UNIVERSITY OF CAPE TOWN

DOCTORAL THESIS

Improving Pan-African Research and Education Networks Through Traffic Engineering: A LISP/SDN Approach

Author:
Josiah CHAVULA

Supervisors:
A/Prof. Hussein SULEMAN,
Dr. Melissa DENSMORE

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

ICT4D Centre
Department of Computer Science

October 5, 2017

Declaration of Authorship

I, Josiah CHAVULA, declare that this thesis titled, "Improving Pan-African Research and Education Networks Through Traffic Engineering: A LISP/SDN Approach" my own original work. Where collaborations with other researchers are involved, or materials generated by other researchers are included, the parties and/or materials are acknowledged or are explicitly referenced as appropriate.

This work is being submitted for the degree of Doctor of Philosophy in Computer Science at the University of Cape Town, South Africa. This thesis has not been submitted to any other university or institution for any other degree or examination.

Signed: Signed by candidate

Date: 5 October, 2017

Publications

Early versions of some of the ideas and figures presented in this dissertation have previously appeared in the following publications:

- **Josiah Chavula**, Nick Feamster, Antoine B. Bagula and Hussein Suleman (2014) Quantifying the Effects of Circuitous Routes on the Latency of Intra-Africa Internet Traffic: A Study of Research and Education Networks, in Proceedings of AFRICOMM 2014, Kampala, Uganda, pp. 64-73.
  Available http://pubs.cs.uct.ac.za/archive/00000995/

- **Josiah Chavula**, Melissa Densmore, and Hussein Suleman (2015) Reducing Latency in African NRENs Using Performance-based LISP/SDN Traffic Engineering, in Proceedings of NetCom 2015, Sydney, Australia, 26-27 December 2015.
  Available http://pubs.cs.uct.ac.za/archive/00001050/

- **Josiah Chavula**, Hussein Suleman and Melissa Densmore (2016) Using SDN and Reinforcement Learning for Traffic Engineering in UbuntuNet Alliance, in Proceedings of 3rd IEEE International Conference on Advances in Computing, Communication and Engineering (ICACCE 2016), Durban, South Africa, 28-29 November 2016.
  Available http://pubs.cs.uct.ac.za/archive/00001162/

- Sanby, Roslyn, Hussein Suleman, and **Josiah Chavula**. (2016) Efficient Topology Discovery for African NRENs, in Proceedings of IST-Africa 2016, Durban, South Africa, 11-13 May 2016.
  Available http://pubs.cs.uct.ac.za/archive/00001166/

- Chantal Yang, Hussein Suleman, and **Josiah Chavula**. (2016) A Topology Visualisation Tool for National Research and Education Networks in Africa, in Proceedings of IST-Africa 2016, Durban, South Africa, 11-13 May 2016.
  Available http://pubs.cs.uct.ac.za/archive/00001167/

Abstract

The UbuntuNet Alliance, a consortium of National Research and Education Networks (NRENs) runs an exclusive data network for education and research in east and southern Africa. Despite a high degree of route redundancy in the Alliance's topology, a large portion of Internet traffic between the NRENs is circuitously routed through Europe. This thesis proposes a performance-based strategy for dynamic ranking of inter-NREN paths to reduce latencies. The thesis makes two contributions: firstly, mapping Africa's inter-NREN topology and quantifying the extent and impact of circuitous routing; and, secondly, a dynamic traffic engineering scheme based on Software Defined Networking (SDN), Locator/Identifier Separation Protocol (LISP) and Reinforcement Learning.

To quantify the extent and impact of circuitous routing among Africa's NRENs, active topology discovery was conducted. Traceroute results showed that up to 75% of traffic from African sources to African NRENs went through inter-continental routes and experienced much higher latencies than that of traffic routed within Africa. An efficient mechanism for topology discovery was implemented by incorporating prior knowledge of overlapping paths to minimize redundancy during measurements. Evaluation of the network probing mechanism showed a 47% reduction in packets required to complete measurements. An interactive geospatial topology visualization tool was designed to evaluate how NREN stakeholders could identify routes between NRENs. Usability evaluation showed that users were able to identify routes with an accuracy level of 68%.

NRENs are faced with at least three problems to optimize traffic engineering, namely: how to discover alternate end-to-end paths; how to measure and monitor performance of different paths; and how to reconfigure alternate end-to-end paths. This work designed and evaluated a traffic engineering mechanism for dynamic discovery and configuration of alternate inter-NREN paths using SDN, LISP and Reinforcement Learning. A LISP/SDN based traffic engineering mechanism was designed to enable NRENs to dynamically rank alternate gateways. Emulation-based evaluation of the mechanism showed that dynamic path ranking was able to achieve 20 % lower latencies compared to the default static path selection. SDN and Reinforcement Learning were used to enable dynamic packet forwarding in a multipath environment, through hop-by-hop ranking of alternate links based on latency and available bandwidth. The solution achieved minimum latencies with significant increases in aggregate throughput compared to static single path packet forwarding.

Overall, this thesis provides evidence that integration of LISP, SDN and Reinforcement Learning, as well as ranking and dynamic configuration of paths could help Africa's NRENs to minimise latencies and to achieve better throughputs.

Acknowledgements

First and foremost, I would like to thank my supervisors, Associate Professor Hussein Suleman and Dr. Melissa Densmore, for their support, guidance and mentorship throughout my studies. Thank you for your patience and all the encouragement. I am thankful for the delightful *braaing* moments that I and my family were privileged to enjoy in your homes.

I sincerely thank Professor Antoine Bagula (now at University of Western Cape), who inspired this project and supervised me for the first year of my studies. I have also enjoyed the encouragement and support of my lab mates in the UCT's ICT4D Centre. Thank you all for the discussions and your feedback.

I was privileged to collaborate with three very talented honours students: Chantal Yang, Roslyn Sanby, and Robert Pasmore. They helped with the implementation of some of the tools used for topology measurements and data visualization.

I thank Professor Nick Feamster for his advice and guidance. His insights into network measurements and SDN were very helpful. I thank Dino Farinacci for his guidance and insights into LISP aspects of the research. I also thank him for allowing the use his *lispers.net* LISP implementation for this research.

I am grateful to the UCT's eResearch unit for providing the high performance computing resources that were used in experiments. I am thankful to Young Hyun and CAIDA for allowing me use of the Archipelago platform for topology measurements. Various individuals working in different African NRENs provided me guidance. I am particularly grateful for the support of Professor Meoli Kashorda, CEO of Kenya's NREN, and Mr. Issac Kasana, CEO of Uganda's NREN. I thank Tiwonge Msulira Banda and Joe Kimaili of UbuntuNet Alliance, who also provided useful input and guidance.

I am grateful for the financial support that I received, initially from Mzuzu University, and later from the Hasso Plattner Institute. I am also thankful for travel grants that I received from the Internet Society (ISOC), NEPAD e-Africa, and the ISP Association of South Africa (ISPA).

Finally, I would like to thank my family and friends for their support. I thank my wife Catherine for her friendship, love, patience and unyielding support. Natasha and Nathaniel, you two have been most adorable, thank you for all the laughter, you made this journey all the more fun and worthwhile. I owe a great debt of gratitude to my parents, brothers and sisters, and my in-laws. I thank the many friends at Mowbray Presbyterian Church for their support and encouragement through out my studies. Thank you to my lab mates in the ISAT Lab and ICT4D Centre at UCT. I am thankful for the friendship of Samuel and Barenice Mayuni, all the friends at SCOM, and many others too numerous to mention.

This thesis would not have been possible without each one of you and I am greatly indebted to you all. THANK YOU!

Contents

List of Figures

List of Tables

List of Abbreviations

AS	**A**utonomous **S**ystem
ASN	**A**utonomous **S**ystem **N**umber
BGP	**B**order **G**ateway **P**rotocol
CAIDA	**C**entre for **A**pplied **I**nternet **D**ata **A**nalysis
DNS	**D**omain **N**ame **S**ystemas
EASSY	**E**ast **A**frica **S**ubmarine **C**able **S**ystem
EIDs	**E**ndpoint **I**Dentifiers
ETR	**E**gress **T**unnel **R**outers
GMPLS	**G**eneralized **M**ultiprotocol **L**abel **S**witching
ICMP	**I**nternet **C**ontrol **M**essage **P**rotocol
IP	**I**nternet **P**rotocol
IRR	**I**nternet **R**outing **R**egistries
ISP	**I**nternet **S**ervice **P**rovider
ITR	**I**ngress **T**unnel **R**outers
IXP	**I**nternet e**X**change **P**oint
LISP	**L**ocator/**I**dentifier **S**eparation **P**rotocol
LLDP	**L**ink **L**ayer **D**iscovery **P**rotocol
MDA	**M**ulti-path **D**iscovery **A**lgorithm
MDP	**M**arkov **D**ecision **P**rocesses
MPTCP	**M**ultiPath **TCP**
NAT	**N**etwork **A**ddress **T**ranslation
NOC	**N**etwork **O**perations **C**entre
NRENs	**N**ational **R**esearch and **E**ducation **N**etworks
PingER	**Ping E**nd-to-end **R**eporting
PoP	**P**oint **o**f **P**resence
QoS	**Q**uality **o**f **S**ervice
RIPE	**R**éseaux **IP E**uropéens
RLOCs	**R**outing **LOC**ators
RREN	**R**egional **R**esearch and **E**duction **N**etwork
RTT	**R**ound **T**rip **T**ime
SDN	**S**oftware **D**efined **N**etworks
SKA	**S**quare-**K**ilometer-**A**rray

SNMP	**S**imple **N**etwork **M**anagement **P**rotocol
STP	**S**panning **T**ree **P**rotocol
TCP	**T**ransmission **C**ontrol **P**rotocol
TD	**T**emporal **D**ifference
TE	**T**raffic **E**ngineering
TEAMS	**T**he **E**ast **A**frican **M**arine **C**able **S**ystem
TTL	**T**ime-**T**o-**L**ive
UCD	**U**ser-**C**entred **D**esign
UDP	**U**ser **D**atagram **P**rotocol

Dedicated to my father, Mr Leonard Z. Chavula

Chapter 1

Introduction

Universities and research institutions in Africa have had a history of limited and expensive interconnectivity. The continent's Internet infrastructure has for a long time not been able to meet the quality of service (QoS) required for collaborative research applications. This has been a problem for universities and the research community, as many education and collaboration oriented applications, have QoS requirements that may not easily be met with the commodity Internet. Scientific research facilities, such as the Square-Kilometre-Array (SKA) Telescope, the Large Hadron Collider (LHC) and other astronomical observatories generate, at high speed, huge amounts of data that needs to be exchanged among research centres around the world. Universities also aim to share a variety of digital resources. Some applications in regard include: real-time remote access and manipulation of telescopes in remote locations by scientists from their home institutions; provisions for remote campus through video conferencing, as is the case with Internet2 connecting Georgetown University to Qatar, New York University to Abu Dhabi and Shanghai, Dartmouth College to Chinese sites; the Indian Institutes of Technology in India expanding reach with the 'Country-wide Classroom' on the National Knowledge Network (NKN) of India; provision of high-speed access for multiple simultaneous users to e-learning courses, such as the Massive Open Online Courses (MOOCs) and Khan Academy (Foley, 2016). In addition, some universities aim to provide virtual libraries and online digital repositories and, for collaborating universities, they aim to make such libraries available to students who are located in other distant institutions. Many universities are also making efforts to provide E-learning platforms (Badger et al., 2013; Perez-Gonzalez, Soto-Acosta, and Popa, 2014; Evans, Burritt, and Guthrie, 2013), availing the teaching and learning material beyond classroom time and campus boundaries. Other universities are beginning to run virtual environments, allowing students in different geographical locations (cross-border) to remotely attend and participate in live lectures. In Europe, for example, the concept has been extended to allow students to create their own curriculum across universities in different countries (Van Dusen, 2014).

Many inter-NREN projects require low latency communication. One such project is the Middleware for Collaborative Applications and Global Virtual Communities (MAGIC) (Janz et al., 2016) seeks to establish a set of agreements for Europe, Latin America and the

Caribbean, Africa, and Asia, with the aim of consolidating middleware necessary for the establishment, among others, of real-time applications for international and inter-continental research groups (Foley, 2016). However, a critical Internet performance challenge for African universities has been the very high latencies for data exchanged between universities and research institutions across the continent. Latency, usually measured as the Round Trip Time (RTT), is the time it takes for a data packet to move from source to the destination, and for the acknowledgement packet to be received by the sender. Latency is an important characteristic of Internet connectivity as it affects the performance and responsiveness of Internet applications, especially real-time and interactive ones. High latencies are particularly problematic for education and collaborative research oriented applications such as real-time remote lecturing, or sharing of virtual computer resources. High latencies make it difficult for research communities to make use of Internet-based collaborative tools such as real-time remote lectures or sharing of virtual resources such as computer processors. Universities have also been confronted with the challenge of limited Internet bandwidth, while at the same time the demand has been increasing due to growing traffic volumes from multimedia and other bandwidth intensive online applications. This situation has resulted in congested Internet links and service delivery that does not meet quality of service requirements.

In order to facilitate better research collaboration, National Research and Education Networks (NRENs) have become a prominent means of interconnecting education and research communities (Fryer, 2014). NRENs have been conceived to provide specialized core network infrastructure dedicated to linking research and education institutions for efficient exchange of data. NRENs refer to both the physical communications network operated for and by the education and research community of various countries, as well as the organizations that operate such networks (Foley, 2016). Such organisation are constituted either as consortia of members, dedicated agencies, companies, NGOs, or other type of bodies. With the emergence of NRENs, it has become standard practice for universities and research centres to interconnect directly with one another and to exchange research and education oriented traffic using their network infrastructure, separate from the commercial or 'commodity' Internet (Foley, 2016). In terms of performance, NRENs aim to reduce latencies between educational institutions, promote bandwidth sharing and improve traffic engineering (Fryer, 2014). Their main goal is to meet the quality of service requirements of educational and research applications through provision of dedicated network backbones that interconnect education and research institutions, as well as through collaborative bandwidth management and peering agreements (Andronico et al., 2011).

NRENs are organised based on a Federated Networks Architecture model (Berman et al., 2014), designed to enable sharing of resources among multiple independent networks with the aim of optimising the use of networked resources, as well as to improve the quality of network-based services and reduce costs. Some of the well known NRENs include

Internet2 in the United States,Janet in the UK, RENATER in France, Rede Nacional de Ensino e Pesquisa (RNP) in Brazil, and CERNET in China. Following the federated networks model, inter-NREN backbones have been built to provide transit services specifically for research and eduction traffic that is generated from research applications, demonstrations and experiments between universities (Li et al., 2010). Apart from direct exchange of research traffic through PoPs, NRENs also interconnect with external network at Open exchange points (OXPs) (Ventre et al., 2017). OXPs are similar to the standard Internet exchange points (IXPs), and are used to enable NRENs to exchange 'commodity' traffic with external non-NREN networks. The main difference between IXPs and OXPs is that while IXPs provide a switched Layer2 infrastructure for multiple participants to exchange traffic through public BGP peering, OXP customers (NRENs or external participants) are able to request the establishment of Layer2 circuits between endpoints, and these circuits can be used for various purposes, including for setting up private BGP sessions. NRENs are thus interconnected to regional backbones, such as GÉANT in Europe, RedCLARA in Latin America.

In the Eastern and Southern Africa region, the UbuntuNet Alliance (Alliance, 2014) has been formed and has led the deployment of increased cross-border fiber optic cables, improving the interconnection between the member NRENs in the region. Recently, the Alliance, through the AfricaConnect Project (Foley, 2016), embarked on building high-speed inter-NREN interconnection through terrestrial fibre optic cables, complementing the East African Marine Cable System (TEAMS) and the East Africa Submarine Cable System (EASSY). The project has resulted in an improved physical interconnection in the UbuntuNet Alliance comprising multiple intra-continental and transcontinental links. Similarly, the West and Central African Research and Education Network (WACREN) has been established and is facilitating deployment of Internet infrastructure and interconnection of NRENs in 22 countries in the West and Central Africa, but as of July 2016, only includes Benin, Burkina Faso, Cameroon, Cote d'Ivoire, Gabon, Ghana, Mali, Niger, Nigeria, Senegal, and Togo (Barry, 2013; Foley, 2016). In North Africa, the Arab States Research and Education Network (ASREN) was launched in December 2010 to link Mediterranean countries to GÉANT, the European Research and Education Network. The African countries that are part of ASREN are Mauritania, Morocco, Algeria, Tunisia, Egypt, Sudan, Djibouti, Somalia, and Comoros (Foley, 2016). Interconnection efforts by these regional NREN alliances has resulted in multiple inter-NREN interconnection points, which has opened up new opportunities for traffic engineering.

Prior to the establishment of NRENs and regional inter-NRENs infrastructure in Africa, the level of interconnectivity among the continent's education and research institutions was severely disjointed. With many African universities obtaining Internet connectivity from Internet Service Providers (ISPs) that do not peer locally among themselves (Steiner et al., 2005; Barry et al., 2010), traffic exchanged among the research and education institutions

tended to traverse higher tier transit providers through global Internet Exchange Points (IXP) and long high latency intercontinental links. One reason for lack of peering is that some ISPs are not physically connected to any IXP. Physical connectivity to an IXP entails a significant cost for small and medium sized ISPs, which are common in many developing countries. Furthermore, due to there being a very small amount of locally hosted content, there is little incentive for investments to connect to IXPs. In some cases, ISPs are physically present at IXPs, but the motivation for peering is inhibited by conflicting business interests. As a result of the disjointed topology, most of the interconnection for research institutions in sub-Sahara Africa has been through expensive intercontinental links, mostly through Europe and North America. This made it easier for African scientific communities to have Internet based interactions with northern hemisphere countries, than among themselves. With the establishment of NRENs, many universities are now accessing the Internet and interconnecting through dedicated NRENs Internet infrastructure. When NRENs connect directly through their own infrastructure, it becomes easier to perform custom traffic engineering between institutions. On the other hand, where universities still connect to the Internet through commercial ISPs, apart from the high cost of bandwidth, is becomes difficult for Africa's universities and research communities to leverage the Internet for collaboration and resource sharing.

