted. From Pretoria, we see the highest median number of hops to UWC with 11, followed by NMU with 10, and UCT with 9. From PE, we also see a high median number of hops to UNISA and UJ with 10 and 11 respectively. We see no variance in the results from PE to UCT, and from the data, we see that packets traversed the same path to get to UCT. The biggest variance is experienced between Durban and PE with an IQR of 3 from PE to DUT and Durban to NMU. We also observe an IQR of 3 from Pretoria to WITS.

We see statistics of the traceroute tests per destination university in Table 4.6. Besides UWC and DUT, the results have high success rates, with no failures, besides tests that failed to initialise. The failed traceroutes to UWC were caused by the university's security policies. With respect to the failed traceroutes to DUT, these failed due to not being able to reach the destination. Packets were sent from within SANReN in South Africa to SEACOM in Mauritius and could not reach the destination thereafter.

Table 4.6: Internal traceroute statistics by destination university.

University	Median no. of hops (Successful)	Successful traceroute (%)	Unsuccessful traceroute (%)
NMU	8	95.95	0
UCT	8	98.33	0
WITS	6	99.52	0
UNISA	7	99.05	0
UJ	8	100	0
UWC	8	93.33	2.38
DUT	6	80	7.14

From the statistics by source city seen in Table 4.7, we observe that the cities have similar results in terms of success rates and median number of hops. Port Elizabeth and Durban have 8 as the median number of hops, one higher than the rest of the cities. PE also has a slightly lower success rate of 92.18% compared to the rest of the cities. Johannesburg, Durban, and Pretoria have success rates of 95.75%, 95.24%, and 95.24% respectively, while Cape Town has the highest success rate and the lowest failure rate of 97.45% and 2.38% respectively.

Table 4.7: Internal traceroute statistics by source city.

City	Median no. of hops (Successful)	Successful traceroute (%)	Unsuccessful traceroute (%)
CPT	7	97.45	0
DBN	8	95.24	0.17
JHB	7	95.75	0.51
PE	8	92.18	0.68
PTA	7	95.24	0.34

External vs Internal Traceroute

Figure 4.12 presents a comparison of the overall number of hops to each university. We observe that from our external experiments, DUT has a high amount of variance in the results with an IQR of 10 hops. This is in contrast to the internal tests to DUT which has an IQR of one. Moreover, DUT is the university with the biggest difference in external and internal median values, with external tests taking 14 hops and internal tests taking six hops. NMU's internal and external results have a big difference in their median values as well, with the internal and external tests taking 8 and 15 hops respectively.

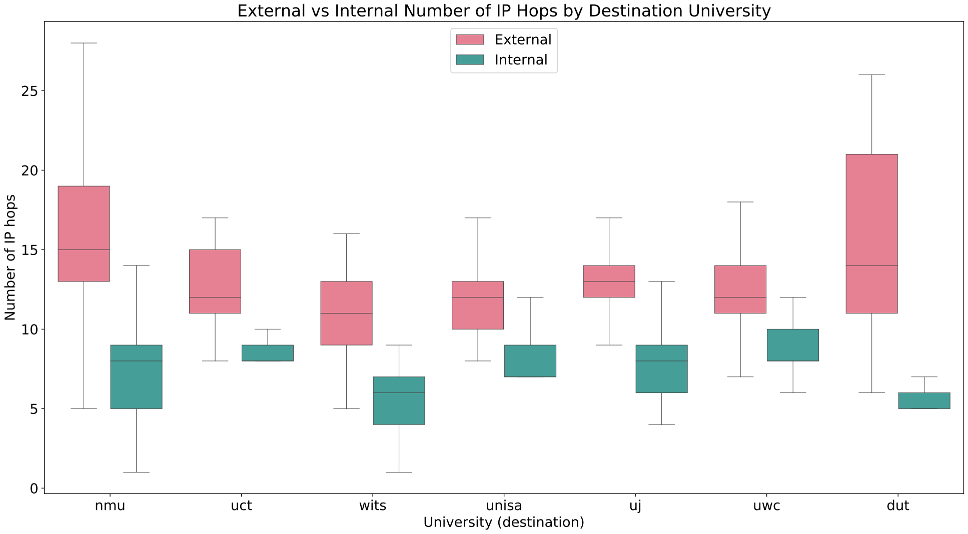


Figure 4.12: Number of IP hops for traceroute by destination university — external vs internal sources.

From the summary below (Table 4.8), we observe that, for the external tests, the highest number of hops of any source and target pair is 17 from PE to NMU. For the internal tests, the highest number of hops for a source and target pair is 11 from PE to UNISA and from Pretoria to UWC.

Table 4.8: Summary of median external, internal, and overall number of hops for traceroute for each source and target pair.

Source / Dest.	CPT		DBN		JHB		PE		PTA		Overall	
	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.
NMU	14	7	15	6	15	8	17	1	15.5	10	15	8
UCT	10	1	15	8	11	8	14	8	12	9	12	8
WITS	11	7	12	6	9	1	14	9	10	6	11	6
UNISA	12	8	12	7	10	7	15	10	10	1	12	7
UJ	15	9	14	8	12	4	15	11	13	6	13	8
UWC	11	8	15	9	12	9	13	8	12	11	12	8
DUT	15	6	12	1	15	5	15	6	11	6	14	6
Overall	12	7	14	8	12	7	14	8	12	7	13	8

4.1.5 Throughput Results

Throughput from Outside of SANReN

We used the total downloaded bytes divided by the difference of PLT and TTFB of each test to calculate the throughput ($Throughput = \frac{TDB}{PLT-TTFB}$). The results are plotted in Gbps on Figure 4.13. We observe that Cape Town, Durban, and Pretoria experience similar median throughput values (as seen in Table 4.9), while Johannesburg has the highest overall median throughput with a value of 0.01404. The highest value to a specific university is 0.02206 Gbps from Johannesburg to DUT. PE experiences the lowest throughput with an overall median value of 0.00234, where PE to UJ has the lowest throughput to a specific university (0.00104 Gbps). From sources other than PE, the lowest throughput (0.0032 Gbps) is observed from Durban to UCT and Pretoria to UCT. DUT and UWC have high IQRs from all source cities — the highest being from Johannesburg to UWC and DUT (0.08039 and 0.06366 respectively).

In terms of throughput to each target university, we found that UCT and WITS have the lowest values for median throughput with 0.00372 Gbps and 0.0048 Gbps respectively. The

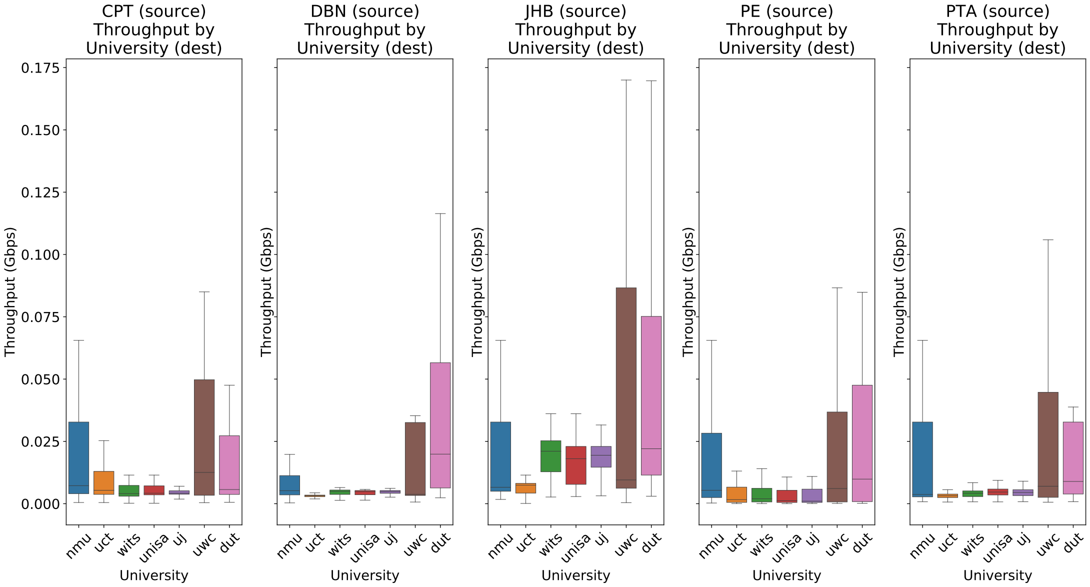


Figure 4.13: External throughput by destination university.

Table 4.9: Total downloaded bytes, time-to-first-byte, and throughput for external page load time tests by source city.

City	Median TDB (bytes)	Median TTFB (ms)	Median throughput (Gbps)
CPT	31594	420	0.0046
DBN	31594	390	0.00477
JHB	31594	274	0.01404
PE	31594	1287	0.00234
PTA	31594	448	0.00421

universities that have the highest median throughput values are DUT and UWC achieving 0.01204 Gbps and 0.00872 Gbps respectively.

Throughput from within SANReN

We conducted throughput experiments from within SANReN to the universities and present the results in Figure 4.14. These tests were conducted using perfSONAR nodes as the source and target because the tests to zero-rated web servers failed (perfSONAR only allowed tests to other perfSONAR nodes). Thus, Figure 4.14 only shows values for throughput from the source city to universities residing in the same city for Johannesburg and Cape Town (from WITS to UJ and from UCT to UWC). The results show that Cape Town experiences lower throughput with an overall median of 5.47 Gbps, compared to the rest of the cities that experience median throughput values between 7.5 Gbps and 8.005 Gbps. From Cape Town to UNISA and UJ, we experience high variances with IQRs of 1.27 and 1.25 respectively. The lowest throughput from source city to target university is 4.91 Gbps from Durban to UCT. Despite this, Durban experiences the highest overall median throughput of 8.005 Gbps, followed by Pretoria with 7.64 Gbps, Johannesburg with 7.55 Gbps, and PE with 7.53 Gbps.

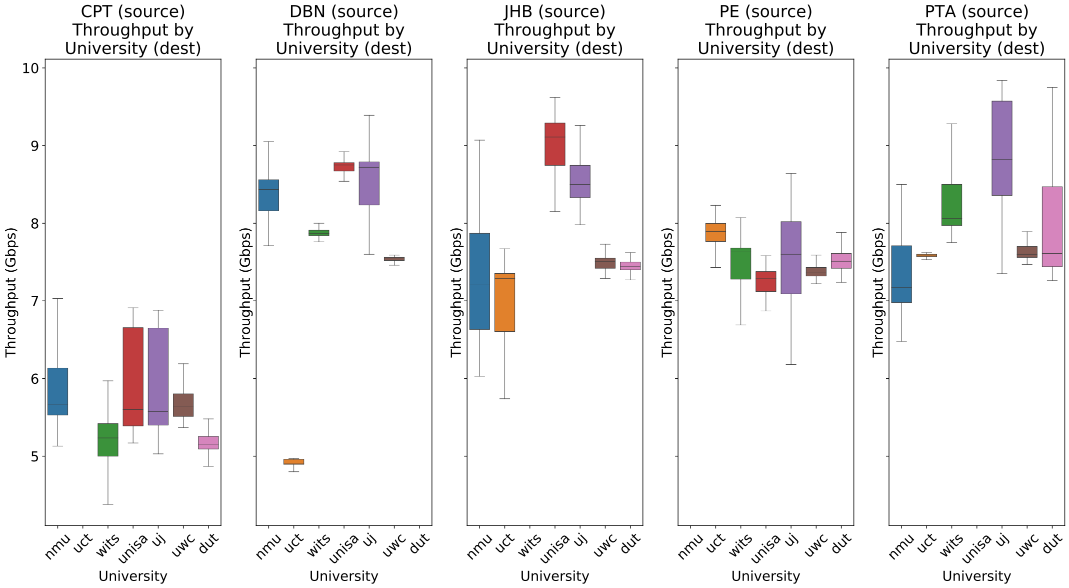


Figure 4.14: Internal throughput by destination university.

External vs Internal Throughput

We experience much higher values for throughput compared to the external tests. The external results have values below 0.1 Gbps because they could not be obtained using the same method as the internal tests. Figure 4.15 visualises the external and internal tests, where A shows the comparison between the sets of measurements and B shows the external results magnified. We observe different trends between the external and internal throughput results. We compare the trends for these tests as a direct comparison is not possible. For the internal results, UJ (8.26 Gbps), UNISA (8.065 Gbps), and WITS (7.79 Gbps) have the median highest throughput values. Whereas for the external tests, DUT (0.012036 Gbps) has the highest median throughput.

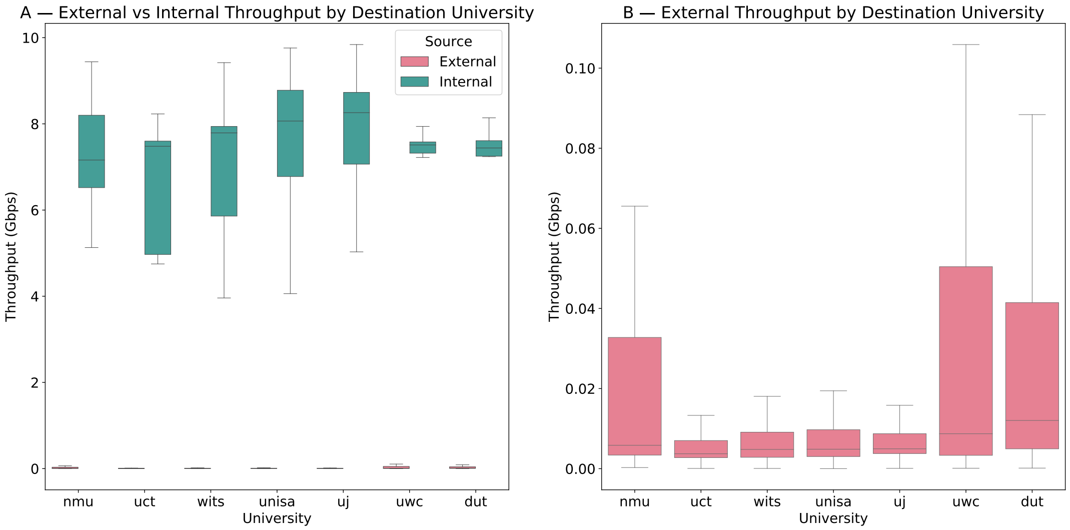


Figure 4.15: External vs internal throughput by destination university.

Table 4.10: Summary of median external and overall throughput (Mbps) for each source and target pair.

Source / Dest.	CPT	DBN	JHB	PE	PTA	Overall
NMU	7.243	5.232	6.6	5.38	3.694	5.834
UCT	5.378	3.199	7.434	1.631	3.199	3.717
WITS	4.047	4.956	21.063	1.968	4.213	4.803
UNISA	4.218	4.861	18.054	1.198	4.681	4.861
UJ	4.215	4.861	19.442	1.031	4.513	4.956
UWC	12.551	3.772	9.541	6.091	7.016	8.716
DUT	5.682	19.89	22.058	9.868	8.928	12.036
Overall	4.595	4.769	14.042	2.34	4.213	5.158

Table 4.11: Summary of median internal and overall throughput (Gbps) for each source and target pair.

Source / Dest.	CPT	DBN	JHB	PE	PTA	Overall
NMU	5.67	8.435	7.205		7.17	7.16
UCT		4.91	7.29	7.895	7.58	7.48
WITS	5.235	7.87		7.63	8.06	7.79
UNISA	5.6	8.75	9.11	7.285		8.065
UJ	5.575	8.72	8.5	7.6	8.82	8.26
UWC	5.645	7.54	7.505	7.36	7.6	7.51
DUT	5.155		7.44	7.51	7.61	7.44
Overall	5.47	8.005	7.55	7.53	7.64	7.55

4.2 SANReN UCT Traffic Flow Analysis

In this section, we present the passive measurements analysis carried out on the netflow dataset of UCT's traffic through SANReN. The network trends are shown by breaking down the dataset by traffic per week, inbound and outbound flows, South African traffic, country, ISP, and protocol. We show the throughput and volume in each breakdown by using line graphs for the volume and box plots for the throughput.

4.2.1 Volume and Throughput — Global Traffic

In Figure 4.16 and Figure 4.17 we see the volume and throughput trends of all the flows to and from UCT over a two-month period. The plot for all the traffic (Figure 4.16) shows that the median throughput of the traffic for every week is under 0.00012 Gbps, with the boxes being positively skewed. The volumes (left of Figure 4.16) experienced during the first two weeks we investigated were show 8945.84 GB for week 1 and 9034.23 week 2. We see a spike in the volume of traffic in week 3 (15-21 July 2020). In Figure 4.17 we see that majority of the data that was transferred in week 3 were into UCT. We investigated the spike in volume and found that the flows causing the spike were from the United States (seen in Figure 4.20), since the ISP with the most data transferred during this period was Lightower Fiber Networks [1] (looked up with MaxMind, IP-API, and IP-WHOIS). UCT requested 4408.96 GB from Lightower during week 3, while only 1960.88 GB was requested from TENET. Since the source and target IPs are anonymised, we cannot see what the traffic is that caused the spike.

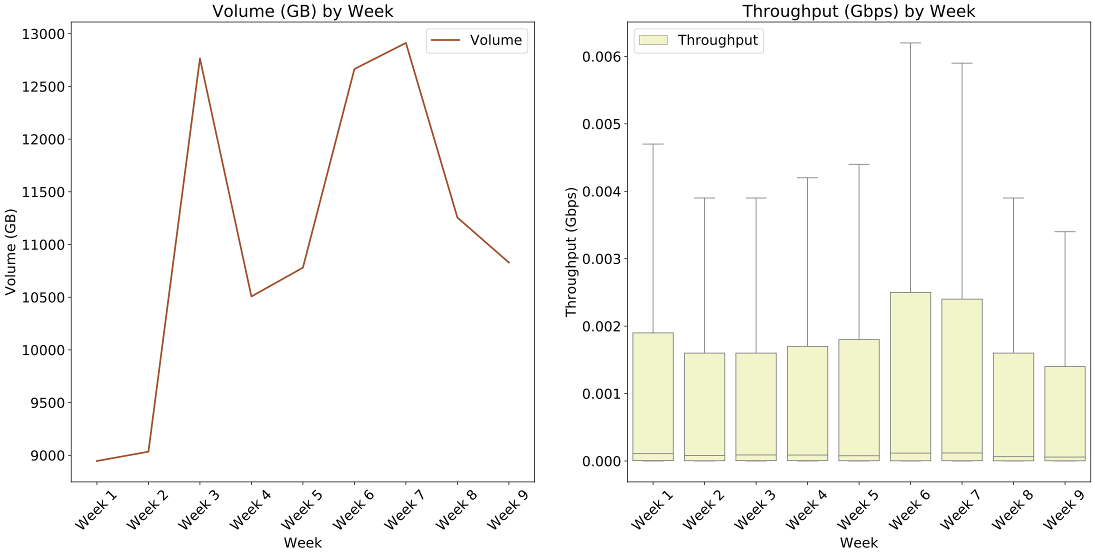


Figure 4.16: Volume (GB) and throughput (Gbps) by week.

The breakdown of throughput between inbound and outbound traffic (Figure 4.17) shows that the throughput for traffic flowing into UCT is significantly better than traffic sent out from UCT. The median throughput for each week is shown in Table 4.12. We see from Figure 4.17 and Table 4.12 that the traffic flowing out of UCT (outbound) experiences lower throughput values than that of the traffic flowing into UCT (inbound). We see that the median inbound throughput for all the weeks is 0.000759 Gbps and the median for outbound traffic for all the weeks is 0.000068 Gbps, which means that inbound traffic is 11.16

---

[1] https://www.crunchbase.com/organization/lightower-fiber-networks

times faster than outbound traffic overall. We observe the biggest difference in throughput performance was during week 4, when the inbound traffic was 25.93 times faster. In week 9, we observe the smallest difference in throughput performance, with inbound traffic being 1.11 times faster. We see an increase in the volume of traffic starting from week 4 (22 July 2020). This increase reflects students being allowed to return to campus and residences during the Covid-19 pandemic [50]. We see a further increase in the volume of traffic during weeks 5 - 7, which coincides with the start of UCT's third quarter during the Covid-19 pandemic in 2020 [51].