1.1 Motivation and Problem Statement

Real-time collaborative inter-university applications require low latency, sufficient bandwidth and network availability for effective performance. While networks in other parts of the world have been able to reduce end-to-end latencies using appropriate Border Gateway Protocol (BGP) peering and IXPs, circuitous routing has been persistent in Africa. One of the key challenges in this regard has been the low level of local peering among Africa's ISPs. Recent work on the African Internet topology (Gupta et al., 2014) showed that most African ISPs do not peer among themselves at national or regional level, but rather at larger European IXPs such as those in London and Amsterdam. As a consequence, traffic exchanged between Africa's Internet users is routed outside the continent, resulting in high latencies. In the case of UbuntuNet Alliance's topology, the UbuntuNet, the circuitous routing is partly a consequence of the Alliance's interconnection through GÉANT in London and Amsterdam, as well as its peering with global transit providers through European Internet Exchange Points. The UbuntuNet has historically interconnected in Europe for inter-NREN traffic exchange, resulting in significant 'tromboning' (Edmundson et al., 2016), a practice where networks exchange domestic traffic through remote interconnection points. This is further confounded by the fact that the level of peering and interconnectivity among Africa's ISPs is low (Gupta et al., 2014).

The lack of local peering amongst Africa's networks means that the standard BGP-based traffic engineering solutions can not be as effective in Africa. This motivates the investigation and testing of other models and mechanisms for implementing inter-domain traffic engineering in Africa. Furthermore, standard approaches for influencing selection of paths across multiple domains have relied on manipulating the BGP, but these approaches have been unreliable and inefficient (Saucez et al., 2008). A key problem is that BGP is an inherently single path system where alternate routes are not disseminated by routers. Each BGP router selects and advertises only the single best path (Xu and Rexford, 2006) to its neighbours. By sending only a path (default route) to neighbouring domains, the multipath diversity available in an internetwork is diminished. Furthermore, it is not always the case that the default BGP routes offer the best performance. He and Rexford (2008) showed that, in multipath environments, better alternative paths with lower loss rate and delay are available between 30% and 80% of the time.

Through the AfricaConnect project, the UbuntuNet has been greatly improved, with a number of points of presence (PoPs) being established in the Alliance's members countries in eastern and southern Africa. These PoPs have been interconnected by broadband cross-border links, enabling the UbuntuNet network to keep more African traffic in Africa, thereby reducing end-to-end delays. However, the fact that the major inter-NREN connection points are located in Europe is particularly disadvantageous for NRENs located in southern Africa because they are geographically the furthest from the interconnection points. This geographical length of the physical links results in very high latencies for traffic exchanged between the Alliance's NRENs through these interconnection points. For instance, the fibre optic cable that runs from the Southern African tip in Cape Town, to London, is about 15,000 km long. This implies that traffic exchanged between Southern African networks, but routed through London, covers a unidirectional distance of about 30,000 km. Of course, these inter-continental links have high bandwidth capacity and are very useful for high volume traffic exchanges between NRENs. The improved physical topology of the Alliance has opened up new possibilities for traffic engineering in NRENs as there are now opportunities for dynamic selection of end-to-end paths based on QoS requirements. In other words, the UbuntuNet now has the opportunity to balance the routing of inter-NREN traffic between the high-capacity high-latency inter-continental cables and the lower-bandwidth low-latency intra-continental cables.

For Africa's research networks to reduce high latencies that are caused by circuitous routing, and to enhance the utility of the growing fibre optic cable system across the continent, there is need for protocols and traffic engineering frameworks that can allow dynamic discovery and configuration of low latency paths. Also, given the multipath topology, the ability to perform multipath routing has the potential to offer performance enhancements and cost savings for NRENs. It may be necessary, therefore, that the interconnected NRENs

have the capability for discovering alternate paths and optimally redirecting traffic.

The Locator/Identifier Separation Protocol (LISP) (Li, Wang, and Wang, 2011), coupled with Software Defined Networking (SDN) (Raghavan et al., 2012), provide new opportunities for dynamic and flexible traffic engineering. In particular, LISP (Li, Wang, and Wang, 2011) provides new opportunities to allow networks to announce, through a mapping server, multiple gateways, thereby making alternate routes more visible and accessible (Secci, Liu, and Jabbari, 2013; Saucez et al., 2012). NRENs can use the LISP protocol to retrieve from a mapping server multiple locators through which to reach each other. The availability of multiple remote gateways for the same destination enhances path diversity and allows source networks to forward traffic to a particular destination through multiple remote gateways.

Software Defined Networking provides new opportunities for dynamic and remote configuration of traffic forwarding paths across remote switches. By using Openflow's (Rothenberg et al., 2012) ability to customize packet forwarding rules, and by appropriately matching packets into flows using header tags, it is possible to dynamically configure traffic engineering rules. This provides a good opportunity for solving the problem of circuitous routes, through effective traffic engineering techniques. In the context of pan-African NRENs, this could entail optimising usage of inter-continental links and cross-border terrestrial links, taking into consideration factors such as QoS requirements of network applications, provisioning on the links, as well as cost of data transmission. With this capability, a group of NRENs can also jointly form traffic engineering strategies specifically for certain applications of common interest, e.g. inter-university video streaming, or access to e-library sites within the domain.

Both SDN and LISP make use of centralized topology managers (network controller and mapping database, respectively) that have the ability to obtain knowledge of the structure of the entire topology. This allows for the collection and analysis of global network performance statistics, which can be used for optimal path selection using data driven approaches. In particular, Reinforcement Learning (Xu, Zuo, and Huang, 2014) can be applied to such topology data and performance metrics to achieve dynamic of adjustment of path selection rules.

1.2 Research Questions and Methodology

The main aim of the research was to consider *"how African Research and Education Networks' logical topology can be improved to promote the exchange of knowledge and collaboration among research institutions in Africa"*. This thesis attempts to address this by focusing on the UbuntuNet Alliance in two steps: firstly to map logical topology of the UbuntuNet; and secondly,

to use the discovered topology information to devise and evaluate traffic engineering strategies that can be used to optimize routing and reduce latencies and maximize throughput in federated inter-NREN topologies.

This research therefore had two main phases. The first phase involved mapping the logical topology of the UbuntuNet NRENs, and to investigate how traffic is routed between Africa's research and education institutions. Within the aspect of mapping the topology, a further aim was to devise an efficient mechanism for running active topology measurements, as well as devising an effective topology visualization tool. The second phase involved the design of a traffic engineering mechanism that utilizes LISP, SDN and Reinforcement Learning to discover and configure better traffic paths. The overall aim was to reduce usage of inter-continental links for African inter-NREN traffic exchange, and in the process, reduce end-to-end latencies.

The topology mapping exercise included all NRENs in the UbuntuNet Alliance region, including those that were not physically connected to the Alliance's network at the time of study. However, traffic engineering emulations assumed a situation whereby all member NRENs would physically be connected to the UbuntuNet.

1.2.1 Mapping the UbuntuNet's Logical Topology

The first aspect of the research aimed to investigate the network topology of the UbuntuNet Alliance (the UbuntuNet). This was necessary because the task of designing cross-border interconnectivity and performing optimal traffic engineering requires a good knowledge of the logical inter-NREN topology. Although there exists information on the physical topology of the African Internet, especially the terrestrial and undersea fibre optic cable networks [1], not much work has been done to map the logical topology of the African Internet, especially the research and education networks. The aim, therefore, was to design and build a topology measurement infrastructure that would automatically and continually map the internetwork of UbuntuNet NRENs. Such data could be used to optimize traffic exchange among the NRENs.

The key objective of this phase was to obtain a mapping of the logical topology of the African research and education Internet topology, with a specific focus on the UbuntuNet Alliance member NRENs. The study sampled research and education institutions within the UbuntuNet Alliance area, without consideration to whether such institutions were physically connected to UbuntuNet at the time of study. A secondary objective was to analyse performance of traffic that uses inter-continental links versus traffic that is exchanged within the continent. The investigation further addressed the question of whether the UbuntuNet

[1] https://afterfibre.nsrc.org/

topology could efficiently and reliably be discovered using public network measurement infrastructure and, secondly, whether building an interactive visualization tool for the topology could effectively and accurately inform NREN users about the structure of the topology.

Specifically, the topology mapping phase focused on these questions:

1. How can one efficiently discover the topology of the UbuntuNet NRENs?
2. What is the PoP level and AS level inter-NREN topology of the African NREN?
3. What are the performance differences for traffic routed via inter-continental undersea cables and the traffic routed within Africa's terrestrial fibre optic cables?
4. Can interactive geo-spatial visualisation effectively and accurately communicate the physical and logical topology of UbuntuNet NRENs and routes to NREN stakeholders?

To address these questions, topology discovery experiments were conducted to gain insight into the logical structure of the UbuntuNet Alliance. In the course of mapping the topology,there was also need to evaluate a mechanism for efficient topology discovery using the Ripe Atlas platform (Question 1). The first main task was to map the logical topology and performance of the UbuntuNet Alliance (Questions 2 and Question 3). Lastly, an interactive topology geo-spatial visualisation tool was implemented and evaluated (Question 4).

Logical UbuntuNet Topology

Internet topology discovery measurements were undertaken to characterize the level of interconnectivity among Africa's research and education networks. This was also aimed at evaluating performance of traffic exchange in terms of latencies.

In this study, active network topology discovery techniques (Traceroute and Ping tools) were used to characterize performance of traffic originating in Africa and destined for African research and education institutions. Active measurements were also conducted to obtain a logical interconnectivity map of the Africa's universities. Measurements were conducted from five Africa based vantage points that are part of CAIDA's Internet measurement platform - Archipelago (Hyun, 2006). Archipelago's five vantage points in Africa are located in Morocco, Gambia, Senegal, South Africa and Rwanda. Of interest from each of the measurements was the round-trip times for traffic flows from these vantage points to Africa based destinations, as well as the geo-location of the IP hops traversed by the traffic. This helped to characterize the level of logical interconnectivity and peering among Africa's NRENs. Another interest was to analyse latency for traffic routed through inter-continental links in comparison to traffic routed within the continent.

Efficient Topology Discovery

This experiment was aimed at investigating how topology data could be collected reliably and efficiently for the purpose of discovering the logical interconnectivity of the UbuntuNet NRENs. This was motivated with a consideration that active topology measurements are costly to networks in the sense that they introduce additional traffic. To limit flow of probing packets, topology measurement infrastructures, such as Ripe Atlas, assign a cost and limit the number and/or rate of measurement packets that can be sent. Furthermore, network probing is generally blocked by routers, which entails that it may require more measurements to evaluate all network paths. Where probe packets are allowed, the presence of multiple alternate paths and load balancing also render it difficult to evaluate all the alternate paths.

The key objective therefore was to implement a reliable and efficient mechanism for collecting traceroute data for the discovery of the UbuntuNet Alliance's topology. A distributed network probing method was used. A number of topology measurement platforms are in existence, including Archipelago, DIMES, iPlane and RIPE Atlas. Unfortunately, many of them have very few vantage points within the African continent, let alone inside the NRENs.

For this study, 14 Ripe Atlas vantage points from five NRENs within the UbuntuNet Alliance were used.

Topology Visualisation

This experiment was designed to test the effectiveness of a geospatial visualisation in helping NREN stakeholders to identify topology problems and understand paths taken by traffic as it traverses between NRENs (continental vs intercontinental). A geospatial visualisation was designed as a graphical representation of the network topology of NRENs.

The experiment followed a User Centred Design methodology and evaluated, through usability tests, the effectiveness and accuracy of the visualisation at communicating the network topology (physical and logical) of UbuntuNet NRENs. Effectiveness and accuracy were evaluated by checking the correctness of responses to visual queries. These queries related to identification of network links, geographic location of traffic source, destination and intermediate hops, as well as routes at country and continental level.

1.2.2 UbuntuNet Traffic Engineering

The second aspect of the study was aimed at investigating optimal traffic engineering strategies for UbuntuNet inter-NREN communication. This thesis proposes that the use of dynamic path selection, where NRENs cooperate to dynamically reconfigure best end-to-end paths, can minimize circuitous routing and help reduce latencies between UbuntuNet Alliance NRENs.

The objective of this research was to propose and evaluate optimal traffic engineering using Software Defined Networking and LISP. This research explored a mechanism for selection of optimal interconnection points, first by ranking remote LISP gateways based on end-to-end latencies and, secondly, through the use of Reinforcement Learning (Xu, Zuo, and Huang, 2014) to achieve dynamic adjustment of forwarding rules in a LISP/SDN based topology. The approach made use of mechanisms for active and passive collection of network performance and statistical data at each node in the topology. Each interconnection point switch had a Reinforcement Learning module to continually evaluate packet forwarding decisions and gather performance data for every path. In the framework, a network controller maintains a list of routers/switches in the core topology and facilitates setting up of paths between NRENs.

The specific questions investigated with regards to the traffic engineering frameworks were as follow:

5. To what extent can LISP and SDN support optimal interconnectivity among the UbuntuNet NRENs?

6. To what extent can Reinforcement Learning help optimize inter-NREN traffic engineering across multiple interconnection points?

To address these questions, traffic engineering experiments were designed to investigate the utility for implementing SDN, LISP and Reinforcement Learning for dynamic discovery and configuration of low latency inter-NREN paths. The approach was to monitor the network performance, and to use network metrics to guide path selection.

The experiments were conducted using a network emulation approach, in which real elements of a physical network system, such as end hosts and protocol implementations, are combined with synthetic/simulated elements such as the network links (Quereilhac et al., 2011). The topologies were built in an SDN emulator called Mininet (Heller et al., 2012).

SDN/LISP Traffic Engineering

To address Research Question 5, the first traffic engineering experiment framework (described in more detail in Chapter 5) was designed, comprising an SDN topology and LISP gateways. The framework imagines a scenario in which traffic source gateways have the ability to select the destination's ingress gateway based on metrics of the edge-to-edge path. The network borders are implemented using LISP, such that gateways are able to learn a destination's multiple gateways from a LISP mapping system. The traffic engineering mechanism implemented a dynamic latency-based ranking of network gateways such that routes were selected based on the latency between the source and destination gateways. The core

topology was based on the SDN architecture, where a network controller is used for setting up end-to-end paths between a pair of selected source and destination gateways.

The evaluation analysed the network performance in networks that employ the gateway ranking mechanism, comparing such performance against the default LISP system, in which forwarding paths are selected based on the destinations' announced locator priorities.

With regards to Research Question 6, a second set of experiments, presented in Chapter 6, evaluated how Reinforcement Learning could be employed in an SDN/LISP network of the UbuntuNet Alliance. The emulated UbuntuNet core topology was made up of SDN switches, cross-border links that interconnect the NRENs, as well as inter-continental links that connect to NRENs in Europe. The NRENs' gateways were LISP routes. The topology's state information, comprising link performance and utilization, was applied to a reinforcement learning algorithm to determine optimal routes across the topology. The evaluation for this experiment measured the QoS in terms of end-to-end latency, throughput, jitter, and packet loss, for the inter-NREN traffic.

1.3 Structure of Thesis

This chapter has introduced the main problem that this thesis attempts to address. To this end, this chapter has provided the motivation and problem statement of the thesis, and has described research questions and methodology.

Chapter 2 introduces the technologies employed in this thesis. More specifically, (Section 2.2) discusses the techniques that are used for mapping Internet topologies, highlighting active and distributed methods. The chapter also gives an introduction traffic engineering technologies used in this thesis, including Software Defined Networking (Section 2.4), Locator/Identifier Separation Protocol (Section 2.3), and Reinforcement Learning (Section 2.5).

Chapter 3 provides a review of existing literature pertaining to Africa's Internet topology, as well as existing traffic engineering mechanisms.

Chapter 4 focuses on Internet measurement exercises that were undertaken to map the logical topology of the African research and education networks. The chapter also presents results of topology mapping exercises carried out using CAIDA's Archipelago and Ripe Atlas Internet measurement platforms. Also presented in this chapter is an interactive topology visualization tool, as well as results of usability tests that were carried out to study the effectiveness of the tool in helping users to understand the topology.

Chapter 5 presents a traffic engineering framework and experiments that were designed evaluate the extent to which SDN and LISP could help improve performance of Africa's research and education networks through dynamic configuration of lower latency paths. The SDN/LISP traffic engineering framework presented in Chapter 5 relies on source networks to dynamically rank destination gateways based on end-to-end latencies.

Chapter 6 builds on the work in Chapter 5 and employs Reinforcement Learning in an SDN/LISP traffic engineering framework. Chapter 6 also discusses how an emulated topology of the UbuntuNet Alliance was built for the purpose of evaluating the traffic engineering solution.

A summary of the research findings and contributions of this thesis is presented in Chapter 7.

Chapter 2

Background

This chapter provides a background to the UbuntuNet and the main technologies employed in this thesis. The first section (Section 2.1) introduces UbuntuNet network, highlighting the Alliance's history and its current topology. Section 2.2 looks at techniques used for mapping Internet topologies, focusing on active and distributed topology discovery techniques. Thereafter, the chapter provides a background to Software Defined Networking (Section 2.4), Locator/Identifier Separation Protocol (Section 2.3), and Reinforcement Learning (Section 2.5).