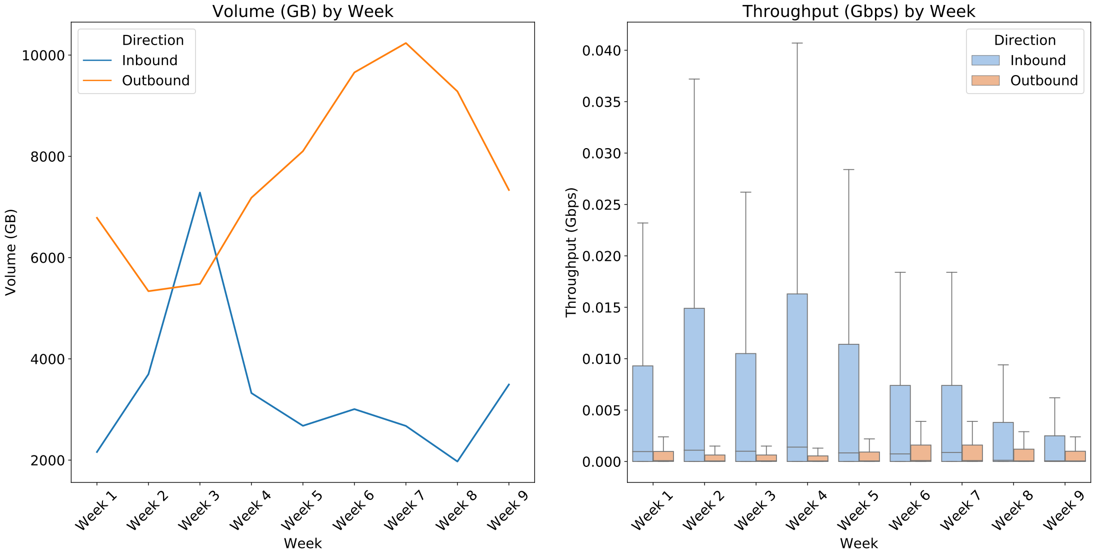


Figure 4.17: Volume (GB) and throughput (Gbps) by week — inbound and outbound.

Table 4.12 below shows the breakdown volume and throughput for global traffic to and from UCT. The values referred to in the text are highlighted in bold. The inbound and outbound values for throughput do not add up to the overall values because they are all median values. Moreover, the median values for overall throughput is closer to the median outbound throughput values due to the higher volume of outbound traffic.

Table 4.12: Global traffic statistics to and from UCT for volume and throughput.

Period	Volume (GB)			Median throughput (Gbps)		
	Inbound	Outbound	**Overall**	Inbound	Outbound	**Overall**
Week 1	2160.24	6785.61	8945.85	0.000965	0.000084	0.000109
Week 2	3697.52	5336.71	9034.23	0.0011	0.000056	0.000081
Week 3	**7285.2**	5480.08	12765.28	0.001	0.000064	0.00009
Week 4	3324.33	7182.59	10506.92	**0.0014**	**0.000054**	0.000088
Week 5	2678.11	8102.28	10780.39	0.000837	0.000059	0.000078
Week 6	3007.4	**9657.55**	12664.95	0.00074	0.000094	0.000117
Week 7	2675.22	**10237.5**	12912.72	0.000876	0.000091	0.000119
Week 8	1972.68	**9283.1**	11255.78	0.000122	0.000061	0.000064
Week 9	3492.94	7335.53	10828.47	0.000063	**0.000057**	**0.000058**
Overall	30293.64	69400.95	99694.59	**0.000759**	**0.000068**	0.000087

4.2.2 Volume and Throughput — South African Traffic

We isolated South African traffic from the dataset and present the results in Figure 4.18. The results show that the median throughput is 0.622 Mbps, which is 7.15 times faster than the median throughput of global traffic to and from UCT (0.087 Mbps). Moreover, the upper quartile for South African traffic is 5.8 Mbps whereas the upper quartile of throughput for global traffic is 1.8 Mbps. Therefore, the throughput for South African traffic is 3.22 times more than the global traffic. With the exception of week three, the volume trend line for South African traffic is similar to the trend line for global traffic, with the peak in week 7 being apparent as well. The total volume of South African traffic transferred during the two-month period we studied is 82534.65 GB. Since the total volume of traffic transferred during the two-month period is 99694.59 GB, we note that 82.79% of UCT's traffic either flows to South African destinations or comes from South African sources.

When we look at the breakdown for inbound and outbound traffic (4.19), we see that the volume of outbound data is high from Week 6 to Week 8, coinciding with the start of UCT's second semester in 2020. During this period, the ISP which UCT sent the most data to

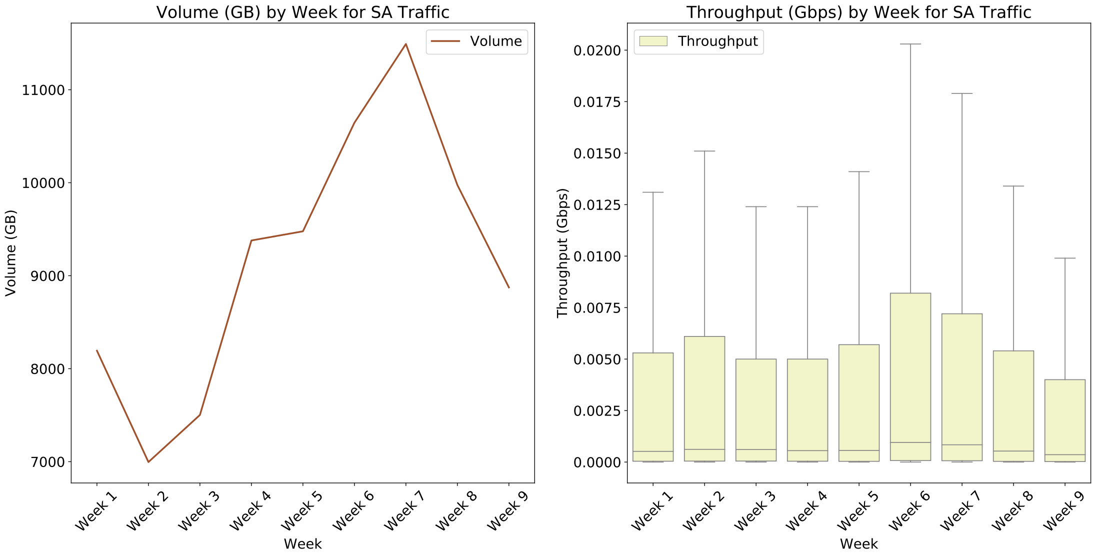


Figure 4.18: Volume (GB) and throughput (Gbps) by Week SA Traffic.

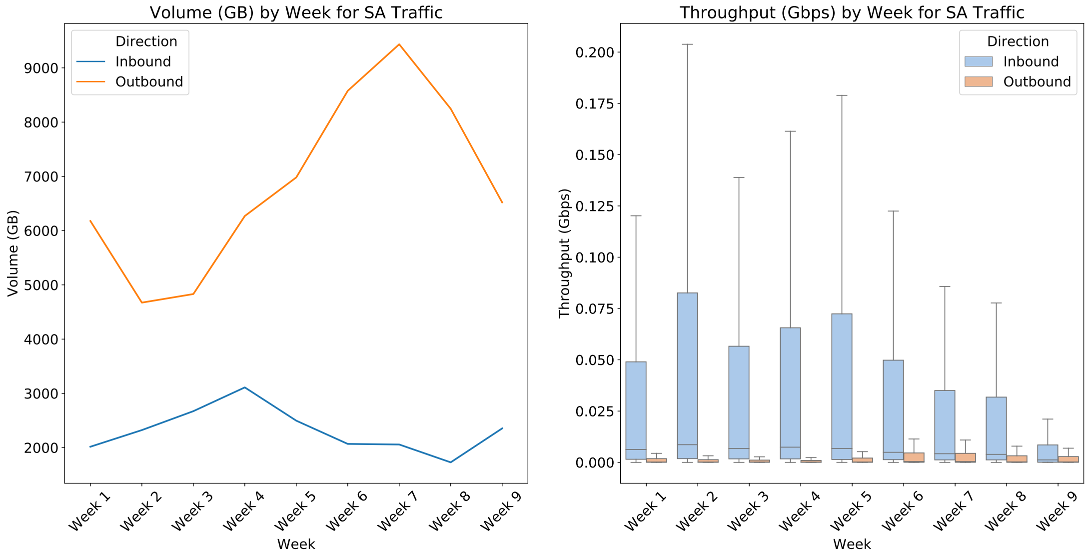


Figure 4.19: Volume (GB) and throughput (Gbps) by Week for SA Traffic — Inbound and Outbound.

was Telkom (AS 37457) with 7779.24 GB of data, which accounts for 24.23% of the total data flowing out of UCT to South African destinations during these weeks. We see that the performance of inbound traffic once again surpasses that of the outbound traffic, with data being transferred out of UCT at a median of 0.000199 Gbps and data flowing into UCT at a median of 0.0051 Gbps (25.63 times faster).

4.2.3 Countries with the Highest Volume of Traffic

In Figure 4.20, we see the trends of volume for the ten countries that transferred the most data to and from UCT during the period studied. This is also shown in Table 4.13 alongside the figures for median throughput. Due to the difference in volume per country, we broke down the figure into four plots to visualise the countries with lower volumes. We observe that South Africa is the country with the highest amount of usage, followed by the United States, and Australia. We see that the United States is the country with the second highest volume during every week studied and experiences a peak in week three from the US ISP Lightower as mentioned previously. Australia has the third highest volume, due to a peak in the 5th week caused by fetching data from the Australian Academic and Research Network (AARNet) totalling 378.46 GB. A few countries have an increase in volume in the last week of the period investigated — Ireland, South Korea, Canada, and Australia.

Figure 4.21 shows the throughput of the countries with the highest volumes, where Figure 4.21 *B* excludes South Africa and Namibia. We observe that besides South Africa, Namibia has the highest median throughput of 0.000224 Gbps. The United States has a median throughput of 0.000013 Gbps, Ireland 0.000006 Gbps, and 0.000004 Gbps for Australia, Canada, and the Netherlands. China, United Kingdom, and South Korea have lower median throughput values, with 000000295 Gbps.

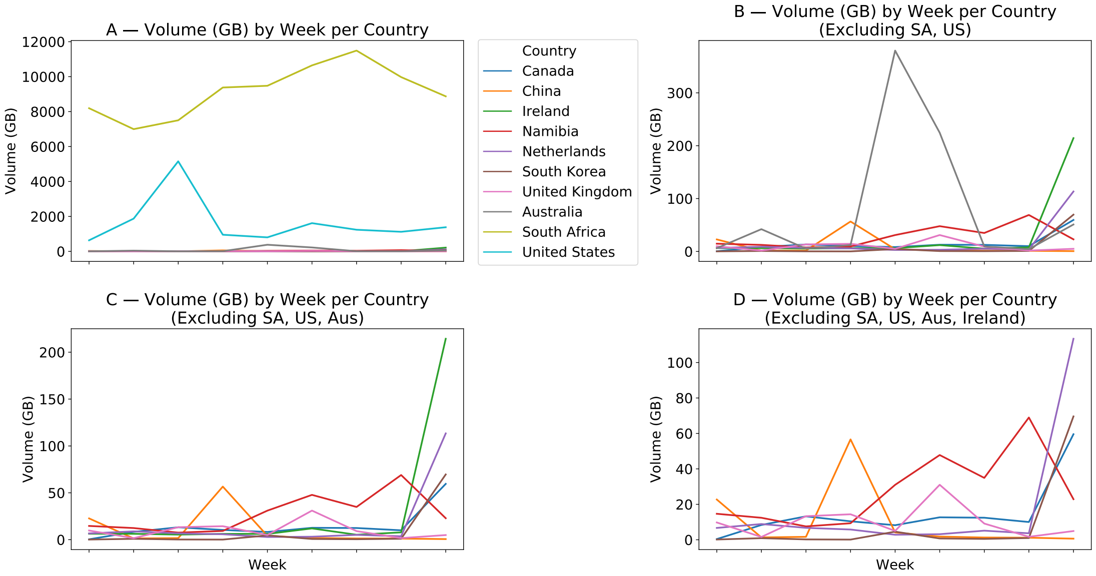


Figure 4.20: Volume (GB) by Week per Country.

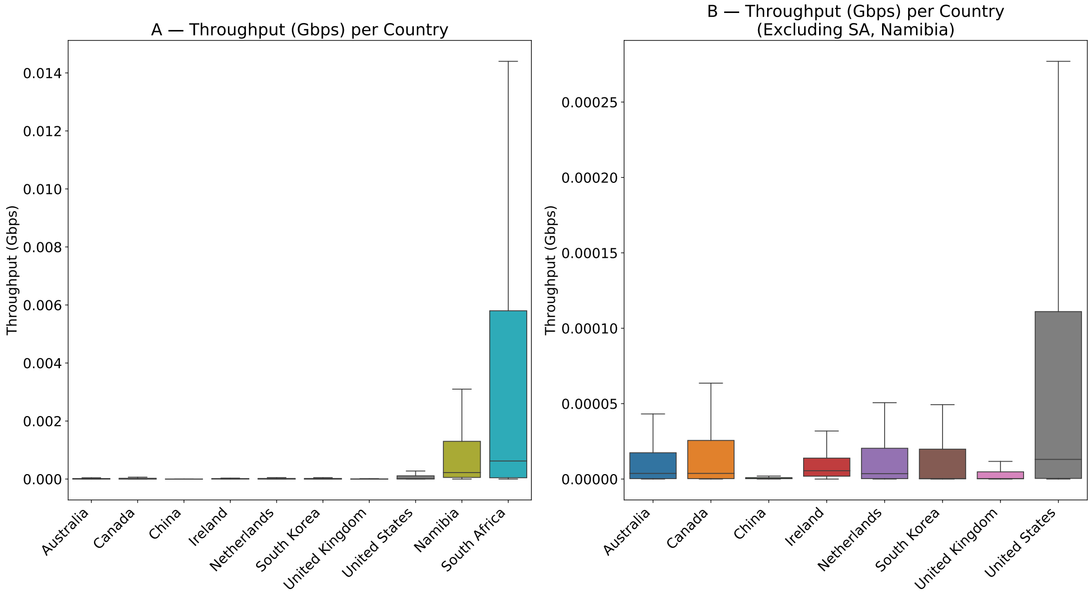


Figure 4.21: Throughput (Gbps) per Country.

Table 4.13: Volume (GB) and median throughput (Gbps) for high volume countries.

Country	Volume (GB)	Median throughput (Gbps)
South Africa	82534.66	0.000622
United States	14777.91	0.000013
Australia	734.97	0.000004
Ireland	269.81	0.000006
Namibia	249.53	0.000224
Netherlands	156.26	0.000004
Canada	135.29	0.000004
China	91.38	0.000000295
United Kingdom	90.83	0.000000295
South Korea	77.78	0.000000295

4.2.4 Volume and Throughput by ISP

The ten ISPs with the highest volumes of traffic are shown in Figure 4.22. We observe that Telkom and TENET have the highest volumes — 17479.11 GB and 14505.68 GB respectively. As expected, TENET has the highest median throughput of 0.0027 Gbps, followed by Internet Solutions (IS) and Telkom (both 0.0021 Gbps). These ten ISPs account for 75.56% of the traffic transferred over the nine-week period. The only two ISPs that are hosted outside of SA are Lightower and Microsoft Corporation. Thus, majority of the network traffic flows in and out of UCT via SA ISPs, where these eight ISPs account for 64.35% of the total traffic.

Figure 4.23 shows the throughput of each ISP, with the right plot focusing on the ISPs with lower throughput values. We observe that despite the spike in volume during week three, Lightower is the ISP with the lowest median throughput. We observe that the throughput experienced by TENET is similar to IS and Telkom, even though TENET hosts UCT. Table 4.14 shows a summary of the volume and throughput for these ten ISPs.

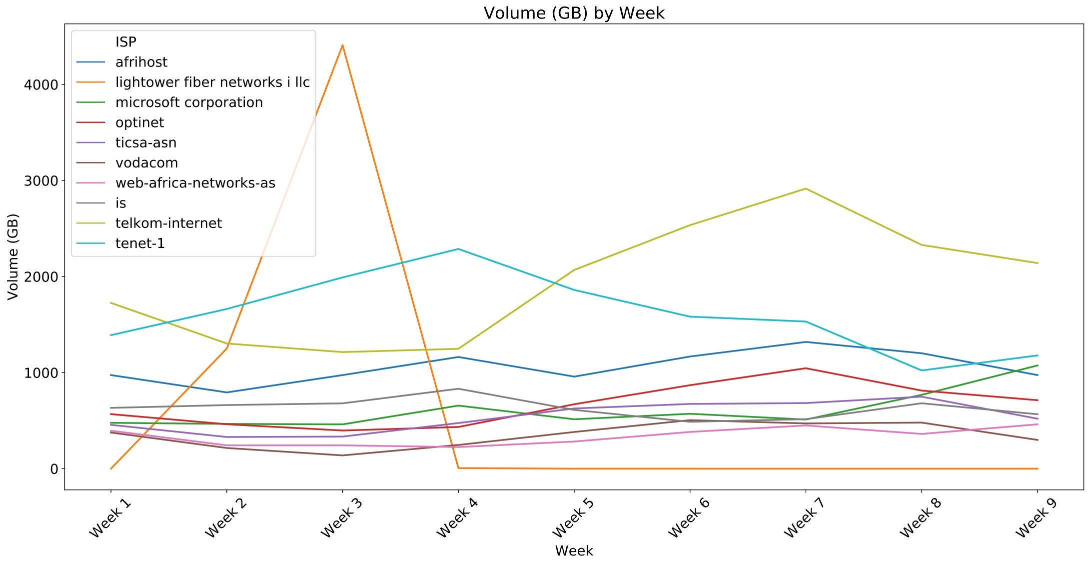


Figure 4.22: Volume (GB) by Week per ISP.

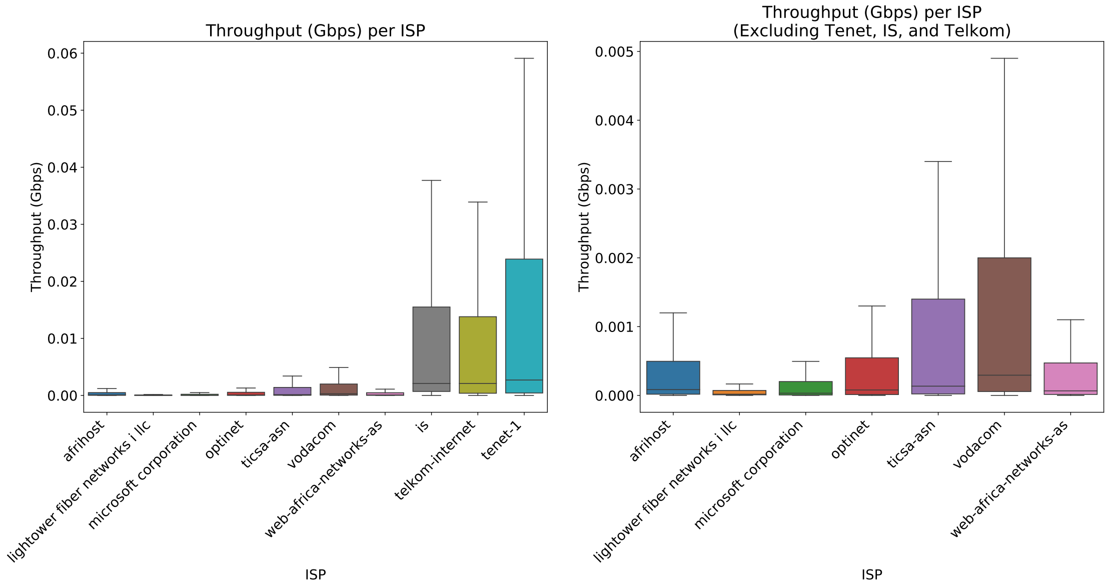


Figure 4.23: Throughput (Gbps) by ISP.

Table 4.14: Volume (GB) and median throughput (Gbps) for high volume ISPs.

Country	Volume (GB)	Median throughput (Gbps)
Telkom-Internet	17479.11	0.0021
TENET-1	14505.68	0.0027
Afrihost	9525.33	0.000085
OPTINET	5971.93	0.000079
IS	5669.95	0.0021
Lightower Fiber	5666.5	0.000018
Microsoft Corporation	5501.57	0.000032
TICSA-ASN	4848.11	0.000135
Vodacom	3113.68	0.000294
Web-Africa-Networks	3043.43	0.000066

4.2.5 Volume and Throughput by Protocol

In terms of the volumes for each protocol, we observe that TCP and UDP are the most common protocols used in the network, with TCP accounting for 65.09% of traffic, followed by UDP with 34.71%. Encapsulating Security Payload (ESP), Generic Routing Protocol (GRP), and ICMP are the other 3 protocols that are used by the network, although the volume of data sent through the network using these protocols is too low to be seen in comparison to TCP and UDP.

We notice that the median throughput values are slightly higher for UDP compared to TCP. ESP, GRE, and ICMP experience lower throughput values. These volume trends are important to note when designing our SDN model.

Table 4.15: Volume (GB) and median throughput (Gbps) for each protocol.