2.1 The UbuntuNet Alliance

The UbuntuNet Alliance is the regional research and eduction network (RREN) comprising NRENs in Eastern and Southern Africa. It is both an association of NRENs in the region, as well as a data network interconnecting the member NRENs. The Alliance was conceptualized in 2004 [1] following the then growing terrestrial fibre cable network, as well as the then ongoing deployment of the Eastern Africa Submarine Cable System (EASSy) along the east and south coast of Africa. The alliance was formally established in 2006 and was initially incorporated in the Netherlands as a not-for-profit association of five founding NREN member countries: South Africa, Malawi, Kenya, Mozambique, and Tanzania. It was later registered as a Trust in Malawi. The Alliance was founded with the aim of fostering development and interconnectivity of NRENs in the region, and to create an enabling environment for member NRENs to have sufficient and affordable connection to the international research community (Banda, Pehrson, and Jensen, 2007; Tusubira, 2009). The UbuntuNet Alliance is, as of 2016, an association of fifteen NRENs:include Eb@le (Democratic Republic of Congo); EthERNet (Ethiopia); iRENALA (Madagascar); KENET (Kenya); MAREN (Malawi); MoRENet (Mozambique); RENU (Uganda); RwEdNet (Rwanda); SomaliREN (Somalia); SudREN (Sudan); TENET (South Africa); TERNET (Tanzania); XNet (Namibia); ZAMREN (Zambia) and BERNET (Burundi).

[1] http://www.internet2.edu/presentations/spring08/20080421-ubuntunet-tusubira.pdf

The topology of the UbuntuNet Alliance was designed to comprise of three clusters for the East, the South, and the West sub-regions. The Eastern cluster was designed as comprising countries that could connect to a landing point on the east cost of Africa, such as the EASSy cable. These countries included Kenya, Mozambique and Tanzania, as well as landlocked countries such as Botswana, Malawi, Rwanda, Uganda and Zambia. The South cluster comprised countries whose NRENs could be connected to submarine cable landing points in South Africa (Cape Town and Mtunzini). The West cluster was for countries that could be connected to landing points of the SAT-3 West Africa Cable System (WACS) other than the South African landing points.

Since 2011, the UbuntuNet Alliance has been working to improve the core topology by implementing the AfricaConnect Project, with the aim to expand the interconnection among its member NRENs through the use of terrestrial network facilities. The project has involved establishment of Points of Presence (PoPs) in major cities in the region - notably in Mtunzini, Maputo, Dar es Salaam, Nairobi, Kampala and Kigali, and interconnecting them using cross-border fibre optic cable system to create a regional research network.

According to a technical report by the UbuntuNet Alliance (UbuntuNetAlliance, 2016), as of 2016, the UbuntuNet backbone topology (Figure 2.1) was made up of ten points of presence (PoPs); eight located within the Alliance region and two in Europe (London and Amsterdam). Transcontinental links have been established between Nairobi and the UbuntuNet Alliance PoP in Amsterdam, as well as from Cape Town to London. In total, the topology has an aggregate transit capacity of 2.18 Gbps to its Europe-based PoPs. The intra-Africa topology currently (2016) serves seven NRENs: TENET (South Africa), MoRENet (Mozambique), TERNET (Tanzania), KENET (Kenya), RENU (Uganda) and RwEdNet (Rwanda). The UbuntuNet topology is technically managed by TENET, the Tertiary Education and Research Network of South Africa. TENET also manages the South African National Research Network (SANReN). TENET connects to the UbuntuNet topology with a capacity of 10 Gbps through the Alliance's PoPs in Mtunzini, London and Amsterdam. TENET also has its international capacity through a 10 Gbps link to London on the SEACOM cable, 20 Gbps on the WACS cable system[2]. Part of TENETs international capacity from Cape Town to London, a single STM-4 (622 Mbps) and 2 STM-1 links (2 X 155 Mbps), is used for UbuntuNet traffic. TENET also has the 10 Gbps SANReN backbone through Johannesburg, Pretoria, Cape Town and Durban.

On the other hand, MoRENet (Maputo,Mozambique) and TERNET (Dar es Salaam, Tanzania) each connect to London at 155 Mbps via STM-1 circuits through the SEACOM submarine cable. Zambia's ZAMREN connects with a capacity of 622 Mbps to UbuntuNet's PoP in Lusaka, and has its international traffic routed through the Alliance's PoPs in Dar es Salaam and Cape Town. Uganda's RENU is connected to Alliance through a 370 Mbps circuit to

[2]http://www.tenet.ac.za/

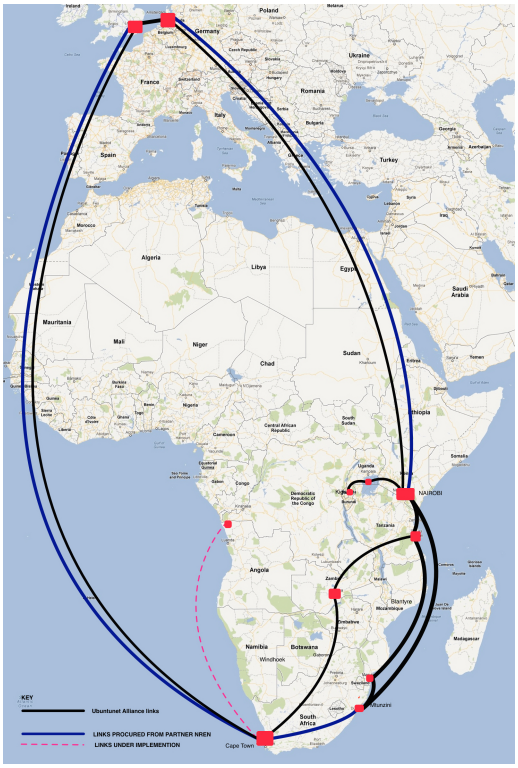


FIGURE 2.1: The UbuntuNet Alliance regional network

the Kampala PoP. Eb@le, the NREN for the Democratic Republic of Congo connects via the Cape Town PoP with a single STM-1 (155 Mbps) link from Moanda(DRC).

KENET is connected to the UbuntuNet core topology through a 155 Mpbs terminating at the Ubuntunet Nairobi PoP. Furthermore, KENET is connected through its hub in Nairobi to UbuntuNet PoPs in Amsterdam and London PoPs with an aggregate bandwidth capacity of at least 4 Gbps. This consist of two STM-4 circuits on the TEAMS submarine cable and a single STM-1 circuit on the SEACOM cable, all terminating in London, as well as an STM-16 link to the Amsterdam PoP.[3]

The UbuntuNet Alliance obtains global Internet connectivity through its peering agreements with GÉANT and other transit providers in London and Amsterdam. Through the GÉANT peering arrangement, the Alliance also obtains transit to other education and research networks. As of 2014, the UbuntuNet had established settlement-free peering with more than three hundred networks through Amsterdam and London (AMS-IX and LINX respectively) (Banda, Pehrson, and Jensen, 2007). UbuntuNet also has peering relations with commodity Internet transit providers at the London IXP (LINX), Amsterdam IXP (AMS-IX), as well as at NAPAfrica IXPs in Johannesburg and Cape Town[4].

[3]http://www.internet2.edu/presentations/spring08/20080421-ubuntunet-tusubira.pdf
[4]http://www.tenet.ac.za/about/about-tenet-1

2.2 Techniques for Internet Topology Discovery

The Internet is an interconnection of many privately managed networks known as Autonomous Systems (ASes) (Shavitt and Weinsberg, 2011). Traffic exchange among ASes is facilitated through the Border Gateway Protocol (BGP), a single path routing system that conveys AS-level paths between domains, enabling them to interconnect and exchange traffic. Any two ASes can exchange traffic if they have some direct logical connection between them, or if they both have access to other higher level providers that can transit traffic between them (Ahmad and Guha, 2011). Due to this hierarchical structure of the Internet, traffic whose source and destination networks are geographically close may sometimes have to traverse circuitous remote links in search of interconnecting paths - a phenomenon know as tromboning (Obar and Clement, 2013). To obtain a clearer understanding of a logical internetwork, topology discovery techniques are used to obtain data for network visualization.

Internet topology discovery techniques can largely be grouped into two: passive techniques; and active techniques (Shavitt and Weinsberg, 2011). Passive methods take place in the control plane where network monitors, such as BGP monitors, collect control information and statistical data based on traffic flows over the topology links (Motamedi, Rejaie, and Willinger, 2015). Passive network measurements involve the analysis of such management data to infer the network performance as well as topological structure in terms of logical relationships among networks.

2.2.1 Active Topology Measurements

Active measurements take place in the data plane and along the paths traversed by packets, with the aim of monitoring reachability and performance of the Internet paths (Allalouf, Kaplan, and Shavitt, 2009). Active measurements rely on sending specially crafted packets into the network with the aim of soliciting topology information. These techniques attempt to exploit network management protocols such as SNMP and ICMP to solicit responses from a set of network destinations, and then analyse such responses to infer topological characteristics such as route paths, round-trip-times (RTTs) and packet loss. Active measurements are often used for collecting data to discover topologies whereas passive measurements are used for traffic profiling.

Some of the most widely used active probing tools are Ping and Traceroute (Branigan et al., 2001). Ping is a network utility tool that is used to test the reachability of hosts in IP networks. The tool sends Internet Control Message Protocol (ICMP) Echo Request packets to a target host and waits for an ICMP Echo Reply. Ping is thus used to measure the round-trip time for a network packet to travel from a source host to a destination host, and for a response to get back to the source. Ping reports the statistical summary, which includes the minimum, maximum, the mean and standard deviation of the mean round-trip times, as

well as the packet loss. On the other hand, Traceroute is a network tool used for discovering IP paths between a host and a destination, and is the de-facto method for discovering Internet topologies. The tool is used to gather information about a topology and how traffic is routed between and within networks. Traceroute works by sending IP packets with increasing time-to-live (TTL) values, in such a way that packets continually expire on their way and cause routers to respond with ICMP time-exceeded messages. Through the responses, the routers reveal the paths towards a destination.

There are three main variants of Traceroute using different protocols (Branigan et al., 2001; Luckie, 2010). The standard Traceroute uses User Datagram Protocol (UDP) probes and receives Internet Control Message Protocol (ICMP) responses. The second variant uses only ICMP, sending ICMP echo requests and receiving ICMP echo replies. These two variants do encounter errors if a router does not have the ICMP protocol enabled or if a router employs ICMP rate limiting (Branigan et al., 2001). The third variant makes use of Transport Control Protocol (TCP) packets and sends TCP SYN packets to try to get past the most common firewall filters (Donnet and Friedman, 2007).

One further variant of Traceroute is Paris Traceroute (Augustin, Friedman, and Teixeira, 2007a). Paris Traceroute is made up of ICMP-Paris and UDP-Paris (Luckie, Hyun, and Huffaker, 2008) and helps in discovering alternate paths. It avoids missing links and nodes as well as false links which could appear because of load balancing. This is done by controlling and varying the packet header contents when conducting Traceroute measurements.

In terms of granularity, Internet topology information can be collected on four different levels – Internet Protocol (IP) Interface, Router, Point of Presence (PoP) and Autonomous System (AS) levels (Donnet and Friedman, 2007; Motamedi, Rejaie, and Willinger, 2015; Mao et al., 2003). A Point of Presence is a collection of routers belonging to one AS (Mao et al., 2003). There are three main methods to collect data at the PoP level. Firstly, data can be aggregated from active measurements to identify PoPs. Secondly, end-to-end delays obtained from active measurements can be used to infer co-located routers. Finally, information can be retrieved ISPs' published topology data. Although using published information method may provide more accurate data than the active measurements, the technique is not always reliable as the information could be outdated (Mao et al., 2003). Topology discovery at the PoP level provides information and limitations about latencies between PoPs. This helps with understanding the geographical properties of Internet paths, such as where ASes can connect and the coverage of ASes (Mao et al., 2003). Autonomous Systems (ASes) are privately managed networks, which are all interconnected making up the Internet (Donnet and Friedman, 2007). ASes are identified by a unique 16-bit AS number. To collect information for the AS level, data is collected from BGP tables, traceroute measurements and Internet Routing Registries (IRR) (Mao et al., 2003). NRENs mainly operate at the AS level but could constitute several PoPs.

2.2.2 Distributed Topology Discovery

Distributed topology discovery involves the sending of probing packets from a diverse set of locations and networks. Network devices that are the origin of probe packets are known as vantage points (Shavitt and Weinsberg, 2011). In order to generate more accurate Internet maps, distributed topology discovery systems are designed to have numerous vantage points in a diverse set of locations and networks. Diversity in vantage points' distribution has been shown to increase the probability of discovering and measuring more links between networks (Shavitt and Weinsberg, 2011; Shavitt and Shir, 2005). For this reason, in addition to stand-alone Internet measurement tools, a number of distributed network probing infrastructures are in existence. Notable measurement infrastructures include the Archipelago, PingER (Ping End-to-end Reporting) project (Matthews and Cottrell, 2000), Speed Checker, DIMES (Shavitt and Shir, 2005), iPlane (Madhyastha et al., 2006) and RIPE Atlas (Atlas, 2015). These infrastructures have significantly reduced the efforts required by Internet researchers to implement and run topology measurements from a wide rage of vantage points (Shavitt and Weinsberg, 2011; Hyun, 2006).

RIPE (Réseaux IP Européens) Atlas (Atlas, 2015) is a platform that makes use of thousands of small USB-powered hardware devices around the world, known as probes, to measure Internet connectivity and reachability. These probes are attached to host networks and are used to conduct measurements, such as Ping, Traceroute, DNS and SSLcert, and relays the measurement data to the RIPE Network Coordination Centre (NCC) servers. This data is aggregated with data collected from other RIPE Atlas probes. The RIPE Atlas platform is advantageous for researching the topology of African NRENs. Using RIPE's API, custom measurements can be sent from a probe to any IP address, allowing the collection of data that is required to study routes between African NRENs.

The Archipelago (Ark) platform (Hyun, 2006) is an active measurement infrastructure for Internet topology discovery, operated by CAIDA. The infrastructure is based on a network measurement tool called Scamper (Luckie, 2010), which implements Paris-traceroute (Augustin, Friedman, and Teixeira, 2007a), a variant of traceroute based on Multi-path Discovery Algorithm (MDA) (Augustin, Friedman, and Teixeira, 2007b; Augustin, Friedman, and Teixeira, 2011). As of December 2016, there were 170 Ark monitors distributed in 59 unique countries [5]. Of the 170 monitors, 14 are located in African countries.

The PingER project (Matthews and Cottrell, 2000) is infrastructure designed to monitor network performance between laboratories, universities and institutes collaborating on high energy nuclear and particle physics experiments. The resulting Internet end-to-end performance monitoring project reflects the wide geographical spread of the collaborations across a large number of research and commercial networks. Presently, the project is being used by researchers for wider Internet performance, including the measuring of global digital

[5] http://www.caida.org/projects/ark/locations/

divide. Zennaro et al. (2006) used PingER data to quantify the difference in performance between developed and developing countries.

SpeedChecker[6] is a crowd-source platform that is supported by Internet end-user devices that have a SpeedChecker client installed. The platform provides an API-based mechanism (ProbeAPI) that allows researchers to conduct different types of measurements from end-user devices towards a set of fixed servers.

2.3 Locator/Identifier Separation Protocol

The Internet is based on a mechanism where nodes' IP addresses are used as routing identities and are set according to their location in the Internet topology. In other words, a host's IP address is part of an IP range (IP prefix) that belongs to a host's network. The IP address is thus determined by and is indicative of a hosts' network. The current IP scheme is constrained in terms of IP mobility, which is a challenge for devices that need to change locations (network) but need to maintain their unique identities (IP address). Furthermore, while the prefix-based addressing scheme is necessary for Internet routing, it has nonetheless caused scalability problems, as it has led ISPs to de-aggregating their IP prefixes in order to achieve fine-grained control of packets flowing between networks. The result of this has been an inefficient traffic engineering that has also led to bloated global routing table size (Saucez et al., 2012).

The Locator/Identifier Separation Protocol (LISP) (Li, Wang, and Wang, 2011; Saucez et al., 2012) is a routing mechanism that was designed to deal with the problem of scalability due to growing routing tables in the Internet Default-Free Zone (DFZ) (Rodriguez-Natal et al., 2015). LISP separates two combined functions of IP addresses: the topological location of an Internet host; and the unique identification of a host. This was achieved by splitting the Internet address space into two: Endpoint IDentifiers (EIDs) that are only visible within domains; and LOCators (RLOCs), interdomain routers that are globally routed (Phung et al., 2014). The edge routers have their external interfaces refereed to as RLOCs, or simply locators. On the other hand, edge hosts are EIDs and have a scope limited to within the edge networks. LISP therefore makes it possible to confine the prefix de-aggregating traffic engineering techniques to the EID space, keeping the locators space stable and not prone to de-aggregation attempts by ASes (Rodriguez-Natal et al., 2015). In essence, LISP prevents edge network specific prefixes from appearing in the transit core, thereby reducing the DFZ BGP routing table size (Wang, Bi, and Wu, 2010). When a host in an edge network communicates with another host in a remote edge network, then the IP addresses of both the source and destination hosts are mapped to the corresponding edge networks' transit IP addresses, the RLOCs.

[6] http://probeapi.speedchecker.xyz/

LISP (Li, Wang, and Wang, 2011; Saucez et al., 2012) decouples the *locator* and *identifier* functions of IP address. Such a separation allows edge networks to announce multiple Internet gateways, known as Route Locators (RLOCs), and to influence the selection of incoming paths by specifying ranks and priorities for gateways. Through a mapping system, LISP allows networks to announce preferences for multiple RLOCs. By making available multiple locators for the same destination, LISP increases path diversity (Secci et al., 2011a; Secci, Liu, and Jabbari, 2013) and simplifies Internet multihoming. LISP-based networks therefore have the capability to select any of the multiple local and remote gateways for transferring traffic towards a destination network, making possible multiple paths between hosts located in remote LISP-based networks.

Through a mapping system, LISP networks can announce and specify preferences for multiple incoming gateways (RLOCs). RLOCs IP addresses are used for routing packets across the core topology. At the border gateways (locators), end hosts' EID addresses are mapped to the respective source and destination RLOCs. Outgoing packets are encapsulated and are routed through the Internet core based on addresses of the respective border LISP routers (locators). The underlying interdomain routing protocols are used to determine the paths between the remote communicating locators. At the receiving locator, the LISP encapsulated packets are decapsulated before being forwarded to the destination end host using the EID.