Protocol	Volume (GB)	Median throughput (Gbps)
TCP	64895.97	0.0021
UDP	34601.14	0.0027
ESP	151.82	0.000079
GRE	45.25	0.000079
ICMP	0.45	0.000085

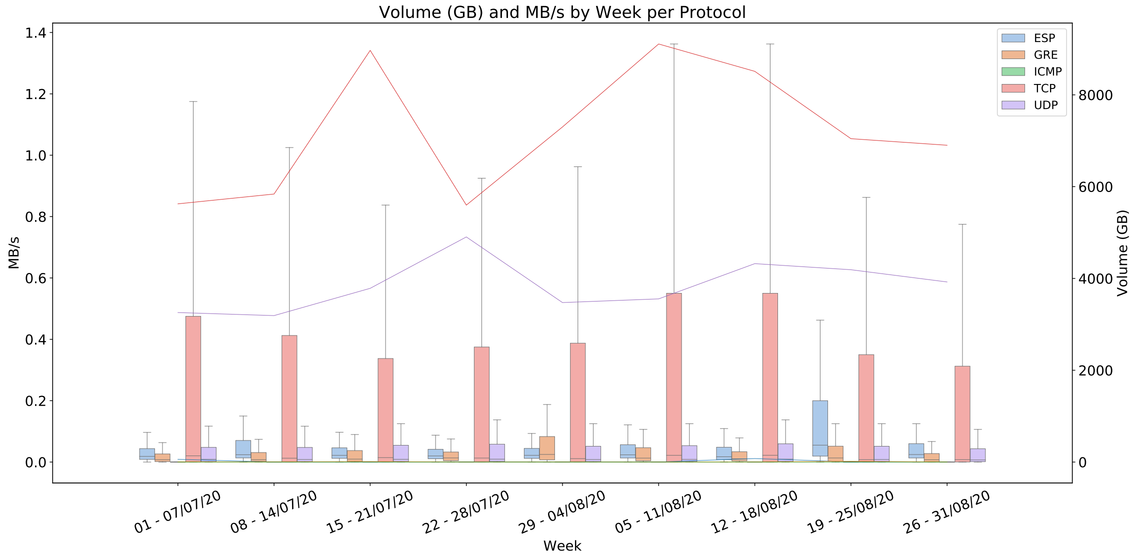


Figure 4.24: Volume (GB) and MB/s by Week per Protocol.

4.3 Performance Modelling Based on Active Measurements and Passive Dataset

In this section, we implement an SDN model of SANReN. We outline the topology and the configuration, followed by describing results from performance tests conducted on the proposed solution. We then compare the results of the SDN solution to the results obtained from active measurements conducted on the SANReN infrastructure.

4.3.1 SDN Model Topology

We used the SANReN backbone map (Figure 2.2) to create the SDN model topology as shown in Figure 3.2. We include all the nodes on the map and set link capacities to 0.001% of their real-world capacity. Hosts for each university tested in this study are attached to the respective city that they reside in. We calculate the delay between each node by using the results from the internal active measurements. To configure the delay between nodes that we do not have results for, we use a standard of 6.5 ms for 1000 km between nodes. The delays calculated from our internal active measurements are shown in Appendix A.

4.3.2 Packet Delay Results

We modelled a virtual SDN to use the delays from our internal active measurements. As shown in Figure 4.25, the overall median delay to each university is similar. However, certain source and target pairs have high median delays in our proposed solution due to the Spanning Tree Protocol closing ports to certain nodes. We observe that the SDN model has a high amount of variance to UWC, as the model achieves low median delays using the shortest path from Cape Town (2.16 ms) and PE (10.1 ms), but achieves higher delays from Durban (41.15 ms) and Pretoria (37.85 ms) because the shortest path is blocked by the spanning tree. When targeting DUT, we also observe a high amount of variance due to Cape Town (33.2 ms) and Port Elizabeth (36.75 ms) being further away and unable to achieve the lower delays achieved from Durban (0.03 ms), Johannesburg (7.09 ms), and Pretoria (7.29 ms).

We breakdown the median results from the SDN model and from the internal active measurements in Table 4.16. The SDN model experiences high delays from Cape Town to DUT with a median of 33.2 ms and Port Elizabeth to DUT with a median of 36.75 ms. This is in contrast to the internal active measurements conducted on SANReN as the delay from Cape Town to DUT from within SANReN is 27.05 ms and from Port Elizabeth to DUT is 20.59 ms. Moreover, the median delay from Durban is higher than that of SANReN. From Durban, the median delay is 38.05 ms to NMU, 33.4 ms to UCT, and 41.15 ms to UWC. In the SDN model, NMU has a higher overall median delay compared to SANReN due to the spanning tree calculating paths based on bandwidth capacity. The links to Port Elizabeth

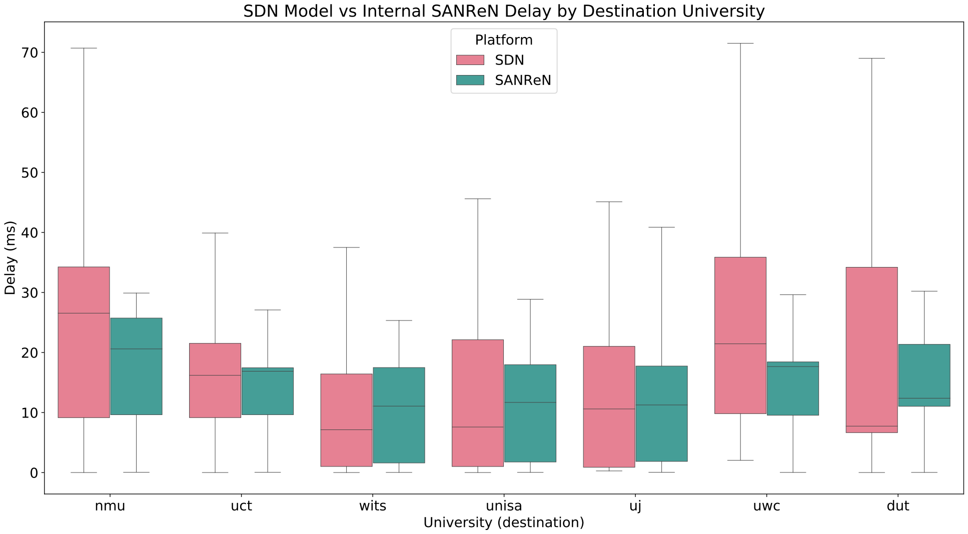


Figure 4.25: SDN model vs internal SANReN delay by destination university.

have low capacities (besides the link between PE and Cape Town with 100 Mbps) with one 1 Mbps link and two 10 Mbps links. This means that packets are travelling from Durban to NMU via Johannesburg and Cape Town, which is a circuitous route calculated by the spanning tree. This is due to the link between Durban and Johannesburg having a bandwidth capacity of 100 Mbps (a cost of 2) and the link between Durban and Pietermaritzburg having a bandwidth capacity of 20 Mbps (a cost of 200000).

Majority of the source and target pairs in the SDN model have lower delays than SANReN. For example, the median delays from Durban to WITS, UNISA, and UJ, as well as from Johannesburg to UCT, UNISA, and DUT are lower in the SDN model. The overall median delays from all cities besides Port Elizabeth are lower in the SDN model. This is due to the prioritisation of bandwidth capacity in the SDN model and the lack of bandwidth availability to and from Port Elizabeth in the SANReN topology. In addition, in terms of the overall median delays for each target university, only NMU and UWC have higher delays. As mentioned previously, with respect to DUT, this is due to the high delays from

Port Elizabeth and Cape Town (33.2 ms and 36.75 ms respectively). With respect to UWC, the high delays from Pretoria and Durban (37.85 and 41.15 ms respectively) are causing the difference between the SDN model's delay and SANReN's delay.

Table 4.16: Summary of median SDN and internal SANReN packet delays (ms) for each source and target pair.

Source	CPT		DBN		JHB		PE		PTA		Overall	
/ Dest.	SDN	Int.	SDN	Int.	SDN	Int.	SDN	Int.	SDN	Int.	SDN	Int.
NMU	9.34	9.78	**38.05**	20.6	26.55	25.42	0.02	0.24	**33.2**	26.24	**26.55**	20.6
UCT	0.02	0.66	**33.4**	26.8	16.2	16.89	9.63	9.67	19.85	17.41	16.2	16.88
WITS	16.1	16.99	7.14	11.01	0.02	0.19	26.85	25.26	1.13	1.64	7.14	11.07
UNISA	20.3	17.62	7.59	11.69	1.07	1.86	**30.55**	25.95	0.02	0.06	7.59	11.69
UJ	19.8	17.28	10.6	11.26	0.37	0.01	29	25.47	0.92	1.89	10.6	11.26
UWC	2.16	1.32	**41.15**	27.35	21.45	17.75	10.1	9.98	**37.85**	18.11	**21.45**	17.66
DUT	**33.2**	27.05	0.03	0.03	7.09	11.25	**36.75**	20.59	7.29	11.92	7.74	12.38
Overall	16.1	16.9	10.9	11.7	7.09	11.01	26.65	24.89	7.29	7.15	**11.7**	11.99

4.3.3 Page Load Time Results

We conducted the page load time tests using *Wget* on the Mininet hosts. We created a file of 31594 bytes in size to be transferred between the hosts. We use a file of 31594 bytes because five universities had a median total downloaded bytes of 31594, it was the highest median total downloaded bytes across the universities, and 83% of all the page load time tests conducted had a total downloaded bytes value equal to or lower than 31594 bytes. Using the same size file allows a comparison between SANReN and the SDN model. By starting each host as a web server using Python's SimpleHTTPServer, we download the file from each university with Wget to simulate a page load. From our results, shown in Figure 4.26, we observe that in the SDN model, the page load times are lower to each university targeted, when compared to SANReN. In Figure 4.26, *A* shows the full box plot and *B* shows the box plot zoomed in on the results.

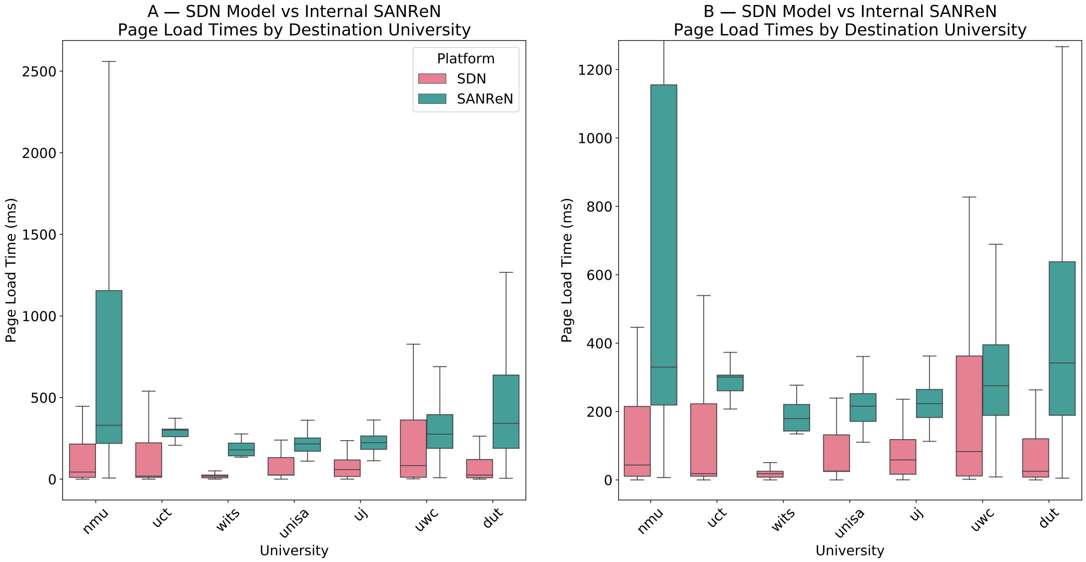


Figure 4.26: SDN model vs internal SANReN page load times by destination university.

We observe that the page load time results are lower with the SDN model compared to the SANReN results. In the SDN model, we see that the overall median page load times from the source cities are in the range of 18.05 - 58.56 ms, whereas the page load times from SANReN are in the range of 188.2 - 262.93 ms. The low page load times in the SDN model are a result of the model being optimised for bandwidth, causing high throughput values. When focusing on results from a source city to a university located in the same city, we see lower page load times in the SDN model. These differences are exaggerated in Durban, Cape Town, and Port Elizabeth due to congestion around these parts of SANReN.

Table 4.17 shows a detailed breakdown of the results of each source and target pair. From the breakdown, we observe the same high and low values compared to the delay tests, as the same areas of the network have high delays between them. The SDN model's page load times from Durban and Pretoria to NMU, UCT, and UWC are high due to the data being transferred on circuitous routes. However, despite the higher delays, the overall page load times to these universities are lower than that of SANReN. Once again, this is a result of the bandwidth prioritisation in the SDN model. Moreover, other factors causing the high page load times in SANReN are: congestion between PE and Cape Town, congestion between Cape Town and Durban, and circuitous routes traversed from PE to Johannesburg.

Table 4.17: Summary of median SDN and internal SANReN page load times (ms) for each source and target pair.

Source	CPT		DBN		JHB		PE		PTA		Overall	
/ Dest.	SDN	Int.	SDN	Int.	SDN	Int.	SDN	Int.	SDN	Int.	SDN	Int.
NMU	11.07	561.49	**219.41**	329.75	43.58	374.75	0.04	212.48	**233.17**	314.61	43.58	329.75
UCT	0.05	216.39	**244.91**	350.32	18.21	301.88	11.49	261.63	**244.91**	301.81	18.37	300.8
WITS	14.36	216.72	12.04	179.51	0.04	137.71	27.24	237.07	25.28	143.11	14.36	179.55
UNISA	137.67	240.77	25.48	215.63	25.07	173.55	141.36	272.09	0.06	164.49	25.69	215.68
UJ	120.36	256.71	18.86	222.75	0.38	178.66	127.4	280.47	58.56	183.31	58.45	223.11
UWC	1.94	250.73	**374.78**	305.22	82.93	316.56	11.79	229.99	**382.26**	258.65	82.93	275.48
DUT	119.44	510.43	0.05	283.69	8.85	244.75	136.19	362.75	25.28	361	25.28	341.82
Overall	18.05	236.55	26	226.51	18.26	188.2	36.78	262.93	58.56	189.55	**25.48**	237.5

4.3.4 Throughput Results

In the results of the throughput tests conducted with the SDN model, shown in Figure 4.27, we observe that the throughput from Cape Town to UNISA and UJ is 1.16 Mbps and 1.19 Mbps respectively. The rest of the universities experience median throughput values of greater than 9 - 9.46 Mbps. We observe similar throughput values from Durban and Johannesburg with overall medians of 9.04 Mbps and 9.41 Mbps respectively. From PE and Pretoria, we observe lower throughput values to NMU, UNISA, UJ, and UWC. However, these low throughput values are due to Mininet restricting the bandwidth to each switch. Table 4.18 shows the median throughput of each source and target pair. We do not compare these values directly to the SANReN throughput values due to the network being different scales. However, by observing the trends between nodes, we observe that the overall throughput from Cape Town, Durban, and Johannesburg is increased.

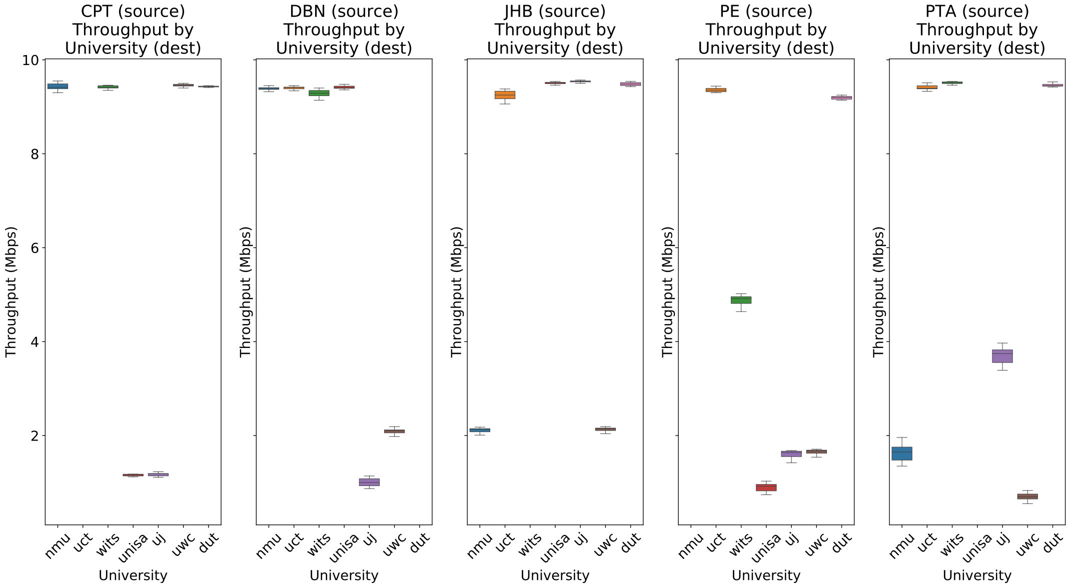


Figure 4.27: SDN model throughput (Mbps) by destination university.

Moreover, we observe that the network makes use of the capacity to each university, within the restrictions of Mininet. Thus, Pretoria and Port Elizabeth have lower throughput values.

Table 4.18: Summary of median throughput (Mbps) for each source and target pair.

Source / Dest.	CPT	DBN	JHB	PE	PTA	Overall
nmu	9.42	9.39	2.13		1.65	**5.74**
uct		9.41	**9.25**	9.36	9.4	9.38
wits	**9.42**	9.29		4.92	9.51	9.37
unisa	1.16	9.41	9.51	0.92		**5.27**
uj	**1.19**	1	9.55	1.64	**3.745**	1.64
uwc	9.46	2.11	**2.15**	1.68	0.7	2.07
dut	9.43		9.49	9.2	9.45	9.44
Overall	9.41	9.36	9.41	**3.15**	**6.65**	9.27

Chapter 5

Discussion

This study has focused on finding the performance issues within SANReN. We found that there are gaps in the network in various parts of the country. In this section, we discuss the problem areas of the network and their causes. We discuss the usage patterns of the network as well and how the network can be engineered to focus on the high usage areas. We also discuss the potential of SDN as a solution to these issues.

5.1 Delays

The results of the delay experiments show that SANReN experiences high delays around Cape Town and Port Elizabeth. We observe that the external and internal tests conducted from these cities are the highest across the network. We found that the lowest delays are achieved from Johannesburg and Pretoria for both the internal and external tests. When looking at the target universities, we observe that NMU, UCT, and UWC have the highest median delays. This points to congestion as there are high delays from the cities these universities are located in as well.

5.1.1 Delays to and from Port Elizabeth

The problem area in the network is Port Elizabeth, as the delays to reach a web server hosted by NMU from outside of SANReN (57 ms) is 35.71% higher compared to the overall median of reaching a university from outside of SANReN (42 ms). Moreover, for the external tests, the overall median delay from PE to the universities tested (53 ms) is 26.19% higher than the median of all the source cities combined (42 ms). When comparing the internal median results to the results from PE, we see that PE's delays are 107.59% higher than the overall median. We also see that the delay when targeting NMU is 71.81% higher than the overall median. The high delays for the internal tests are due to packets travelling via Cape Town to Johannesburg or Pretoria, which means that the paths to Johannesburg via East London and Bloemfontein are ignored. The distance from PE to Johannesburg via East London and Bloemfontein is ~1070.51 km, whereas travelling to Johannesburg via Cape Town is ~1926.57 km.

SANReN has four links connecting to PE — two 10 Gbps links to Makhanda and East London, one 1 Gbps link to George and one 100 Gbps link to Cape Town. PE is not considered a metropolitan network, and, as such, the only metropolitan network that has a point of presence (POP) directly to PE is Cape Town, the other the metropolitan networks do not go directly to PE. Therefore, the 100 Gbps link is used to transfer data to PE from various parts of the country. However, there are shorter paths to transfer data from PE. With SDN, the network can be configured to transfer data intensive traffic via Cape Town, and time sensitive traffic via East London and Bloemfontein. We show that the SDN model also has high delays to NMU, caused by low bandwidth capacities in to NMU, and the spanning tree prioritising high bandwidth links. However, the SDN model can be configured to use these redundant links in the future, and the model can be expanded, and different types of traffic (VoIP, HTTP, Gaming, Streaming, etc.) can be catered for.

5.2 Page Load Times

5.2.1 Page Load Times from Port Elizabeth

Our results show that PE has high page load times as well. The external results when targeting NMU produced a median of 1727 ms, which is high compared to the overall median from all source cities of 491 ms (Table 4.2). When looking at the external traceroute results, we see that congestion is occurring inside SANReN at the nodes in Cape Town when trying to reach NMU, with hops in the Cape Town TENET nodes taking up to 7724 ms, where the median to reach the node is 104 ms. We show that the SDN model has a median page load time which is 86.01% lower than the internal active measurement results.