A number of LISP implementations[7] exist, and prominent open source ones include OpenLISP (Saucez, Iannone, and Bonaventure, 2009), LISPMob and (Cabellos et al., 2011). OpenLISP was one of the first full implementations of the LISP proposal as described in the RCF. The implementation includes a FreeBSD based version that provides LISP's control and data plane, as well as a Linux based implementation supporting only LISP's control plane. OpenLISP's control plane provides all basic control plane functions, including the Mapping Server and the Map Resolver. LISPmob is another open source implementation that was designed for LISP mobile node technology, but later included other LISP functionalities in both the control and data plane, including the locators and the mapping database. The LISPMob implementation and was recently re-branded as Open Overlay Router (OOR)[8], and now includes the LISP Mobile Node implementation for Linux, Android and OpenWrt. More recently, a new closed-source LISP implementation called *lispers.net*[9] was made available for research purposes. Lispers.net is implemented in the Python programming language and provides APIs that allow LISP hosts to interact with a mapping system.

[7] http://www.lisp4.net/implementations/
[8] http://www.openoverlayrouter.org/
[9] https://www.lispers.net/

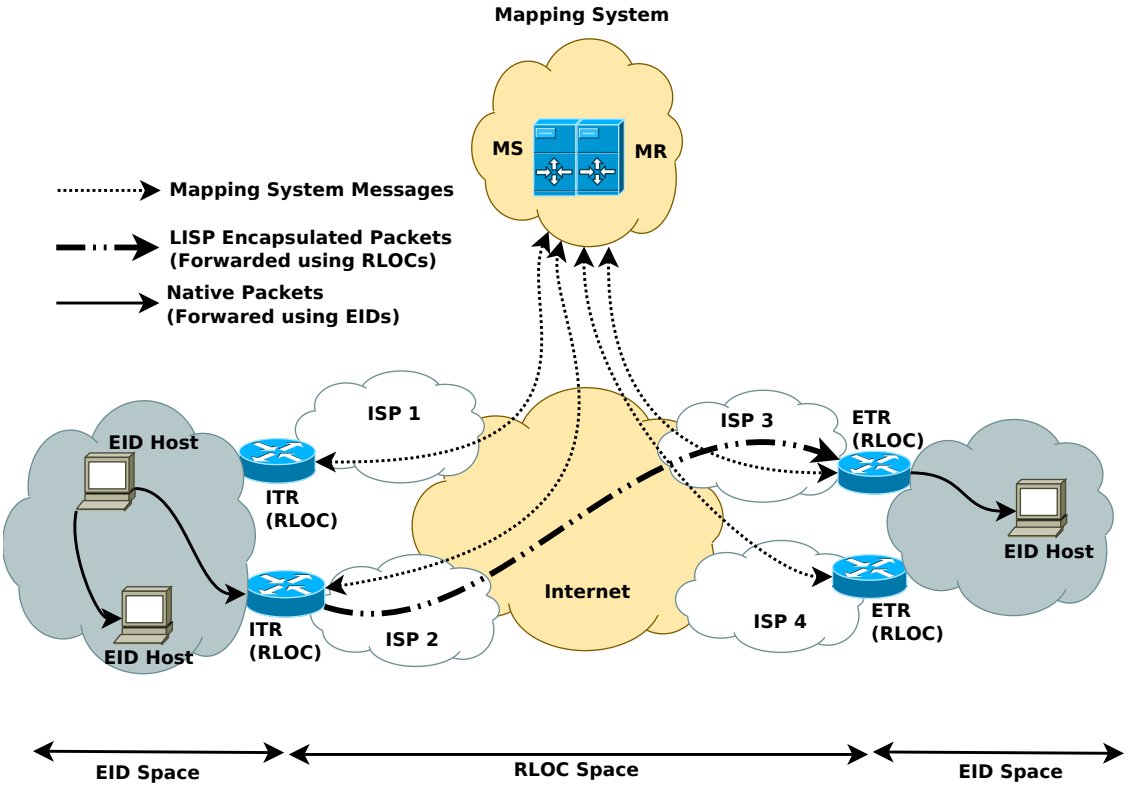


FIGURE 2.2: LISP Architecture

2.3.1 LISP Network Components

Figure 2.2 depicts the LISP components.

Endpoint IDentifier

Endpoint Identifiers (EIDs) are 32-bit or 128-bit addresses (for IPv4 and IPv6 respectively) used to identify hosts inside LISP sites, where a LISP site refers to a network administrative domain whose gateway to the wider Internet is through LISP-based gateways and whose end hosts reachable from external networks through the domain's announced RLOC addresses. The EIDs are used for communication within individual LISP sites but are not globally routed. LISP gateway routers are used for communication between hosts located in disjoint LISP sites. Just like in IPv4/IPv6, EID are aggregated into prefix sets and are assigned to specific autonomous systems.

Routing Locator

Routing locators (RLOCs) are 32-bit or 128-bit globally routed addresses that are assigned to border routers (locators) of LISP sites. The RLOC is actually the IP address of a locator, which may in fact have multiple RLOCs. In LISP parlance, the border routers/locators are called ingress tunnel routers (ITR) and egress tunnel routers (ETR), and generalized as

xTRs. Packets originating from inside a LISP network are forwarded into the global Internet through the edge locators, with the packets encapsulated with locator address.

Ingress Tunnel Router

An ingress tunnel router (ITR) is the outbound gateway for a LISP site responsible for forwarding packets from local EID hosts towards remote LISP sites and the Internet in general. Upon receiving outbound packets from a local EID client, an ITR queries the LISP mapping system for EID-to-RLOC mapping, then encapsulate each packet inside a LISP header before forwarding externally. The LISP header encodes the ITR's own globally routable RLOCs as the source address, and the destinations' RLOC as the destination address.

Egress Tunnel Router

The egress tunnel router (ETR) is the inbound gateway for a LISP site and is responsible for receiving packets from the Internet and forwarding them into its end hosts. Upon receiving a LISP encapsulated packet destined to one of the local EIDs, an ETR de-encapsulates the packet by removing LISP header, before forwarding the packet to the local EID client. Typically, the ETR and ITR functions are performed by the same router, in which case the gateway is referred to as an xTR. An ETR also has the responsibility to register its domain's EID prefix into the mapping system.

Mapping System

The Mapping System (MS) is a principal component of the LISP technology. The mapping system facilitates end-to-end packet forwarding through a process of mapping and encapsulation (Map & Encap), where packet addresses are mapped between EIDs and RLOCs through addition and removal of extra packet headers. An endpoint identifier is mapped to its home network through the querying of the LISP mapping system.

LISP employs a hierarchical database similar to the Domain Name System (DNS). The database maintains mappings between EIDs and RLOCs, and provides such information to LISP routers upon being queried. The database architecture is supported by two subcomponents: map servers and map resolvers. Map servers store EID-to-RLOC mapping information provided by ETRs during a registration process. On the other hand, map resolvers (MRs) are queried by ITRs during the encapsulation process, upon which the map resolver hierarchically queries the LISP distributed database system to find the authoritative map server responsible for the specified EID-to-RLOC mapping. The map resolver either returns a negative map-reply if the queried EID is not assigned to a LISP site, or requests the mapping from the mapping server.

2.3.2 LISP Operation

Figure 2.3 illustrates the operation of LISP. LISP operates by encapsulating EID addressed packets inside RLOC headers, in a fashion similar to NAT, the principal difference being that while private IP addresses used in NAT are required to be unique only within a network, the EIDs used in LISP are globally unique. To achieve encapsulation and de-encapsulation, LISP relies on a distributed database system that stores mapping information associating EIDs to RLOCs. The EID-to-RLOC mappings are registered into the mapping system by the respective ETRs, and each registration comprises a set of EID prefixes (block of EID addresses) associated to a list of RLOCs. Each LISP site that has an EID prefix performs an EID-to-RLOC mapping registration **(0)** by sending a map-register message to the mapping server through its ETRs. The registration message encodes EID-prefixes associated with RLOCs through which the sites are reachable.

A traffic source makes use of a LISP mapping system to discover the locators that can be used to reach a particular destination network. Any network running the LISP protocol first has to register its gateway routers (locators). Each locator is registered with a priority (0 255), and a weight (0 100) that is used to determine how traffic is load balanced when more than one locator can be used for the same destination. When sending traffic, a source network queries the LISP mapping system to obtain a list of the destination's locators.

Each mapped RLOC is assigned values for priority, weight and a reachability flag (Saucez, 2011). A priority value is used to select the RLOC that must be preferred for reaching a EID prefix if multiple RLOCs are associated with the same EID prefix. Ideally, the RLOC with the lowest priority value is supposed to be selected, although the source RLOC has the liberty to ignore the priority values and employ its own mechanism for selecting RLOCs when multiple exist. In a standard LISP operation, a source network will forward traffic through destination RLOC that has the highest priority among the registered locators. If locators have the same priority, the weight values are used to determine the percentage of traffic that has to go through each of the locators. The mapping also includes a reachability flag that indicates whether the RLOC is online.

Within a LISP network, packets are forwarded based on node EIDs. An EID host sending packets to a remote EID **(1)** host will put its EID as the source address of the packets, and the remote EID as the destination address. At the gateway locator, the outbound packets destined to an EID host located outside the home LISP network triggers a query to the LISP's mapping database. The source locator (ie the ingress transit router, ITR) performs a mapping lookup **(2)** to obtain the corresponding remote EID's locator addresses (RLOCs). In some instances, the ITR already has the mapping information stored in its local cache, in which case it proceeds with the encapsulation without querying the mapping system. When queried, the mapping server responds **(3)** with a list of locators that have been registered as

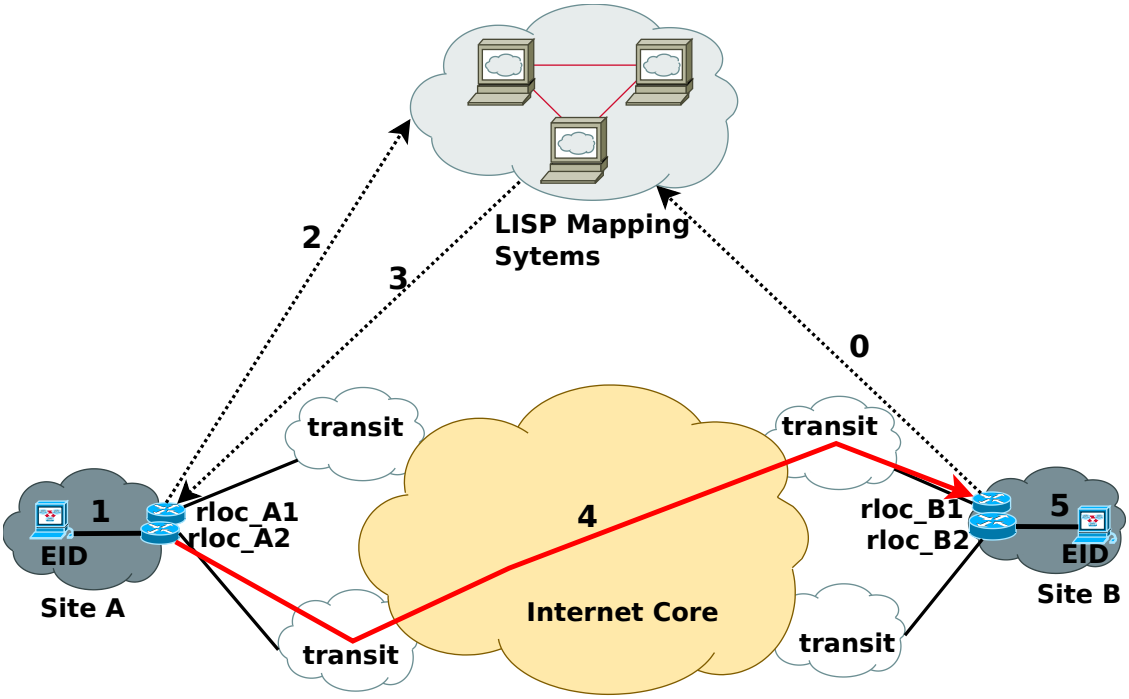


FIGURE 2.3: LISP Operation

gateways for the requested EID. The ITR also caches the obtained mapping information, which it uses for subsequent mappings until the record expires.

In standard LISP operation, each source locator simply selects a single destination locator that has highest priority value. The source locator then encapsulates the packets with a LISP header, where the source and the selected destination locator address (RLOCs) are used respectively as the packets' source and destination address. The encapsulated packets are forwarded through the Internet using the underlying interdomain routing protocols **(4)**. At the destination locator (ETR), the packets are decapsulated with the removal of the LISP header before being forwarded to the ultimate destination EID host **(5)**.

2.4 Software Defined Networking

Software Defined Networking (SDN), an emerging paradigm for network design separates a network's control plane from the forwarding plane (Raghavan et al., 2012; Rothenberg et al., 2012). This separation enables remote and dynamic configuration of forwarding tables and provides new opportunities for flexible management of Internet routing and packet forwarding (Rothenberg et al., 2012). In traditional networks, the control plane, which decides how to handle network traffic, and the data plane that implements the packet forwarding decisions, are bundled together inside the networking devices. As a consequence, automatic

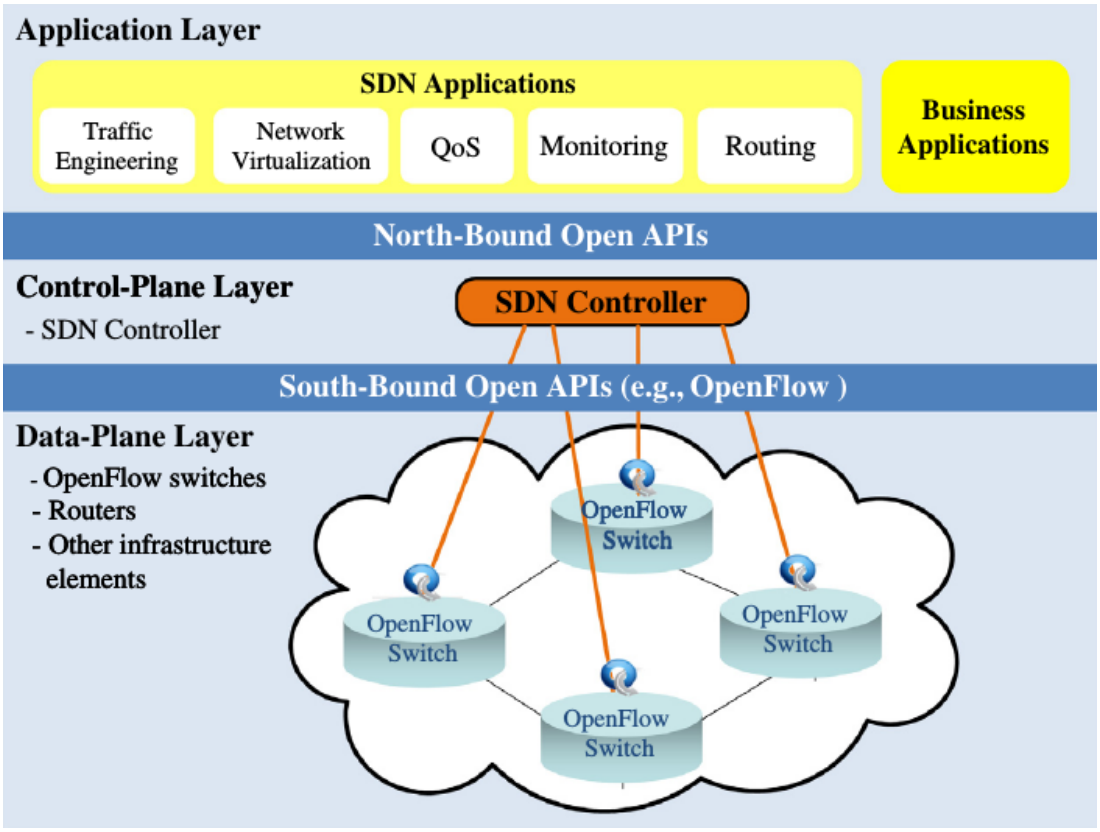


FIGURE 2.4: SDN Architecture (Akyildiz et al., 2014)

reconfiguration and enforcement of dynamic policies is highly challenging. On the other hand, in the SDN paradigm, a network controller uses decoupled communication protocols such as OpenFlow to access the network device's forwarding plane and perform path configuration functions, translating network routing policies into packet forwarding rules (Lara, Kolasani, and Ramamurthy, 2014).

The SDN architecture is made up of three layers (Figure 2.4): the controller plane; the data plane, and the application layer. The data plane includes programmable data forwarding devices such as switches and routers. At the centre of the SDN architecture is the controller plane that globally manages the data forwarding plane by configuring packet handling policies. The controller is able to communicate with devices in the data plane through the OpenFlow protocol (Lara, Kolasani, and Ramamurthy, 2014). The OpenFlow protocol allows the network controllers to access and configure the forwarding plane of the programmable forwarding devices. An SDN controller is usually a logically centralized function that determines the forwarding path for each flow in the network (Agarwal, Kodialam, and Lakshman, 2013). The controller determines the logic for packet forwarding and is implemented through the forwarding tables in the SDN devices. An SDN topology will therefore have one or multiple network controllers that have access to and are able to

update forwarding tables in all the network elements. During runtime, the SDN forwarding devices are able to communicate with the controller and obtain instructions to process incoming packets whose forwarding rules have not been established yet.

On the other hand, the application plane provides a set of Open APIs, such as the REST API, that are used for communication between the service control applications and the network controllers. The North-bound APIs allow the creation of network management applications, such as for routing, traffic engineering, multicasting, security, access control, bandwidth management, quality of service (QoS) (Akyildiz et al., 2014).