5.2.2 Page Load Times from Durban

From the results, we see that page load times from Durban to UCT and to UWC are high as the external and internal median results are higher than the overall median results to these universities (Table 4.2). Since the median TTFB from Durban to all universities (390 ms) is similar to the overall median (416 ms), we look at the throughput of these tests to find the cause of the issue. Our external results show that from Durban to UCT and UWC, we experience a throughput of 3.2 Mbps and 3.77 Mbps respectively. Whereas our internal results show that the throughput from Durban to UCT and UWC is 4910 Mbps and 7540 Mbps respectively. The throughput from Durban to UCT is the lowest of all that were tested. These low bandwidths are causing the high page load times in the network between these two cities. Our traceroute results show that packets are travelling from Durban to Cape Town via PE. When comparing the links between Durban and Cape Town, we see that there are links with capacities of 10 Gbps between Durban and PE, and a link of 100 Gbps between PE and Cape Town. However, the throughput experienced while using these paths are not reflective of the current capabilities. The throughput is affected by the congestion between Cape Town and Port Elizabeth, as well as the overall usage of SANReN between these cities. In terms of our SDN model, we see that the throughput between Durban and Cape Town reflects the network capabilities, as the SDN model achieves 9.41 Mbps from Durban to UCT and 2.11 Mbps to UWC.

5.3 Effects of Performance on QoE

When looking at the SANReN usage patterns, we notice that the network has high throughput values (median of 0.759 Mbps) when traffic flows inbound (UCT receiving data from other sources). However, users that request data from outside of UCT (outbound flows) experience lower throughput (median of 0.068 Mbps), and thus, slower performance. Since the majority of traffic is outbound (69.61% — Table 4.12), majority of the traffic is transferred with lower throughput. Furthermore, since majority of the data being transferred through the network uses TCP, users at UCT requesting data from other destinations are relying on the network to deliver content accurately and in full. Therefore, to achieve the best performance, the network would need to increase the bandwidth links of the shortest paths, since these paths are being ignored for high bandwidth paths.

5.4 Challenges and Limitations

5.4.1 Active measurements

One of the challenges faced in the study was that the zero-rated web servers hosted by some universities did not allow us to ping them. Thus, we resorted to using the perfSONAR nodes which also reside inside SANReN for our tests. This skewed some of the data in the internal results section. To mitigate the skewed results, we balanced out the tests by adding each university's perfSONAR node as a destination web server. In terms of the internal page load time tests, perfSONAR does not return the details of time-to-first-byte and total downloaded bytes. Thus, we were unable to calculate the throughput experienced in the same manner as the external tests. Speedchecker, however, does not have a dedicated feature for testing throughput. Thus, we could not test the throughput and compare it to the internal tests. Instead, we compare the trends of throughput to each university from external and internal sources.

5.4.2 Passive measurements

The SANReN dataset provided information on the flows of traffic to and from UCT. However, the dataset does not contain the type of traffic for each flow. Therefore, we are unable to find out what the purpose of the traffic was besides looking up the ISP and the ASN that the IPs reside in. This limits our analysis because we could not break down the trends using parameters other than the ISP, protocol, and ASN. In addition, the SANReN dataset only contained flows during a two-month period in 2020. We could not gain broader insights into the trends of the traffic and the impact of the Coronavirus on UCT's traffic flow due to this constraint.

5.4.3 SDN Model

Our SDN model uses a standard implementation of the Spanning Tree Protocol with Ryu and Mininet. From our results, the optimal path between some nodes are blocked by the spanning tree and thus, we do not get optimal delays between all of our hosts. Path costs are chosen based on bandwidth capacity. A lower bandwidth link will have a higher cost than a high bandwidth link, regardless of delays between the nodes. Thus, we see high delays when targeting NMU and DUT. This SDN model can be used and built on to find the optimal setup for SANReN. The costs allocated to the links in the network can be changed to improve performance of the network.

Chapter 6

Conclusion

6.1 Summary

In this study, we focused on finding the performance issues existing in SANReN by using active and passive measurements. We explore how an SDN model using SANReN's topology would affect performance of the network. We posed three research questions in 1.2, and the answers to these research questions are outlined below:

Where are the performance gaps in the topology of the network, what is causing these issues and how can we improve the network resource utilisation?

Our measurements found that Port Elizabeth is the city with the highest delays and highest page load times to universities in other cities. Cape Town has the second highest delays and page load times to universities in other cities, followed by Durban. The high delays and page load times experienced from PE is caused by traffic flowing via circuitous routes, using Cape Town as a PoP. The high delay to PE is amplified by going through Cape Town because Cape Town is located far from the rest of the cities in the network, and thus, also experiences high delays. The high page load times from Durban to UCT and to UWC is a result of low bandwidth capacities between these two cities. Although traffic flows via PE, which is linked to Cape Town via a 100 Gbps link, the throughput experienced is much lower at 4.91 Gbps to UCT and 7.54 Gbps to UWC when the network is in use.

Which countries, ISPs, and protocols produce the highest volume of traffic and during which periods does the network experience the most traffic flow?

Our passive measurements show that majority of traffic flows outbound from UCT (uploaded data). However, the throughput of these flows is lower compared to the throughput of data flowing inbound (downloaded data). UCT experiences typical high usage during the week and low usage during the weekend. Majority of the traffic flowing in and out of UCT are using South African ISPs, and thus, increasing the inter-connectivity within SANReN in South Africa would be beneficial for most UCT users of the network. The three ISPs with the highest data usage are Telkom (17479.11 GB), TENET (14505.68 GB), and Afrihost (9525.33 GB). Our results show that besides South Africa, the United States is the country with the highest amount of traffic flowing to and from UCT, followed by Australia. The peak traffic flow in these countries were caused by traffic received from Lightower in the US and by fetching data from the Australian Academic and Research Network (AARNet) totalling 378.46 GB. In terms of protocols, TCP had the highest amount of data usage with 64895.97 GB, followed by UDP with 34601.14 GB. Thus, to improve performance for most use cases, the focus should be on traffic using TCP.

To what extent, if any, would an SDN framework improve SANReN's network performance in terms of delays, page load times, and throughput?

We created an SDN model to compare the performance to that of the current SANReN design. Our model was able to produce lower page load times and lower delays to 5 of the 7 universities tested with an overall median delay of 11.7 ms for the SDN model compared to the median delay of 11.99 ms for internal SANReN tests. With respect to PLT, the SDN model achieved an overall median of 25.48 ms whereas the internal SANReN tests had an overall median of 237.5 ms. Since the SDN model is a scaled down version of SANReN, we do not compare the throughput values. However, by observing the trends between nodes, we observe that the overall throughput from Cape Town, Durban, and Johannesburg is increased. Moreover, we observe that the network makes use of the capacity to each university, within the restrictions of Mininet. Thus, Pretoria and Port Elizabeth have lower throughput values. These limitations can be mitigated by tweaking the configuration of the SDN model for optimal performance.

6.2 Future Work

For future work, we could increase the number of cities and universities included in the study to gain a broader perspective of the network's performance. We could also use rest of the universities' traffic flow data to conduct the passive measurements on their networks and find the performance and usage statistics of those universities. In addition to delay and PLT, we could conduct video streaming tests to find out robustness of connectivity and download speed.

In terms of SDN, the configuration for the network can be expanded upon, and the network can be tweaked to find the optimal performance. Additional traffic engineering paradigms can be incorporated such as Software Resolved Networking (SRN) which is SDN with a DNS resolver and Segment Routing [52].

Appendices

Appendix A

Appendices I

Table A.1: Delays between hosts and switches calculated from internal active measurements.

Host	Switch	Bandwidth (Mbps)	Delay (ms)	Cost
NMU	PE	10	0.12	2000000
UCT	Cape Town	10	0.33	2000000
WITS	JHB	10	0.1	2000000
UNISA	Tshwane	10	0.03	2000000
UJ	JHB	10	0.01	2000000
UWC	Cape Town	10	0.66	2000000
DUT	eThekwini	10	0.02	2000000

Table A.2: Delays calculated from internal active measurements.

Node 1	Node 2	Bandwidth (Mbps)	Delay (ms)	Cost
Cape Town	JHB	100	7.31	2
Cape Town	PE	100	3.97	2
JHB	Tshwane	10	0.31	2000000
eThekwini	JHB	100	3.00	2
Cape Town	Bloemfontein	20	5.53	200000
East London	Bloemfontein	30	3.94	20000
PE	East London	10	2.11	2000000
Mtunzini	Tshwane	10	4.28	2000000
Mtunzini	JHB	60	4.11	200
eThekwini	Mtunzini	20	1.02	200000
Bloemfontein	JHB	50	3.16	2000
Cape Town	Yzerfontein	20	0.60	200000
Cape Town	Hermanus	10	0.77	2000000
Cape Town	SKA1	10	4.02	2000000
Cape Town	SALT	10	2.40	2000000
Cape Town	Beaufort West	10	3.63	2000000
Cape Town	George	10	3.16	2000000
Hermanus	George	1	2.57	20000000
George	PE	1	2.48	20000000
Beaufort West	Bloemfontein	10	4.28	2000000
Beaufort West	SKA1	10	1.37	2000000
SKA1	SKA2	10	0.52	2000000
Kimberly	Welkom	10	1.80	2000000
Kimberly	Potchefstroom	10	2.74	2000000
PE	Makhanda	10	0.94	2000000
Makhanda	Bhisho	10	0.86	2000000
Bhisho	East London	10	0.38	2000000
Bhisho	Alice	10	0.48	2000000
Bhisho	Whittlesea	10	0.82	2000000
Bhisho	Potsdam	10	0.18	2000000

Table A.3: Delays calculated from internal active measurements (continued).

Node 1	Node 2	Bandwidth (Mbps)	Delay (ms)	Cost
East London	Potsdam	10	0.18	2000000
East London	Butterworth	10	0.70	2000000
East London	Mthatha	10	1.56	2000000
Butterworth	Mthatha	10	0.88	2000000
Whittlesea	Komani	10	0.26	2000000
Komani	Mthatha	10	0.88	2000000
Mthatha	Pietermaritzburg	10	2.31	2000000
Bloemfontein	Potchefstroom	100	2.40	2
Bloemfontein	Welkom	10	1.20	2000000
Welkom	Kroonstad	1	0.52	20000000
Welkom	Phuthaditjhaba	10	1.84	2000000
Kroonstad	Bethlehem	1	1.06	20000000
Bethlehem	Phuthaditjhaba	1	0.52	20000000
Bethlehem	eThekwini	1	2.74	20000000
Vanderbijlpark	Potchefstroom	10	0.60	2000000
Vanderbijlpark	JHB	10	0.52	2000000
eThekwini	Pietermaritzburg	20	0.56	200000
JHB	SANSA	10	0.42	2000000
SANSA	Tshwane	10	0.46	2000000
Tshwane	eMalahleni	10	0.88	2000000
Tshwane	Polokwane	10	2.06	2000000
Tshwane	Rustenburg	10	0.86	2000000
Potchefstroom	Mahikeng	10	1.48	2000000
Rustenburg	Mahikeng	10	1.37	2000000
eMalahleni	Middelburg	10	0.20	2000000
Middelburg	Mbombela	10	1.32	2000000
Mbombela	Thohoyandou	10	2.50	2000000
Thohoyandou	Makhado	10	0.52	2000000
Makhado	Turfloop	10	0.54	2000000
Turfloop	Polokwane	10	0.26	2000000

Bibliography

[1] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, *et al.*, "The lockdown effect: Implications of the covid-19 pandemic on internet traffic," in *Proceedings of the ACM Internet Measurement Conference*, pp. 1–18, 2020.