Software Defined Networking (SDN) provides new opportunities for flexible management of Internet routing and packet forwarding (Rothenberg et al., 2012). One challenge with inter-domain multi-path routing and end-to-end traffic engineering is with regard to enforcement of paths across different domains. SDN has three important characteristics that are useful for interdomain traffic engineering (Gupta et al., 2013). In traditional networks, a packet's source networks can not direct the packets' path beyond its own border router. With SDN, a controller can consolidate control messages from multiple remote networks, such that source and destination networks can remotely configure forwarding paths through a controller. Secondly, the controller's direct control of the data plane enables dynamic/programmatic configuration of the forwarding tables. For example, an SDN-based IXP (Gupta et al., 2013) allows IXP participants to have access to an SDN controller and to write policies that override the default policies of the IXP's BGP route server. Furthermore, in contrast to traditional switches that forward traffic based only on the destination MAC address, SDN enables packet forwarding based on multiple header fields.

Another advantage of SDN is that, unlike in traditional networks, data forwarding rules can be changed in real-time. The first incoming packets of each flow are sent to the controller for a forwarding decision, and thereafter, all subsequent packets belonging to that flow are forwarded based on the initial decision. The controller manages the data plane elements, such as switches, via a standard application programming interface (API). The most prominent SDN API is the OpenFlow protocol, through which the controller installs forwarding rules on network switches. Furthermore, with the separation of the control plane from the data plane, SDN makes possible the creation of multiple separate logical networks over the same physical architecture (Lara, Kolasani, and Ramamurthy, 2014). With these SDN opportunities, it is possible to allow edge networks some control over selection of inter-domain forwarding paths at Internet exchange points, thereby having more control on the end-to-end paths.

Many SDN controllers are designed to achieve the throughput required by enterprise networks and data centres (Raghavan et al., 2012; Lara, Kolasani, and Ramamurthy, 2014). These controllers are largely designed as multi-threaded systems to leverage the parallelism

of multi-core computer architectures. For example, controllers such as Beacon, OpenDayLight and Floodlight, have been reported to be able to handle more than 12 million flows per second, using large size computing nodes of cloud providers. Other controllers such as, Ryu (Ryu, 2015), are aimed for less specific environments such cloud infrastructures, and carrier grade networks. In this thesis, three prominent open-source OpenFlow controllers were tested; Floodlight (Java based), OpenDayLight (Java based), and Ryu (Python based). Performance comparison of the three controllers suggested that Floodlight and OpenDaylight required much higher CPU than Ryu controllers. For this reason, traffic engineering modules used in this thesis were built and evaluated in Ryu.

2.5 Reinforcement Learning

Optimal end-to-end path selection can be achieved when quality of links in the topology is continually evaluated so that paths with better performance are utilized more (Desai and Patil, 2015). This means that traffic engineering decisions taken in the edge networks may not always be optimal across the entire path. One way of dynamically controlling how traffic flows across the entire path is through implementation of mechanisms in network hops so that they are able to adjust their forwarding behaviour based on experience. This problem can de done using reinforcement learning approaches, where experience gathered from iterative routing decisions can be used subsequently to select better forwarding paths. Some studies (Wolf et al., 2012; Rouskas et al., 2013) have shown that correlations learned from network controller data can be utilized to improve resource allocation and network performance. It is worthwhile therefore to investigate a data driven (Yin et al., 2014) approach where SDN nodes can use existing controllers' data.

2.5.1 Reinforcement Learning for Traffic Engineering

The problems solved by Reinforcement Learning (RL) generally involve sequential decisions that can be modelled as Markov Decision Processes (MDPs) (Xu, Zuo, and Huang, 2014). A forwarding device in a multipath topology can be modelled as a Reinforcement Learning agent defined by a Markov Decision Process (MDP) (Xu, Zuo, and Huang, 2014; Boyan and Littman, 1994). Each MDP state has a collection of actions that can be performed in the particular state and the actions result in the transition of the system into a new state. MDP's state transitions are described by a transition function $T(s, a, s')$, where a is an action performed at the current state s, and s' is some new state. MDPs obey the *Markov property*, which holds that the probability of a system being in a given state is dependent only on the previous state. Thus, the system's state at any given time is determined solely by the transition function and the action taken at the previous step:

$$P(S_t = s'|S_{t-1} = s,\ a_t = a) = T(s, a, s')$$

An MDP environment also has a reward function $R : S \mapsto \mathbb{R}$ that assigns some value $R(s)$ to agents for transitioning to some state $s \in S$. The goal of a Markov Decision Process is to move from the current state s to some final state in a way that a) maximizes the immediate reward $R(s)$ and b) maximizes R's potential value in the future.

An MDP environment is modelled as a tuple (S, A, P, R); where S is a set of states, A is a set of actions, and $P(s'|s, a)$ is a transition model for the probability of entering state s' after executing action a at state s, with the condition that there is at least one corresponding action a such that $P(s'|s, a) = 1$, i.e executing action a at s implies sending a data packet from router s towards s' results in the packet subsequently being at s'. $R(s, a, s')$ represents the reward given to the learning agent for executing action a at s that caused transition into state s'. Given the sequential decision process of an MDP, the quality of the action a at state s must, apart from the immediate reward $r(s, a)$, be determined by the potential future rewards made possible by selection of action a. A routing agent learns to adjust path selection policies based on experience and rewards and, through continuous modification of action selection policies, attempts to maximize some cumulative pay-off (Xu, Zuo, and Huang, 2014).

A function $Q(s, a)$ is used to represent the value $V(s)$ achieved by the action $a \in A$. However, the consequences of actions (i.e., the rewards) and the effects of policies are not always known beforehand. As such, mechanisms are required for controlling and adjusting the action-selection policy. These mechanisms are collectively referred to as *Reinforcement Learning*. Reinforcement learning requires a *policy* function $\Pi : S \mapsto A$ for selecting the appropriate action $a \in A$ given the current state $s \in S$.

2.5.2 Q-Learning

Q-learning is a temporal difference (TD) control algorithm that uses the reinforcement learning (RL) rewards to influence selection of future state-actions (Wang and Wang, 2006). In Q-learning, the state-action pair's utility value is called the "Q-value", and is calculated by the Q-function $Q(s, a)$ (Watkins and Dayan, 1992). The algorithm approximates an optimal action-value (Q-value) function by assigning rewards for each action taken by a learning agent at each state. The *Q-learning* algorithm is responsible for evaluating the function $Q(s, a)$ through an iterative learning process that uses performance rewards to update the Q-value function that is used for selection of future actions (Watkins and Dayan, 1992; Wang and Wang, 2006). The rewards therefore act as the reinforcement signals for adjusting forwarding link priorities. Link priorities act as probabilities with which to select a particular forwarding link for outgoing traffic.

The Q-learning procedure for an agent involves three iterative processes: observing the environment's state; selecting an action; and receiving a reinforcement signal from the environment. The agent observes the state of the environment and appropriately selects one of the available actions in order to transition to the next possible state. The environment monitors the performance emanating from the agent's selected action, generates a reinforcement signal (reward) and transmits it to the agent. Lastly, the agent computes and updates its action selection policy, taking into account the reinforcement signal received for prior actions. Thus, a learning agent needs to be able to perceive the state of the environment, and be able to receive rewards, and have a learning algorithm for updating the action selection policy.

The Q-learning algorithm utilizes the MDP to compute an optimal action-value function called Q-value through an iterative approximation procedure, by assigning rewards for each action taken by a learning agent in each state and accordingly adjusting action selection policies. Positive rewards are assigned if the preceding action resulted in a state that is more desirable (approaching the goal state), and a negative reward (punishment) is given if the new state does not lead to the goal. In a routing application, Q-learning would attempt to improve the probability that better forwarding paths (next-hop selection) are selected by positively reinforcing better paths.

A Q-learning agent finds an optimal control policy by iteratively approximating its Q-values using prior Q-value estimates, a short-term reward $r = \rho(s, a) \in R$, and a discounted future reward. The updated Q-value Q^* is thus estimated based on the reward r from task a executed in state s, and the maximum expected reward from action available at subsequent state s'. Thus, the goal of maximizing the cumulative reward is represented by an action-value function $Q(s, a)$:

$$Q^*(s, a) = (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a')] \tag{2.1}$$

where the learning rate $\alpha \in (0, 1]$ models the rate of updating the Q-values, i.e how fast new information overrides previous information, and $\gamma \in (0, 1]$ represents a discount factor that scales the importance of the immediate reward (obtained for the action at s) versus rewards obtainable for actions at the subsequent state s'.

1. Repeat:

   (a) For each $s \in S$:

      i. Select an action $a = f(s)$.

      ii. $Q^*(s,a) = (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')]$

      iii. $\Pi(s) = \underset{a}{\operatorname{argmax}}\, Q(s,a)$.

ALGORITHM 1: One-Step Q-Learning

The Q-learning formula can be turned into a packet forwarding algorithm where the outgoing interface (action) is selected by a some function $f(s)$. In one-step Q-learning implementation (Algorithm 1) (Wang and Wang, 2006), a user-defined function $f(s)$ returns the appropriate policy action $\Pi(s)$, which usually is the action-option with the highest Q-value (Wang and Wang, 2006) among all the actions among $s \in S$:

Chapter 3

Literature Review

Efforts by Pan-African NRENs to deploy inter-NREN physical infrastructure has resulted in a topology that is highly multipathed. The improved physical topology of the Alliance has further opened up new possibilities for traffic engineering in NRENs, include the opportunities for dynamic selection of end-to-end paths based on QoS requirements. However, recent studies on inter-NREN communication in Africa indicate that the problem of sub-optimal routing, which causes high end-to-end latencies, is still persistent. On the other hand, the Internet's interdomain protocol, BGP, was not designed for multipath routing and is therefore inherently a single path protocol. The inability of BGP to support multipath routing makes it difficult for networks to concurrently utilize the multiple paths that generally exist between Internet nodes. The ability for concurrent use of multiple paths would be beneficial in terms of traffic engineering, with the objective of improving quality of service for inter-NREN communication, such as in terms of reducing latency and improving achievable throughputs.

This chapter first takes a look at previous efforts that have been taken to map Africa's Internet topology. Thereafter, the chapter reviews some of the solutions that have been proposed over the years to achieve traffic engineering in interdomain environments. Focus is then shifted to mechanisms that have employed SDN and LISP to achieve multipath traffic engineering.

3.1 Mapping the African Internet Topology

The discovery of Internet topologies and the monitoring of Internet traffic is a highly researched field. Studies (Motamedi, Rejaie, and Willinger, 2015; Allalouf, Kaplan, and Shavitt, 2009; Shavitt and Weinsberg, 2011; Matthews and Cottrell, 2000; Zennaro et al., 2006; Luckie, 2010) have been conducted to map the Internet and to gain a better understanding of how Internet traffic is routed and how latencies, bandwidth and other metrics could be improved.

One of the early sources of data for Africa's Internet routing and end-to-end performance was from the Pinger project (Matthews and Cottrell, 2000). By analysing measurement data from a Pinger monitoring station in South Africa, Zennaro et al. showed how, as of the year

2006, most Internet traffic from South Africa to other African countries was being routed through Europe and North America. This was noted to drastically increase end-to-end latencies, and also increased the cost of communication due to the utilization of expensive intercontinental links.

Another African routing study was conducted by Gilmore, Huysamen, and Krzesinski (2007) who carried out topology measurements from a South African vantage point. The study showed that traffic originating from the South African Tertiary Education Network, and destined for other African countries, was mostly routed via the United Kingdom, Scandinavia and America.

Chetty et al. (2013) also reported that Internet performance in sub-Saharan Africa is generally characterised by high latencies resulting from circuitous paths. Internet traffic traverses geographic locations that are far from the shortest geographic path between endpoints. Furthermore, there are significant inconsistencies between geographical distances between endpoints and their associated end-to-end latencies. These inconsistencies are mainly because Africa's end-to-end paths go through IXPs in Europe, rather than within local or regional exchange points. According to data available from PeeringDB, an open database of peering relationships, most of the IXPs in Africa have no more than 2 participants, suggesting very low levels of peering. It should be noted, however, that in some cases, networks are actually peering without necessarily announcing their peering in the database. This is the case because PeeringDB data is voluntarily provided by network operators, and it is not always the case that networks announce their peering statuses. Internet measurements research has, however, confirmed that low level of peering at Africa's IXPs. For example, Gupta et al. (2014) characterized the IXP peering situation in Africa and showed that most African ISPs do not peer among themselves at national or regional level, but rather at larger European IXPs such as those in London and Amsterdam. The exception is South Africa, which with 5 IXPs and an average of 30 networks per IXP, has the highest level of national peering. In comparison, a study by Ager et al. (2012) has shown that a single European IXP ecosystem has over 400 networks, with over 50 thousand actively used peering links, exceeding the total estimated number of all non-IXP peering links in the entire Internet. On the other hand, in parts of the developing world where IXPs are becoming prevalent and more utilized, significant Internet performance gains and cost reductions have been registered (Galperin, 2013; DeNardis, 2012). A report (Kende and Hurpy, 2012) on the impact of IXPs in Kenya and Nigeria indicated that the price of international capacity and the latency for exchanging traffic and accessing domestic content was significantly reduced.

Fanou, Francois, and Aben (2015) assessed the African interdomain routing topology and showed that a lack of interconnection among African ISPs (South Africa being an exception) still persists. The study further analysed circuitous paths with high RTTs, highlighting a

reliance on intercontinental ISPs for Africa's interconnectivity. Furthermore, the study evaluated impact of new IXPs in Africa, showing as expected, the performance benefits in terms of reduced RTTs among IXP members. Fanou et al. (2016) showed that Africa's sub-optimal Internet performance is caused by significant inter-AS delays in the continent, which contributes to local ISPs not sharing their cache capacity. They also showed that the poor Internet performance is exacerbated by the fact that most of the Web content consumed in Africa is served from the US and Europe. Additionally, poor DNS configuration used by some ISPs on the continent go against attempts of providers to optimise interconnectivity and content delivery (Fanou et al., 2016).

As a result of the low level peering among network operators, Internet performance in sub-Saharan Africa is characterised by high latencies resulting from circuitous paths. This is because Internet traffic traverses geographic locations that are far from the shortest geographic path between endpoints. Recent work on the African Internet topology (Gupta et al., 2014) has shown that about 66% of traffic between South African Internet vantage points and Africa-based Google cache servers is routed outside the continent. The same work also characterized the IXP peering situation in Africa and showed that most African ISPs do not peer among themselves at national or regional IXPs, but rather prefer to peer at larger European IXPs such as London and Amsterdam, presumably to achieve better economies of scale with access to global networks.

3.2 Traffic Engineering

Traffic engineering is the process of managing how traffic flows through the Internet, in many cases with the aim of enforcing routes that are subject to specific constraints (Xiao and Ni, 1999; Feamster, Borkenhagen, and Rexford, 2003). The Internet itself is an interconnection of many privately managed networks known as Autonomous Systems (ASes)(Shavitt and Weinsberg, 2011). Traffic exchange among ASes is facilitated through the Border Gateway Protocol (BGP), a single path routing system that conveys AS-level paths between domains, enabling them to create logical interconnectivity and exchange traffic. Any two ASes can exchange traffic if they have some logical connection between them, which could be through direct peering, or through some other transit providers that transport traffic between them(Ahmad and Guha, 2011). Due to this hierarchical structure of the Internet, traffic whose source and destination networks are geographically close but do not have direct interconnections, may sometimes have to traverse circuitous remote links through other ASes in search of interconnecting paths.

Traffic engineering (TE) research has for the past years given attention to two main issues; quality of service (QoS) provisioning and network resilience (Wang et al., 2008). The QoS aspect has been motivated by the emergence of multimedia applications that have high

demand for bandwidth and also require stringent QoS guarantees, such as in terms of end-to-end delay, jitter and loss. Network resilience continues to be important as node and link failure still occur, and there is always need to minimize impact of such failures on service availability, network performance and resource utilization.

Wang et al. (2008) categorizes traffic engineering according to four orthogonal criteria: scope (intradomain vs interdomain), routing enforcement (MPLS vs IP-based), time-scale of operations (online vs offline), traffic type (unicast vs multicast) :

MPLS vs IP-based: Multi Protocol Label Switching (MPLS) operates by setting dedicated label switched paths (LSPs) for delivering encapsulated IP packets while IP-based TE controls intradomain traffic by setting link weights on internal gateway protocols. Other IP-based approaches involve tweaking the BGP by tuning routing attributes on a per destination prefix basis. Dedicated flexibility and scalability of the TE mechanisms (as the overhead of setting up LSPs is very high in large-sized networks). But on the other hand, the IP-based approach are unpredictable (as they rely on multiple implicit topology tuning). However, IP-based methods provide better scalability and resilience.

Offline vs Online TE: The difference between offline and online traffic engineering is on the availability of information concerning the overall traffic demand from all the flows on the network, and time scale for traffic manipulation. In offline traffic engineering, traffic forecasting, based on service level specifications and/or network measurements, is used to map traffic into the physical network. The mapping gets revised after a period known as the resource provisioning cycle. Online TE does not use traffic forecasting, but instead attempts to dynamically respond to fluctuating traffic patterns, typically on a time scale of minutes to hours. The approach endeavours to evenly distribute traffic so that future flow assignments are assigned without causing congestion. Sometimes existing flows are rerouted to reserve resources for new and future flows. A major challenge for online TE is the lack global network view, which is vital for global optimization.

Intradomain vs interdomain: Intradomain traffic engineering aims to optimize customer traffic within a single domain, while interdomain attempts to optimize traffic across multiple Autonomous Systems (ASes) through optimal selection of ingress and egress points if multiple potential border routers exist. Intradomain and interdomain traffic engineering mechanisms sometimes impact each other, and if uncoordinated, result in sub optimal performance. For example, for the interdomain mechanism to make use of a particular egress router, the traffic may have to traverse specific internal links, thereby impacting the intradomain performance.

3.2.1 Interdomain Traffic Engineering

In inter-domain contexts, traffic engineering is employed to control traffic entering or exiting an AS, with the objective of load balancing over multiple interdomain links. The Internet is composed of backbone and customer networks. Backbone networks mainly provide transit service for traffic exchanged between customer networks. Based on hierarchy in the global routing architecture, backbone networks are categorized into different tiers. Interdomain relationships broadly exist in two categories. The first category involves transit service in customer-provider relationships between lower and higher tier ASes. The second category, known as peering, involves neighbouring and roughly equal size ASes. Furthermore, ASes are either transit networks that offer transit services, or edge networks (also called stub networks) that only send or receive traffic.

In general, transit and edge networks have different TE objectives. Transit networks aim to optimize network resources to maximize revenues. Stub networks, which, according to Wang et al. (2008), constitute more than 80% of all Internet ASes, aim to minimize cost and achieving best application level QoS. Furthermore, content provider networks generally aim to optimize outbound traffic links, while content consumers are more interested in inbound TE. Transit providers are typically concerned with both inbound and outbound TE.

Achieving optimal end-to-end routing in interdomain networks is problematic. This is the case because although source routers are responsible for selecting gateways for sending traffic towards the destination, it is difficult to control the entire traffic path(Secci, Liu, and Jabbari, 2013; Saucez et al., 2008). Current approaches of the Border Gateway Protocol (BGP) path pre-pending and local preferences to perform TE have been shown to be unreliable and inefficient. Furthermore, due to the limitations of BGP, an inherently single path system, alternate routes are not disseminated as each router selects and advertises a single best path(Xu and Rexford, 2006). Over the years, traffic engineering research has focused on the application of BGP to control traffic within ASes (Xiao and Ni, 1999; Afergan and Wroclawski, 2004; Rajahalme et al., 2011).

While many interdomain traffic engineering schemes have been proposed, practical implementation has remained a challenge. Wang et al. (2008) attributes the practicality problem of interdomain traffic engineering to the non-cooperative and sometimes conflicting routing strategies of individual ASes. Research and education networks are a special category of backbones providing transit service only for research and education networks at national and regional scales. Unlike commodity backbones that provide global transit, RENs provide partial-transit, only admitting traffic between research and eduction networks. In federated environments such as the NRENs core topologies, it is not hard to imagine traffic engineering cooperation between participating NRENs. In fact, as partial-transit providers, research and education backbones exchange partial routing tables comprising only of NRENs (Li et al., 2010).

For end-to-end communication, the challenge is that selection of Internet route paths is mostly influenced by routing policies that are optimized for interests of individual ASes. Such policies do not provide guarantees for optimal end-to-end connections and sometimes result in packets not traversing the shortest paths to their destinations. To deal with this challenge, inter-domain traffic engineering (TE) techniques aim to optimize resource utilization and internetwork performance through mechanisms that identify and dynamically use optimal low-latency paths. This requires routing systems that are able to learn and make use of inter-domain topology information in choosing routing paths.

Well known mechanisms for outbound TE make use of BGP, the Internet's de facto interdomain routing protocol for exchanging routing information. These mechanism include setting the local preference priorities from different border routers so as to influence selection of egress points in a domain. For stub networks, proactive and reactive online TE mechanisms have been used to dynamically select appropriate egress links with the objective of minimizing overall expenses, reducing end-to-end latency, and to achieve bandwidth requirements. Proactive solutions rely on traffic forecasting to autoconfigure TE algorithms. One of the early examples on proactive TE is presented in (Goldenberg et al., 2004), where traffic prediction is based on a sequence of independent preceding traffic measurements. On the other hand, reactive solutions dynamically adapt to incoming traffic demands to auto tune TE algorithms. For example, in (Lee, Zhang, and Nelakuditi, 2004), a multihomed network dynamically switches traffic between access links based on measured end-to-end path delay, and reportedly achieves a 40% improvement in performance compared to random selection of provider network.

Inbound TE is more complicated to implement than outbound TE, principally because traffic sources and transit nodes have the liberty to forward traffic according their own local preferences. In BGP routing, a few mechanisms have been used to achieve inbound TE include selective advertisement, prefix de-aggregation, and AS path pre-pending (Secci et al., 2011c). In selective advertisement, routes to different destinations are purposefully advertised only on specific ingress links, while prefix de-aggregation ensures that incoming traffic is matched to longer and more specific prefixes. Prefix de-aggregation has the negative impact of increasing router table sizes on the Internet. In AS path pre-pending, multiple instances of the same AS number are added to an AS-level path so that it appears longer and less attractive for incoming traffic. Other inbound TE approaches include the use of multi-exit Discriminator (MED), community values, and Network Address Translation (NAT). These approaches have been shown to be inefficient and computationally complex (Secci et al., 2011b; Secci et al., 2011c).

Interdomain TE is generally difficult to implement for at least two reasons (Wang et al., 2008): firstly, the policy-based routing infrastructure of BGP was not designed with consideration for how individual routing policies can be systematically integrated to optimize

interdomain traffic. As a result, it is quite difficult to achieve global optimality in an environment of non-cooperative and sometimes conflicting domain specific routing strategies. The TE strategies and policies of [adjacent] domains are sometimes conflicting and have adverse effects on different domains. The second challenge with interdomain TE is that since each domain is privately managed and routing decisions cannot be directly enforced on other domains, achieving end-to-end traffic engineering cannot be guaranteed. Meaningful global optimization and end-to-end traffic engineering requires some level of cooperation among the neighbouring networks.

A large body of research has looked at the use of BGP to support cooperative traffic engineering between different ISPs (Shrimali, Akella, and Mutapcic, 2010; Mahajan, Wetherall, and Anderson, 2007; Mahajan, Wetherall, and Anderson, 2005; Feigenbaum et al., 2005; Machiraju and Katz, 2004; Suksomboon et al., 2010). Shrimali, Akella, and Mutapcic (2010) proposed an interdomain cooperative BGP-based traffic engineering model using Nash bargaining and dual decomposition. The model enables ISPs to use an iterative procedure to optimize a joint cost function. Under this scheme, the global optimization problem is broken down into subproblems based on interdomain flows, which are then solved independently in a decentralized manner by individual ISPs. The approach eliminates the requirement for ISPs to share any sensitive internal information, such as network topology or link weights, to achieve optimal traffic engineering.

Mahajan, Wetherall, and Anderson (2005) analysed how neighbouring ISPs could cooperate with each other for inter-domain traffic engineering. The solution formulated a negotiation algorithm as a basis for cooperation, allowing adjacent ISPs to share information using coarse preferences and jointly computing the paths for the traffic exchanged between them. Later, Mahajan, Wetherall, and Anderson (2007) implemented a BGP extension that enabled ISPs to jointly control routing to produce efficient end-to-end paths between them even when each ISP acts in their own selfish interests. The extension enables each ISP to select routes that provide a compromise between each ISP's own optimization objectives and those of other ISPs. This then allows ISPs to individually optimize routing based on their own optimization criteria.

Jacob and Davie (2005) provided a cooperative mechanism for traffic engineering between ASes. Suksomboon et al. (2010) proposed a cooperative method based on a series of game theoretic learning processes. Two communicating edge ASes can directly negotiate with each other for joint load balancing optimization for each one's incoming traffic. This approach works in situations where traffic between two edge ASes are balanced. This optimization scenario has been analyzed in previous works (Secci et al., 2011a; Secci, Liu, and Jabbari, 2013).

3.2.2 Multipath Traffic Engineering

The Internet is multipath environment, with several ASes having multiple interconnections between them. ASes generally deploy multiple interconnection links with other ASes with the aim of increasing resilience, as well as to improve performance and reduce transit costs through policy based routing.

Multipath forwarding is a link and network layer mechanism that aims to take advantage of multipath environments. Packets from source to destination are made to flow over multiple paths. Concurrent multipath forwarding is when data is split and forwarded over multiple paths concurrently. Concurrent multipath forwarding has performance advantages in terms of increasing the available bandwidth through aggregation as well as the possibility of using shorter delay paths. Apart from performance enhancement, multipath forwarding can also be used to achieve connection resiliency in that traffic flow does not get disrupted with the failure of a single link if other active links exist between the communicating parties (Shu et al., 2016). Furthermore, multipath forwarding enables traffic engineering and load balancing, where the selection of paths depends on application requirements, such as in terms of bandwidth or latency.

For many years, the networking research community has been looking at ways of utilizing the Internet's inherent redundancy and path diversity to improve network performance through flexible traffic engineering. There are motivations for flexible multipath routing, including the ability to configure paths according to application QoS requirements and for improving end-to-end path resilience (He and Rexford, 2008). In this regard, multipath routing mechanisms aim to enable selection of multiple routes while preserving normal business relationships such as transit, peering, siblings, partial transit (Camacho et al., 2013). Such flexibility is hard to achieve as it requires dissemination of multiple paths for the same destination, yet a larger proportion of the interdomain path diversity is diminished by BGP's propagation of a single best path per destination (Mérindol et al., 2009). A recent version of TCP incorporates multipath forwarding and is aptly named MultiPath TCP (MPTCP) (Raiciu et al., 2010). In MPTCP, the source node splits a single flow into multiple subflows and employs TCP options for data re-sequencing at the receiver. A major drawback attributed to MPTCP is its requirement of existence of multiple paths and of the discovery thereof by the underlying routing protocol (Campista et al., 2014). Flexible interdomain TE requires the ability to dynamically determine and reconfigure optimal end-to-end paths. However, since ASes are privately managed and each one independently decides how to forward the Internet traffic coming through it, meaningful end-to-end traffic engineering requires some level of AS coordination mechanism (Secci, Liu, and Jabbari, 2013; Mahajan, Wetherall, and Anderson, 2004).

However, despite the recognized potential of multipath forwarding, the approach has not been fully exploited in the Internet, especially at the edge and core levels (Valera et

al., 2011). He and Rexford (2008) showed that, in multipath Internet environments, better alternative paths with lower loss rate and delay are available between 30% and 80% of the time. This does suggest that ASes can improve their routing performance by discovering better paths and optimally re-distributing traffic. Although using alternative paths could significantly improve packet forwarding performance, this capacity has been left untapped by networks for fear of the negative impact that multipath routing has on the upper layers. For example, packets from one TCP connection may arrive out of order if path diversity is used at a fine-grained packet level, instead of at the flow level (Campista et al., 2014).

Standard approaches for influencing selection of paths across multiple domains have relied on manipulating the Border Gateway Protocol, but these approaches have been unreliable and inefficient (Saucez et al., 2008). A key problem is that, as an inherently single path system, BGP does not disseminate alternate routes. Each BGP router selects and advertises only the best path (Xu and Rexford, 2006) to its neighbours. By propagating only a single path (default route), the multipath diversity available in an internetwork is diminished. Yet, it is not always the case that the default BGP routes offer the best performance.

Intradomain Multipath

Early multipath traffic engineering efforts mainly focus on intradomain use cases (Wójcik et al., 2016). First, consider that for intradomain Layer 2 networks, the Spanning Tree Protocol (STP) is used to ensure loop-free packet forwarding. However, STP has the disadvantage of pruning topologies, diminishing the multipath capability of networks (Wang, He, and Su, 2015). Paths that could be utilized to achieve increased throughput are removed from the list of active paths. This constitutes a waste of network capacity that might otherwise be utilized. A possible solution to this is Link aggregation (IEEE 802.3ad) (Nong et al., 2014), but this also fails to work for paths traversing multiple switches (Chiesa, Kindler, and Schapira, 2014). To achieve multipath in intradomain environments, prominent implementations, such as the Open Shortest Path First (OSPF) (Fortz and Thorup, 2000) and Intermediate System to Intermediate System (ISIS) employ Equal Cost Multipath Routing (ECMP) (Singh, Das, and Jukan, 2015; Chiesa, Kindler, and Schapira, 2014) .

ECMP is a forwarding technique employed in Layer 2 and Layer 3 and has been largely implemented for intradomain traffic engineering schemes (Singh, Das, and Jukan, 2015; Chiesa, Kindler, and Schapira, 2014). With ECMP, routers maintain multiple STPs, and at least two paths are chosen interchangeably for forwarding the traffic as long as the paths have the same cost. The cost is determined using different network metrics, including the number of hops to the destination or link speed. To implement ECMP, a network needs to have a set of loop-free paths that can be configured by IP routers with appropriate OSPF/IS-IS link weights and applying the Dijkstra's algorithm (Wang et al., 2008). Wang, Wang, and

Zhang (2001) showed that an arbitrary set of loop-free routes can be resolved into shortest paths by applying a linear programming formulation to a set of positive link weights. With each IP router directly computing the set of shortest paths, there is no need for maintaining label-switched-paths (LSPs), and the traffic engineering can effectively be accomplished through native hop-by-hop-based routing, thereby avoiding the complexity and cost of MPLS (Wang et al., 2008). Given multiple shortest paths with equal IGP/IS-IS link weights toward the same destination, an ECMP mechanism evenly splits traffic onto each path's next hop router. In the flow-granularity approach, the routing mechanism computes a hash over the packet fields and forwards it to all the hops on a chosen path. As a result, all packets belonging to the same flow take the same path. Packet-granularity, on the other hand, splits the flows and forwards single packets over the multiple paths, aiming to use as much network capacity is possible. To avoid out-of-order packet arrivals, the splitting of traffic in ECMP is mostly on a per flow basis as opposed to per packet basis. Flow-level ECMP approaches do not provide aggregated capacity, as all packets in a flow are forwarded over the same path.

Fortz and Thorup (2000) employed an ECMP optimization of OSPF/IS-IS link weights for the purpose of load balancing in a multipath environment. Their approach was to adjust the weights outgoing links from each particular node, with the intention that new paths with equal cost can be set up from that node toward a destination. The mechanism results in splitting of traffic that would have travelled on a single path towards the destination, into multiple even streams.

Interdomain Multipath

In an Internet ecosystem where networks are multihomed, it is common to have multiple alternate paths between a pair of source and destination networks. Therefore achieving optimal end to end performance requires not just the ability control traffic flows from source to destination (Secci, Liu, and Jabbari, 2013; Saucez et al., 2008), but also importantly the ability to discover multiple paths and obtain path metrics from different domains. In multipath environments, each end to end path has its own unique path metrics in terms bandwidth, delay, and loss. Also, different players aim for different aspects of traffic flow optimization, which sometimes results in conflicting and negative overall effects. For example, while content providers are more interested in optimizing their outgoing links, access networks are interested in optimizing their incoming traffic links. On the other hand, multi-homed enterprise networks are interested in optimizing traffic on multiple access links to achieve certain levels of QoS (Quoitin and Bonaventure, 2005). Due to the diversity in domain-specific flow optimization goals, it is difficult to guarantee that the forwarding paths selected by the source network are the most optimal from the perspective of the destination. Although

it has been shown (Saucez, Donnet, and Bonaventure, 2008) that multihoming allows networks to choose better QoS paths over the Internet, a globally optimal TE solution needs coordination and collaboration among several domains that form part of the path (Quoitin and Bonaventure, 2005).

One of the early frameworks for collaborative traffic engineering is presented in (Awduche, Agogbua, and McManus, 1998). The framework aimed to determine optimal locations for peering points between ASes, with the objective to minimize the cost of bilateral peering and improving the efficiency of inter-domain traffic exchange. The framework used an integer programming formulation to determine the optimal peering point location, and also to determine the set of nodes that can optimally use each of the selected peering points. Although the framework is aimed at solving a bilateral peering problem, the framework does not provide for joint determination of the optimal peering point, as each one of the ASes unilaterally computes an optimal solution from its own perspective. A similar framework in (Johari and Tsitsiklis, 2004) formalizes the problem for optimal placement of interconnection links in an environment where network providers act in their own self interest first, but need to agree simultaneously on the placement peering points. Their work did show that given two providers with similar networks, under some symmetrical traffic conditions, there exists a unique peering point placement that simultaneously satisfies both providers.

In practice, a number of inter-domain multipath routing schemes have been implemented and adopted. Prominent ones include the Generalized Multiprotocol Label Switching (GMPLS) (Mannie, 2004), BGP Add-Paths (Walton et al., 2016), Locator/ID Separation Protocol (LISP) (Li, Wang, and Wang, 2011; Saucez et al., 2012) and Segment Routing (Greene et al., 1991). GMPLS, a successor to MPLS, uses implicit packet labelling to represent and identify different paths (LSPs) using some physical property of the received data stream. Schemes used by GMPLS to identify LSPs include time slots through the Time Division Multiplexed (TDM), wavelength through Wavelength Division Multiplexed (WDM), and the physical port. BGP Add-Paths is a BGP protocol extension designed to solve the lack of router-level path diversity in BGP architectures. The technique works on the basis of enabling BGP sessions to disseminate multiple BGP paths towards the same IP prefix. Segment Routing (SR) is a network architecture that incorporates source routing and tunnelling paradigms, enabling network nodes to steer packets over paths using a sequence of instructions embedded within packet headers. Such packet header instructions, known as segments, specify path segments to be traversed by the packet. Since the path is specified within the packet, SR supports implementation of routing policies that do not require per-flow entries in intermediate routers.

To achieve multipath traffic engineering, Xu and Rexford (2006) uses parallel mechanisms to propagate additional paths between ASes. Another approach presented in (Bhatia, 2003) uses path aggregation to advertise multipath sets. A recent proposal for multiple path

advertisement called BGP extended multipath (BGP-XM) (Camacho et al., 2013) implements multipath route advertisement by aggregating multiple paths to the same prefix in a single BGP update message.

Other research efforts have focused on theoretical modelling and complexity analysis of interdomain traffic engineering interactions. A number of such work has looked at using game theory to model interdomain coordination for optimal traffic engineering (Altman et al., 2006). The approaches in (Orda, Rom, and Shimkin, 1993; Altman and Kameda, 2005) use cooperative games to model interaction where each player has a specific amount of traffic and a set of parallel paths to split the traffic into. In other works (Secci et al., 2011b; Secci et al., 2011c; Liu, Jabbari, and Secci, 2013), non-cooperative games have been used to mitigate lack of coordination among peering ASes. By considering routing and congestions costs for source and destination networks, a peering equilibrium multipath coordination framework is built to adaptively preventing link congestions and excessive route deviations that are caused by network impairments.