[2] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for sdn? implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, 2013.

[3] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[4] V. Mohan, Y. J. Reddy, and K. Kalpana, "Active and passive network measurements: a survey," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 4, pp. 1372–1385, 2011.

[5] M. J. Khokhar, N. A. Saber, T. Spetebroot, and C. Barakat, "On active sampling of controlled experiments for qoe modeling," in *Proceedings of the Workshop on QoE-based Analysis and Management of Data Communication Networks*, pp. 31–36, 2017.

[6] M. Trevisan, I. Drago, and M. Mellia, "Pain: A passive web performance indicator for isps," *Computer Networks*, vol. 149, pp. 115–126, 2019.

[7] P. Casas, M. Seufert, F. Wamser, B. Gardlo, A. Sackl, and R. Schatz, "Next to you: Monitoring quality of experience in cellular networks from the end-devices," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 181–196, 2016.

[8] M. Chetty, S. Sundaresan, S. Muckaden, N. Feamster, and E. Calandro, "Measuring broadband performance in south africa," in *Proceedings of the 4th Annual Symposium on Computing for Development*, pp. 1–10, 2013.

[9] B. Chandra, T. Koizumi, M. Hayashi, and H. Yamamoto, "A study on network reliability evaluation for developing countries," in *2012 18th Asia-Pacific Conference on Communications (APCC)*, pp. 749–753, IEEE, 2012.

[10] J. Chavula, N. Feamster, A. Bagula, and H. Suleman, "Quantifying the effects of circuitous routes on the latency of intra-africa internet traffic: a study of research and education networks," in *International Conference on e-Infrastructure and e-Services for Developing Countries*, pp. 64–73, Springer, 2014.

[11] T. Böttger, G. Ibrahim, and B. Vallis, "How the internet reacted to covid-19: A perspective from facebook's edge network," in *Proceedings of the ACM Internet Measurement Conference*, pp. 34–41, 2020.

[12] B. B. Lowekamp, "Combining active and passive network measurements to build scalable monitoring systems on the grid," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 4, pp. 19–26, 2003.

[13] S. Spoto, R. Gaeta, M. Grangetto, and M. Sereno, "Analysis of pplive through active and passive measurements," in *2009 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1–7, IEEE, 2009.

[14] C.-C. Chiu and T. He, "Stealthy dgos attack under passive and active measurements," in *IEEE Global Communications Conference*, 2020.

[15] J. L. de Vergara, M. Ruiz, L. Gifre, L. Vaquero, J. Zazo, S. López-Buedo, O. G. de Dios,

and L. Velasco, “Demonstration of 100 gbit/s active measurements in dynamically provisioned optical paths,” 2019.

[16] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, “A high-performance, scalable infrastructure for large-scale active dns measurements,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, pp. 1877–1888, 2016.

[17] A. Nikravesh, H. Yao, S. Xu, D. Choffnes, and Z. M. Mao, “Mobilyzer: An open platform for controllable mobile network measurements,” in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 389–404, 2015.

[18] S. Sonntag, L. Schulte, and J. Manner, “Mobile network measurements-it’s not all about signal strength,” in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 4624–4629, IEEE, 2013.

[19] A. Hanemann, J. W. Boote, E. L. Boyd, J. Durand, L. Kudarimoti, R. Łapacz, D. M. Swany, S. Trocha, and J. Zurawski, “Perfsonar: A service oriented architecture for multi-domain network monitoring,” in *International conference on service-oriented computing*, pp. 241–254, Springer, 2005.

[20] N. Duffield *et al.*, “Sampling for passive internet measurement: A review,” *Statistical Science*, vol. 19, no. 3, pp. 472–498, 2004.

[21] W. John, S. Tafvelin, and T. Olovsson, “Passive internet measurement: Overview and guidelines based on experiences,” *Computer Communications*, vol. 33, no. 5, pp. 533–550, 2010.

[22] C. Chaudet, E. Fleury, I. G. Lassous, H. Rivano, and M.-E. Voge, “Optimal positioning

of active and passive monitoring devices," in *Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, pp. 71–82, 2005.

[23] N. L. Van Adrichem, C. Doerr, and F. A. Kuipers, "Opennetmon: Network monitoring in openflow software-defined networks," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–8, IEEE, 2014.

[24] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[25] K. Wierenga and L. Florio, "Eduroam: past, present and future," *Computational methods in science and technology*, vol. 11, no. 2, pp. 169–173, 2005.

[26] J. Zurawski, S. Balasubramanian, A. Brown, E. Kissel, A. Lake, M. Swany, B. Tierney, and M. Zekauskas, "perfsonar: On-board diagnostics for big data," in *1st Workshop on Big Data and Science: Infrastructure and Services. Co-located with IEEE International Conference on Big Data*, Citeseer, 2013.

[27] S. McKee, A. Lake, P. Laurens, H. Severini, T. Wlodek, S. Wolff, and J. Zurawski, "Monitoring the us atlas network infrastructure with perfsonar-ps," in *Journal of Physics: Conference Series*, vol. 396, p. 042038, IOP Publishing, 2012.

[28] P.-H. Chen, "Smart browser: Network measurement system based on perfsonar framework," in *2011 13th Asia-Pacific Network Operations and Management Symposium*, pp. 1–4, IEEE, 2011.

[29] Y. Zhang, S. Debroy, and P. Calyam, "Network-wide anomaly event detection and diagnosis with perfsonar," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 666–680, 2016.

[30] S. Campana, A. Brown, D. Bonacorsi, V. Capone, D. De Girolamo, A. Casani, J. Flix, A. Forti, I. Gable, O. Gutsche, *et al.*, “Deployment of a wlcg network monitoring infrastructure based on the perfsonar-ps technology,” in *Journal of Physics: Conference Series*, vol. 513, p. 062008, IOP Publishing, 2014.

[31] T. Arnold, E. Gürmeriçliler, G. Essig, A. Gupta, M. Calder, V. Giotsas, and E. Katz-Bassett, “(how much) does a private wan improve cloud performance?,” in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 79–88, IEEE, 2020.

[32] R. N. Staff, “Ripe atlas: A global internet measurement network,” *Internet Protocol Journal*, vol. 18, no. 3, 2015.

[33] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, “Planetlab: an overlay testbed for broad-coverage services,” *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 3–12, 2003.

[34] A. Formoso, J. Chavula, A. Phokeer, A. Sathiaseelan, and G. Tyson, “Deep diving into africa’s inter-country latencies,” in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 2231–2239, IEEE, 2018.

[35] J. Chavula, A. Phokeer, A. Formoso, and N. Feamster, “Insight into africa’s country-level latencies,” in *2017 IEEE AFRICON*, pp. 938–944, 2017.

[36] M. Candela, V. Luconi, and A. Vecchio, “Impact of the covid-19 pandemic on the internet latency: a large-scale study,” *arXiv preprint arXiv:2005.06127*, 2020.

[37] A. Abdelkefi and Y. Jiang, “A structural analysis of network delay,” in *2011 Ninth Annual Communication Networks and Services Research Conference*, pp. 41–48, IEEE, 2011.

[38] H. Pucha, Y. Zhang, Z. M. Mao, and Y. C. Hu, "Understanding network delay changes caused by routing events," *ACM SIGMETRICS performance evaluation review*, vol. 35, no. 1, pp. 73–84, 2007.

[39] J. A. Jacko, A. Sears, and M. S. Borella, "The effect of network delay and media on user perceptions of web resources," *Behaviour & Information Technology*, vol. 19, no. 6, pp. 427–439, 2000.

[40] X. S. Wang, A. Balasubramanian, A. Krishnamurthy, and D. Wetherall, "Demystifying page load performance with wprof," in *10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*, pp. 473–485, 2013.

[41] N. Bhatti, A. Bouch, and A. Kuchinsky, "Integrating user-perceived quality into web server design," *Computer Networks*, vol. 33, no. 1-6, pp. 1–16, 2000.

[42] D. N. da Hora, A. S. Asrese, V. Christophides, R. Teixeira, and D. Rossi, "Narrowing the gap between qos metrics and web qoe using above-the-fold metrics," in *International Conference on Passive and Active Network Measurement*, pp. 31–43, Springer, 2018.

[43] S. A. Jyothi, A. Singla, P. B. Godfrey, and A. Kolla, "Measuring and understanding throughput of network topologies," in *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 761–772, IEEE, 2016.

[44] W. Luan, D. Sharp, and S. Lancashire, "Smart grid communication network capacity planning for power utilities," in *IEEE PES T&D 2010*, pp. 1–4, IEEE, 2010.

[45] M. Mortimer, "iperf3 documentation," 2018.

[46] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-wifi:

Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 384–389, IEEE, 2015.

[47] F. Tomonori, "Introduction to ryu sdn framework," *Open Networking Summit*, pp. 1–14, 2013.

[48] W. Wojdak, "Rapid spanning tree protocol: A new solution from an old technology," *Reprinted from CompactPCI Systems*, 2003.

[49] J. Jay, "Low signal latency in optical fiber networks," in *Corning Optical Fiber, Proc. of the 60th IWCS, Conference, Charlotte, NC, USA,(6-9 Nov. 2011). Citeseer*, Citeseer, 2011.

[50] L. Feris, "Limited invitations to return to residence," *UCT News*, 2020.

[51] L. Lange, "Guidelines on completing 2020 academic year," *UCT News*, 2020.

[52] D. Lebrun, M. Jadin, F. Clad, C. Filsfils, and O. Bonaventure, "Software resolved networks: Rethinking enterprise networks with ipv6 segment routing," in *Proceedings of the Symposium on SDN Research*, pp. 1–14, 2018.