3.2.3 LISP Multipath Traffic Engineering

By separating the host address space from the locator space, LISP introduces a level of indirection between host IP addresses and their location in the Internet topology. The split thus introduces a two-level routing architecture on top of the current BGP/IP infrastructure, making it possible for a host (the EID) to be mapped dynamically to one or multiple locators (RLOCs). A key characteristic of the LISP Mapping System is that an EID can be mapped to multiple locators. As a result, multihoming becomes easier to implement because one EID can be associated to more multiple RLOCs.

LISP capabilities facilitate a new set of traffic engineering applications, including virtual machine mobility, layer-2 and layer-3 virtual private networks, and edge-network-based traffic engineering (Phung et al., 2014). The presence of multiple locators for the same destination radically increases the path diversity (Secci et al., 2011a) as networks are able to retrieve and use multiple gateways for exchanging traffic.

In Figure 2.2 for example, four interdomain paths could be used between the two LISP sites; Site A and Site B. Site A has two locators, rloc-A1 and rloc-A2, and the site's EIDs can be mapped through the LISP mapping system to one or both of the locators. Multiple paths between two remote LISP domains become possible when at least one of them is multihomed. For example, if some identifier in Site A gets announced into the mapping system as being reachable via both the two locators, then it becomes possible for the sender (e.g Site B in this case) to use two paths to reach the EID in Site A. The first path would go through rloc-A1, and the second would be through rloc-A2. In the same vein, Site B would,

depending on its own internal routing policy, forward the traffic destined for Site A's rloc-A1 and rloc-A2 through either rloc-B1 and rloc-B2. In total, the LISP-based dual-homing of both Site A and Site B provides the EIDs in the two sites with four different interdomain paths through which to exchange traffic.

Furthermore, the LISP mapping systems permits the definition of different priority and weight values for each EID-locator mapping, and these priorities/weights are used to indicate preferences for routing to EIDs through specific locators. Priority and weight values can also be used to load balance traffic among multiple locators to the same EID. LISP's multihoming capabilities have also contributed to a simplified traffic engineering mechanism (Saucez, 2011).

In multihomed networks, LISP not only enables source networks to actively select the destination's egress gateway, but also allows the destination networks to influence the selection of incoming paths by ranking their multiple egress gateways. In principle, LISP protocol expects source networks to distribute outgoing traffic with respect to destination's preferences (i.e egress gateway ranking). However, there are no guarantees that all traffic source networks will honour the published preferences (Secci, Liu, and Jabbari, 2013), especially in the absence of incentives for doing so. For example, if the destination's preferred paths are different from the sender's, then the sender is likely to send traffic using its own preferred outgoing path. Some level of cooperation is required between interacting ASes to achieve meaningful traffic engineering. In a cooperative environment, edge networks can work to balance their own local preferences with those of the destination networks. Furthermore, the cooperating networks can achieve performance based traffic engineering by jointly performing network topology measurements to determine best end-to-end paths between them.

As LISP has continued to mature, it has become more popular for its ability to simplify multihoming and traffic engineering (Campista et al., 2014). LISP's traffic engineering capability has been enhanced with the inclusion of the Explicit Locator Path (ELP), a locator encoding that explicitly lists all the intermediate routers from source network to destination. Locator priorities and weights can also be applied in ELPs, making it possible for EIDs to be mapped to multiple paths with different priorities and weights. Using priority and weight attributes, LISP allows the use of different paths for the same source/destination pair, either as single paths in backup fail-over mode, or in multipath load balancing schemes. Each ELP consist of a list of locators, which serves to force packets to traverse the locators in the same order as listed in the explicit path. The intermediate routers are refereed to as Re-encapsulating Tunnel Routers (RTRs) and are responsible for receiving and re-encapsulating packets before forwarding them to the next RTR in the path. Multiple Map & Encap cycles are executed for complete end-to-end forwarding of packets on each chosen path.

One prominent work that uses LISP for traffic engineering is the ISP-Driven Informed

Path Selection(IDIPS) (Saucez, Donnet, and Bonaventure, 2008). IDIPs uses a LISP-based request/response service where server nodes perform network measurements towards popular destinations. Network hosts wishing to send traffic request path rankings for a set of sources and destinations. The IDIPS server ranks the available paths based on a client's ranking preference and measured path metrics (Saucez et al., 2008). Path ranking is further influenced by destination's preferences in the locator mapping. The selected paths therefore reflect not only the source network's ranking criteria, but also the destination's preferences for incoming traffic. However, it is not possible to guarantee that the ranking criterion applied by the source network is not in conflict with traffic engineering strategy of the destination network, or that the destination's preferences provide the best end-to-end performance.

Other recent works have proposed new LISP-based mechanisms for achieving traffic engineering. Saucez et al. (2008) proposed a solution that allows a LISP-enabled ISP to influence flow of its inter-domain traffic. Li, Wang, and Wang (2011) focused on incoming traffic engineering based on priorities assigned to LISP EID/locator mappings. Secci, Liu, and Jabbari (2013) extended the cooperation to allow multiple edge ASes that have significant mutual traffic between them to cooperate in engineering their mutual traffic. The multi-AS cooperation employed by Secci, Liu, and Jabbari (2013) decomposes the multi-AS interactions into game theoretic binary-AS processes that are aimed at achieving mutual optimization objectives. Secci et al. (2011) used a game theoretic framework to model ASes as selfish entities, and devised a LISP-based traffic engineering framework using a non-cooperative game approach.

3.2.4 SDN Traffic Engineering

A key distinguishing feature of SDN-based traffic engineering is the use of a central controller in the configuration and management of data paths. Such a centralized mechanism provides opportunity for globally and dynamically analysing, predicting and regulating the behaviour of transmitted data. SDN has the unique characteristic of having a controller that can maintain a global view of the topology, and formulation of forwarding decisions is centralized as opposed to being distributed. This network visibility and decoupling of the forwarding intelligence from the nodes makes it possible to implement load balancing mechanisms that consider the prevailing network characteristics. Where multiple paths between the source and destination node in an SDN exist, it is possible to design traffic engineering mechanisms that dynamically forward data on multiple paths to achieve specific QoS requirements. Several SDN-based traffic engineering approaches have been proposed, many with the aims of maintaining network availability and improving performance (Shu et al., 2016).

In non-centralized TE mechanisms such as MPLS, each site independently establishes the inter-domain forwarding paths, usually resulting in an unpredictable and sometimes sub-optimal TE solutions. SDN enables a global view of a network topology and traffic pattern through a centralized controller, and makes possible performance based dynamic reconfiguration of forwarding rules in network devices. With these capabilities, SDN has been applied to TE in four main respects; flow management, fault tolerance, topology updates and traffic analysis. Dynamic flow management is one of the major contributions of SDN to traffic engineer. In general, flow management is achieved through a controller device that is responsible for configuring new forwarding rules for unmatched packet flows.

Several traffic engineering applications have been proposed for such purposes such as maximizing aggregate network utilization, optimizing load balancing, as well to enhance energy efficiency. Some of the applications include Hedera (Al-Fares et al., 2010), Aster*x (Handigol et al., 2011), ElasticTree (Heller et al., 2010), OpenFlow-based server load balancing (Wang, Butnariu, and Rexford, 2011), Plug-n-Serve (Handigol et al., 2009), In-packet Bloom filter (Carlos, Rothenberg, and Maurício, 2010), SIMPLE (Qazi et al., 2013), QNOX (Jeong, Kim, and Kim, 2012), and QoS framework (Tomovic, Prasad, and Radusinovic, 2014). Some of the SDN-based multipath traffic engineering approaches have been proposed for improving network resilience and performance (Shu et al., 2016), and have been deployed at data centre scale as well as global WANs (Akyildiz et al., 2014).

ElasticTree (Heller et al., 2010) was designed to dynamically adjust the set of active network elements - links and switches - to response to changing data centre traffic loads. The primary goal was to regulate power consumption in network by minimizing wasteful running on network elements. Hedera (Al-Fares et al., 2010) implements a scheduling mechanism that dynamically modifies data centre flows in response to traffic load. The mechanism was reported to increase network utilization. Plug-n-Serve (Handigol et al., 2009) attempts to minimize HTTP response time by using OpenFlow customization of flows to control the load on network links and Web servers. Aster*x (Handigol et al., 2011) is a distributed network load balancer that uses OpenFlow to monitor the state of network elements, and uses the networks metrics to achieve a more scalable and dynamic control of the data paths. An in-packet Bloom filter based data centre architecture (Carlos, Rothenberg, and Maurício, 2010) utilizes multiple physically distributed OpenFlow controllers to implement load balancing, while attempting to achieve network scalability, performance and fault-tolerance.

SIMPLE (Qazi et al., 2013) is an SDN-based policy enforcement layer to enable traffic engineering within the constraints of legacy middleboxes. The policy enforcement layer is meant to make it possible for SDN to coexist with the existing infrastructure. QoS-aware Network Operating System (QNOX) (Jeong, Kim, and Kim, 2012) is a QoS-aware extension of the Network Operating System (NOX) enhanced with the capabilities for QoS-aware virtual network embedding, end-to-end network QoS assessment, and collaborations among

control elements in other domain network.

An SDN control framework (Tomovic, Prasad, and Radusinovic, 2014) designed for QoS provisioning, dynamically configures the network devices to provide required QoS level for multimedia applications. The controller also monitors state of the network resources and performs smart traffic management according to collected information.

OpenQoS (Egilmez et al., 2012) is an SDN controller that implements a routing mechanism designed to dynamically optimize routes for multimedia traffic. Similarly, video over Software-Defined Networking (VSDN) (Owens II and Durresi, 2015) is an application specific traffic engineering architecture designed to select the optimal path for a video stream by using a controller's network wide view. The system is based on a client/server protocol that allows an application to request the required QoS from the network.

An example of WAN-scale SDN traffic engineering mechanism is Google's B4, a private WAN that connects Google's data centres and edge deployments for cacheable content across the globe (Jain et al., 2013). The architecture, built for load balancing and link optimization, uses OpenFlow to centrally control WAN switches and to split application data flows among multiple paths, taking into consideration capacity and application priority/demands. B4's architecture is based on separation of the routing control plane from data forwarding plane, and implements a centralized TE solution with three key characteristics: balancing competing application demands at the network edge during resource constraint; using multipath forwarding/tunnelling to leverage available network capacity in accordance with application priorities; and dynamically reallocating bandwidth in the face of link/switch failures or shifting application demands (Jain et al., 2013). The B4 architecture is made of three main layers; the switch hardware layer, the site controller layer, and the global controller layer. The switch hardware performs basic forwarding primitives, while a site controller maintain a network state and sets forwarding instructions on switches within a site. At the top level is the global layer that comprises centralized applications that provide global centralized control through the site-level controllers. B4's network graph represents sites as nodes and site-to-site connectivity as links, resulting in site level tunnels that are implemented through IP in IP encapsulation. Further, the architecture aggregates source to destination flows based on QoS requirements to form forwarding groups (FGs). End-to-end tunnelling involves a universal controller installing FG-based rules at multiple site switches.

Bell Labs have used an approach similar to B4, by leveraging the centralized controller to implement dynamic routing for SDN even in cases where there is only a partial deployment of SDN capability in a network (Agarwal, Kodialam, and Lakshman, 2013). In such a partial deployment, an SDN controller computes the forwarding tables only for a few forwarding elements, while the rest of the network performs hop-by-hop routing using existing protocols. The SDN controller peers with other network nodes to obtain link weights and other

topology information, and uses its routing logic to achieve network optimization (improve network utilization and reduce packet loss and delays). With the notion that delay and packet loss are increasing functions of link utilization, the systems key optimization objective is minimize maximum link utilization; ie for every link, the total flow should be less than the product of maximum link utilization (θ) and the capacity of the link. However, in long distance communications, physical distance also contributes significantly to the delay and needs to considered. Furthermore, for interdomain interactions, economic cost and policy preferences are crucial in the selection of paths.

Another SDN WAN example is Microsoft Corporation's implementation of an SDN based WAN (SWAN) (Hong et al., 2013), where a central controller determines when and how much traffic each network service is able to send, and frequently reconfigures the network's data plane to match current traffic demand. SWAN utilizes policy rules to allow inter data centre WANs to carry significantly more traffic for higher-priority services, while maintaining fairness among similar services. Making use of SDN controller's global network, SWAN is able to optimize the network sharing polices, thereby being able to carry more traffic and support flexible network sharing. The authors report that SWAN is able to carry about 98% of the maximum allowed network traffic, whereas in contrast, traditional MPLS-enabled WANs are only able to carry about 60% of the maximum allowed network traffic.

SDN has also been used in a WAN architecture called INFLEX (Araújo et al., 2014) to provide edge networks with greater flexibility and control in end-to-end resilience, by providing on-demand path fail-over for IP traffic. The architecture allows an SDN-enabled routing layer to expose multiple routing planes to the transport layer, by having central SDN controller configure the multiple routing planes onto the edge switches. End hosts are connected to the SDN-based edge switches, and both use in-band signalling (using the Differentiated services (DS) field in each IP packet) to signal which routing plane to use. For outgoing traffic, hosts set a label on the outbound packets according to the assigned forwarding plane, and edge switches map the marked packets to the appropriate forwarding plane. While the INFLEX architecture provides an example of how edge hosts and networks can recover from failures by providing them with multiple routing planes, it does not enable edge networks to dynamically select the forwarding plane based on other factors, such as performance or cost. End-to-end resilience is achieved by allowing the end hosts to request new forwarding planes when they detect failures at the transport layer.

Similar to SDN's controller approach, other centralized traffic engineering mechanisms have been developed. For example Route Control Platform (RCP) (Caesar et al., 2005) uses a centralized BGP route computation engine to perform TE.

3.2.5 SDN in Internet Exchange Points

One solution that reduces circuitous routes is through the use of Internet Exchange Points (IXPs) (Chatzis et al., 2013). IXPs are infrastructure that enable Internet's ASes to establish mutual peering agreements that facilitate direct exchange of Internet traffic within local geographical areas, without the use of longer and usually more expensive links that traverse transit providers. Due to their two primary advantages: minimising the transit cost by keeping local traffic local, and their potential to achieve lower packet transmission delay (latency), IXPs have become common feature in the Internet topology and have greatly contributed to Internet's peering fabric (Ager et al., 2012). Furthermore, apart from peering at one or more IXPs, ASes normally have links to multiple other transit providers, in what is called multi-homing. As a result, ASes usually have multiple possible BGP routing paths between them and the Internet is a highly multipath environment.

Although IXPs have become common in the Internet topology and that there is path multiplexing through the route servers, the traffic engineering solutions explored so far have not considered leveraging the availability of several paths aggregated at the IXPs. Route servers manage route advertisements and establish BGP sessions on behalf of the ASes and act as BGP multiplexers, sending to each member one 'best' path per destination prefix. By sending only the default path to the peering participants, the multipath diversity available at the IXP is diminished. Yet, it is not always the case that default BGP paths are the best in terms of performance. A study (Ahmad and Guha, 2012) that compared the routing performance of IXP paths versus alternate non-IXP paths between the same set of source and destination ASes, showed that of all the possible paths, only about 60% of the best available paths were through the default IXP links.

Other SDN TE mechanisms have focused on implementations in IXPs. For example, Gupta et al. (2013) have proposed an SDN-based IXP architecture where participants have access to an SDN controller and are able to write policies that override the default policies from the IXP's BGP route server. The architecture employs a virtual abstraction mechanism that enables participants to have different logical views of the IXP topology, depending on their peering relationships. A controller-based application performs sequential composition and aggregation of multi-source policies, and then re-configures the IXP forwarding tables to override default forwarding behaviour. By having remote access to the IXP controller, ASes can communicate their route preferences to multiple IXPs and have more control of peering relationships. However, by relying on a BGP route server, the proposed design inherits BGP limitations with regard to multipath routing. Furthermore, the architecture does not provide means for global optimization in the selection of interdomain paths.

Another SDN IXP example is the Google's Cardigan project (Whyte, 2012; Stringer et al., 2014) that aims to create an SDN-based Internet exchange fabric that is physically distributed

in multiple locations. Participants peer through a BGP route server while an OpenFlow controller enforces the peering and forwarding policies across the fabric.

3.2.6 SDN for Multipath Traffic Engineering

The most prevalent traffic engineering solutions, which are mostly based on MPLS technology, have significant challenges with regards to multipath routing (Mendiola et al., 2016b). Mendiola et al. (2016) highlights and suggests how SDN can provide opportunities for solving these challenges, which include: setting traffic splitting rations; suboptimal path computation algorithms; outdated view of network state; and long convergence times for distributed protocols. With regards to traffic splitting, MPLS-based per-packet traffic splitting results in an excessive packet out-of-order arrivals, which generally leads to throughput degradation and increased jitter (Leung, Li, and Yang, 2007). Although per-flow traffic splitting avoids packet reordering, the splitting ratios are still determined by forwarding elements that do not have a global awareness of network state. This lead to the assignment of inappropriate traffic ratios to the available paths, resulting in suboptimal overall network performance. SDN provides better opportunities for splitting traffic a various levels of granularity, as well as computing the traffic splitting ratios from a controller vantage point that affords a global view of network state. Furthermore, SDN is able to provide a logically centralized path computation, which has a better network view that is up-to-date in real-time, and thus has the possibility to compute globally optimal paths (Mendiola et al., 2016b). SDN is also able to provide faster traffic engineering convergence due to the high network programmability as well as the out-of-band management of the network resources.

Owing to the new SDN opportunities over the legacy MPLS-based traffic engineering solutions, several SDN-based multipath solutions for load-sharing have been proposed (Wang, He, and Su, 2015; Yan et al., 2015; Izumi et al., 2015; Subedi, Nguyen, and Cheriet, 2015; Li and Pan, 2013). For example, HiQoS (Yan et al., 2015) makes use of multiple paths between source and destination and applies a queuing mechanism to guarantee QoS for different types of traffic. The approach defines path costs as comprising of a weighted combination of the estimated price for using the path links, link stability and robustness, the physical distance and the bandwidth of the links respectively. The HiQoS controller periodically measures the bandwidth utilization of each queue along the path, and the path with the minimal bandwidth utilization of a queue is selected as the optimal path for a new flow.

Similarly, M2SDN (Wang, He, and Su, 2015) considers link utilization to dynamically schedule flows towards multiple less loaded paths. M2SDN calculates link costs based on utilization and packet drop rate, and attempts to split traffic on multiple paths, applying a path dependency parameter so as to minimize usage of paths with intersections. Izumi et al. (2015) attempt to select multipath routes dynamically based on available network resources.

The approach forwards flow data into multiple routes from the source to the destination based on utilization rate of the network for every route from the source to the destination.

Braun and Menth (2015) presented a congestion managements solution that dynamically load balances traffic whenever there are temporary network overloads. In the multipath framework, every flow has a primary path through which all the traffic is transmitted during normal traffic, and a backup path that is used only when there is network failure or traffic overload. When there is failure or congestion in the primary path, traffic gets redistributed to the backup paths to minimise the congestion. The strategy relies on the use of network monitoring and OpenFlow's fail-over mechanisms.

Zhang et al. (2014) proposed multipath transport framework based on application-level relay (MPTS-AR), an application layer multipath transport framework designed to improve the utilization of network resource, as well as to increase the network throughput and reliability. The mechanism selects the best combination of multiple paths by taking into account routing costs and the traffic load in the paths. The mechanism's operational goal is to compute a path whose relative performance meets a given minimum threshold, but also balances the overall network traffic in the most efficient way possible.

Many of the SDN multipath traffic engineering approaches do not allow aggregated bandwidth since they are based on flow hashing (Jo et al., 2002), an approach that forces all packets belonging to a flow to follow the same path. Banfi et al. (2016) proposed an SDN architecture that would automatically set up multiple forwarding paths based on link delays and bandwidth. By simultaneously using alternate paths, the solution is able to aggregate link capacities and provide higher throughputs. The solution also includes a packet reordering mechanism that resequences out-of-order packets before delivery to the destination end host.

3.2.7 SDN in NRENs

SDN has become an attractive choice for regional research and eduction networks, particularly due to its ability to support rapid deployment and testing of new network functionalities, as well as for enabling flexibility and automation of operational processes. In Europe for example, the XIFI project (Escalona et al., 2013) deployed backbone connectivity network to leverage the pan-European NRENs' infrastructures, and has been used to advance SDN solutions across NREN sites, while guaranteeing quality of service requirements for distributed services. The European research and education network, GEANT, has set out a long-term evolution path towards deployment of SDN in its topology (Ventre et al., 2017). The starting point for such a road-map was the SDN-based implementation for some of its

basic services, including Internet connectivity and a continental facility offering geographical virtual testbeds to the research community. In 2016, GEANT started to used SDN to provision its Open service, which was previously being delivered through a set of traditional (non-SDN) Open eXchange Points (OXP). Using the SDN platform, GEANT customers were able to interconnect via Layer 2 circuits. The SDN based service itself was been built on top of the Open Networking Operating System (ONOS), allowing for example, an operator to remotely deploy, monitor and manage services. The infrastructure was also able to automatically manage network events and adapt to network changes. GEANT also deployed a multidomain capable SDN-based solution for Bandwidth on Demand (BoD) services (Mendiola et al., 2016a). The solution was designed to provide resilient Layer2 services by taking into account bandwidth and VLAN utilization constraints across OpenFlow and non-OpenFlow domains. The solution also enabled selection of optimal intra-domain paths during the process, with traffic being re-routed to alternative pre-computed paths in case of link-failure.

At a global scale, the SDN-based Global Environment for Networking Innovation (GENI) (Berman et al., 2014) has been deployed as a distributed virtual laboratory for scalable experiments in network science. GENI enables a wide variety of experiments, including for SDN implementations, protocol design and evaluation, distributed service offerings, social network integration, content management, and in-network service deployment. GENI was built using a federated network model where resources are owned and operated by different networks. The federated model allows multiple domains to provide a coherent facility and provides a framework for establishment of mutual trust and collaboration. At the same time, each resource provider maintains local autonomy and is able to set policies for use of its resources.

3.2.8 Reinforcement Learning for Traffic Engineering

Reinforcement learning algorithms have been explored for the Internet routing for a number of years (Boyan and Littman, 1994; Choi and Yeung, 1996; Peshkin and Savova, 2002b; Wang and Wang, 2006; Haeri, Arianezhad, and Trajkovic, 2013; Haeri et al., 2013; Peshkin and Savova, 2002a; Desai and Patil, 2015). Prominent reinforcement learning implementations are based on the Q-learning algorithm, in which a learning agent learns to adjust path selection policies based on experience and rewards, and through continuous modification of action utility values (Xu, Zuo, and Huang, 2014).

A multipath network topology can act as a multiagent RL system, in which groups of forwarding devices, such as SDN switches and routers, can act on a set of multipaths in a shared environment (Busoniu, Babuska, and De Schutter, 2008). In such a system, the forwarding devices can be as modelled as being part of multiagent reinforcement learning system where the Markov decision process in which the state transitions are the result of

joint actions of all the agents (Chun et al., 2014). The Q-function as well as the rewards are also conditioned on a joint policy and set of actions (Zhang and Lesser, 2013).

An important element of multiagent RL systems is that the agents need to be able to coordinate (Guestrin, Lagoudakis, and Parr, 2002). This is because the utility of actions of individual agents also depends on the actions taken by the other agents, such that achieving the intended joint utility function requires that there be mechanisms to ensure that all agents' actions are mutually consistent (Guestrin, Lagoudakis, and Parr, 2002). A common multiagent RL approach has been the distributed value function (Zhang and Lesser, 2013), where each agent estimates Q-values based on short-term rewards, as well as on information received from neighbouring agents. To compute the distributed Q-value, each agent exchanges its highest Q-value associated with the current state, and each agent iteratively updates the Q-values based on the immediate reward received as well as the potential reward at the next state as received from the neighbours (Guestrin, Lagoudakis, and Parr, 2002).

In coordinated approaches, the agents are only aware of their individual states and actions, but the Q-value function for each agent in the multiagent set incorporates a global reward (Zhang and Lesser, 2013). One approach for implementing a global Q-value is to decompose the global Q-function into local Q-functions that only depend on the actions of individual agents (Busoniu, Babuska, and De Schutter, 2008; Russell and Zimdars, 2003). For example, multiple network paths between a source and destination can be represented with local Q-functions, such that the end-to-end Q-function $Q(s,a) = Q_1(s,a_1) + Q_2(s,a_2) + ... + Q_n(s,a_n)$, where n is the number of hops in each path. The joint Q-value can thus be maximized by maximizing local value functions and aggregating their solutions.

A coordinated and distributed multiagent RL system also requires mechanisms for monitoring agents' actions and assessing state of the environment (Guestrin, Lagoudakis, and Parr, 2002). Furthermore, to ensure consistency of the individual agents' Q-functions, there is need to ensure that the perceptions of all the agents are the same (Zhang and Lesser, 2013). One way of achieving consistency is by enabling communication between the agents so that they are able to exchange useful information, such as their perception of the environment's state, Q-value tables, as well as action choices (Zhang and Lesser, 2013). Furthermore, a coordinated multiagent system requires a way to achieve distributed constraint optimization so as to ensure that the selection of actions across all the agents results in overall optimal system performance (Busoniu, Babuska, and De Schutter, 2008). Such distributed constraint optimization also requires communication among agents.

One example of a communication based multiagent learning is implemented by Lee, Viswanathan, and Pompili (2016) for an emergency networking solution, where learning is based on knowledge sharing among agents of different ages in an ad hoc mesh topology.

Knowledge from experienced agents is transferred to younger agents, employing bootstrapping and selective exploration to speed-up the learning process. Bootstrapping is where new agents are initialized with knowledge from older agents that already have experience in the given environment, whereas selective exploration refers to new agents being able to avoid exploring already infeasible states by making use of guidelines obtained from older agents.

A number of approaches have been used to deal with the communication requirement in multi-agent systems. Self-organization approaches (Zhang and Lesser, 2013) allow agents to dynamically identify beneficial coordination sets comprising agents with whom to coordinate with in different situations so as to appropriately trade off performance and communication costs. This allows the agents the ability to dynamically set up coordination networks in a distributed manner, reducing the amount of overall communication without significantly compromising the learning performance. This approach attempts to exploit the notion that, in most cases, agents only need to coordinate with a few other agents to achieve optimal performance. Zhang and Lesser (2013) employ an agent interaction measure to quantify how much utility an agent will potentially lose if it does not coordinate with a subgroup of agents. The interaction measure is thus used to identify the coordination set for each agent. The overall interaction network thus gets decomposed into a set of disjointed sub-networks, thereby reducing the amount communication required.

To deal with the requirement for communication, coordination and information exchange among different radios (agents) in multi-agent reinforcement learning system, Chun et al. (2014) implement a learning model that favours single-agent exploration over simultaneous explorations in multi-agent multi-state reinforcement learning. The implementation uses an exploration scheme to achieve dynamic spectrum access and sharing in wireless communications, where cognitive radios act in a distributed manner to decide the best channels to use for maximum spectral efficiency.

An example of Q-learning is implemented for deflection routing (Haeri et al., 2013; Haeri, Arianezhad, and Trajkovic, 2013) and is used for determining optimal output links to deflect traffic flows when contention occurs. In the implementation, each node maintains an implementation of the Q-learning and deflection algorithm, as well as a Q-values table to store accumulated rewards and Q-values for every deflection decision. Each node also maintains a shortest path routing table, with a record for outgoing link and each of the other nodes in the network, resulting in table size of $m(n-1)$, where m and n are the number of outgoing links and number nodes in the network. The complexity of the implementation depends on the number of nodes and number of links in the network, which is a bottleneck in large topologies (Haeri et al., 2013).

3.3 Summary

A number of studies have been carried out to characterize Africa's Internet topology. The studies revealed high levels of circuitous routing as well as high latencies for Internet access in Africa. The early measurements were conducted mostly from a single vantage point in South Africa and, therefore, provided a biased view of the topology reflecting only how South Africa is connected to the rest of the continent. Other studies have looked at the impact of the growing number of IXPs in Africa and showed a lack of interdomain interconnection among African ISPs, whose consequence is circuitous paths and high RTTs. While previous studies looked at the African Internet in general, this thesis focuses on the routing and performance between African NRENs. It is important to study the African NRENs given that substantial effort has been made to improve the interconnectivity between NRENs in the continent. Furthermore, this thesis employs a distributed topology measurement mechanism to reduce the bias that results from measuring with only a single vantage point.

A lingering traffic engineering challenge for multipath environments is how to select the best end-to-end path and how to optimally utilize all the available paths. BGP-based multipath solutions attempt to enable the exchange and propagation of multiple AS-level paths between domains. However, implementation of optimal BGP-based interdomain traffic engineering mechanisms remains technically challenging. Furthermore, edge networks may not be aware of the multiple paths to a particular destination. For NRENs, solving the circuitous routing problem requires the ability to discover the multiple inter-NREN paths, as well as having the ability to learn and make use of topology performance in selecting end-to-end paths. LISP is a recent Internet routing architecture that, through the use of a mapping system, enhances visibility of multiple paths for multihomed networks. However, LISP does not inherently provide a performance-based selection of paths, although the same can be achieved through separate mechanisms for network measurement and path ranking. To dynamically and optimally meet the QoS needs of Internet applications, traffic engineering mechanisms need to not only be able to classify traffic types, but also to dynamically reconfigure data forwarding paths. SDN provides networks with the capability to centrally and dynamically reconfigure data forwarding paths. Reinforcement Learning provide opportunity for adapting path selection policies based on observed performance.

This thesis investigates the utility of jointly employing LISP, SDN, and Reinforcement Learning in a federated networks environment, such as the UbuntuNet Alliance. The aim is to enable the networks to discover multiple interconnection gateways, to continually rank outgoing paths based on performance, and to dynamically configure end-to-end paths using SDN.

Chapter 4

Mapping The UbuntuNet Topology

This chapter highlights the overall problem associated with circuitous routing among Africa's NRENs. Specifically, the chapter quantifies inter-continental routing between NRENs in the UbuntuNet Alliance and provides an analysis of the performance of NREN traffic that uses inter-continental and intra-continental links. To study this problem, Internet measurement exercises were undertaken to map the logical topology of Africa's NRENs. Internet measurements were aimed at discovering the logical topology of the African Internet and the education and research networks, with a focus on the UbuntuNet Alliance topology. Firstly, the chapter presents results of topology measurements that were conducted to map Internet routes serving African research networks. The results are presented in the form of interconnectivity between cities and also among NRENs and Internet service providers. In measuring the UbuntuNet, the study included all its member NRENs, regardless of whether or not they were physically connected to the UbuntuNet at the time. This was necessary so as to highlight the performance differences the regional interconnection brought to the NRENs. Performance analysis was conducted on the traffic that traverses inter-continental links in comparison to traffic that is exchanged within the continent.

This thesis also inspired a tangential study on the question of how the UbuntuNet topology can be efficiently and reliably mapped using the existing public network measurement infrastructure, and furthermore, whether building an interactive visualization tool for the topology could effectively and accurately inform NREN users about the structure of the topology. This part of the work was carried out in collaboration with students pursuing a Computer Science honours degree. The students implemented the topology probing tools and visualization interface, while this author designed the overall research strategy, the probing algorithm and the evaluation. Execution of this research component and the results are presented in Section 4.2.3 and Section 4.3.

4.1 Topology Data Collection

The topology of Africa's NRENs continues to evolve with continued infrastructural investments, such as through the Africa Connect Project. As the physical topology evolves, so do

the traffic paths, performance and utilization of the various links in the topology. In order to monitor this evolution, there is need for a platform that can continually collect and display accurate data about NREN topologies in Africa. Additionally, continually monitoring the topology would assist the NRENs to determine short-term and long-term performance bottlenecks in the topology, thereby being able to institute appropriate corrective measures. By generating and displaying accurate topological data, as well as latencies experienced by African NRENs, such a platform could help researchers and NREN decision makers to monitor and evaluate how and where their interconnectivity needs to be improved.

Active measurements are often used for collecting data relating to network topologies, with Traceroute being the most commonly used method for deducing the paths traversed by traffic between networks. This work used active network topology discovery techniques to obtain a logical connectivity map involving African NRENs as well as to characterise the performance of the traffic originating from and destined to African NRENs. In particular, this work performs active Internet measurements using CAIDA's Archipelago well as Ripe Atlas.

When designing a topology measurement campaign, it is necessary to be cognisant of the fact that active topology measurements are costly to the networks in the sense that they introduce additional traffic into the topology, which may not be desirable in networks that are resource constrained. It is for this reason that prominent Internet topology measurement infrastructures, such as Ripe Atlas and Archipelago, assign a cost and limit the number and rate of sending probe packets into the platform. It is also important to take into consideration that network probing packets are generally blocked by routers on the Internet, which means that it may not be possible to evaluate all the possible topology paths. Where probe packets are allowed, the presence of multiple alternate paths and load balancing renders it difficult to evaluate all the alternate paths.

The first measurement study, described in Section 4.1.1, was conducted using the CAIDA Archipelago platform and was aimed at mapping Internet routes serving African research and education institutions. This included topology mapping of traffic from Africa to selected university campuses across Africa. The second measurement study, described in Section 4.1.2, focused on the UbuntuNet, targetting a sample of campuses in all its member NRENs, regardless of whether or not they were physically connected to the UbuntuNet at the time.

To ensure that a more accurate topology map was obtained, a distributed network probing method was used (Shavitt and Shir, 2005). A distributed approach allowed the use of multiple diverse traceroute measurements from various vantage points, as well as to a diverse set of targets located in NRENs in the UbuntuNet Alliance. The challenge, however, wsa that many of the measurement infrastructure have very few vantage points on the African continent, let alone inside the NRENs. For example, on the African continent, RIPE

Atlas has over 100 active probes within the UbuntuNet countries, although only about 19 of these probes were seen to be hosted inside NREN networks. To increase chances of discovering alternate and load balancing paths, the topology discovery process used multiple probing protocols.

4.1.1 Archipelago Traceroute Measurements

The first task for this measurement exercise was to identify the appropriate vantage points. Since the aim was to evaluate routing for traffic that originates in Africa and destined for African research networks, the vantage points had to be those located in Africa. At the time of running these measurements (April 2014), Archipelago had five vantage points in Africa, located in Morocco, Gambia, Senegal, South Africa and Rwanda. Permission was obtained from CAIDA to run several repeated measurements from the five African Archipelago vantage points. Although the five vantage points may not seem like a large enough number of vantage point for a large network such as the African topology, the positive aspect was that the vantage points were located in diverse locations and networks. The vantage points were distributed across Africa in the north (Morocco), west (Gambia and Senegal), east (Rwanda) and south (South Africa).

The next task was to identify the target IP address for the topology mapping. These target IP addresses had to be those university networks from African countries. Since the focus was on the UbuntuNet Alliance, the sample included the 16 countries with member NRENs, as well as the other countries within the Alliance's area that did not have NRENs yet, such as Angola, Zimbabwe, Lesotho and Swaziland. A further seven countries from west Africa, and two from north Africa were also sampled, making a total country sample of twenty-nine (29). Thereafter, two to five public universities from each of the sampled countries were selected as targets of the measurements. The IP addresses used were those of the universities' websites. Geolocation databases, *Whois* and MaxMind *GeoLite*, were used to verify that the selected IP addresses were indeed located in the same city as the universities. Overall, 95 IP addresses from 29 countries were used as targets for the active measurements.

Lastly, Paris-traceroute measurements were conducted from the five vantage points to each of the 95 IP addresses. Using Paris-traceroute was necessary so as to enable discovery of multiple load-balancing paths between the vantage point and targets. The measurements being repeated four times per day for 14 days from 6 April to 20 April, 2014. The traceroute measurements were launched at 00:00, 06:00, 12:00, and 18:00. This resulted in about 56 traceroute measurements per source-destination pair, and in total, there roughly 26.6k traceroute samples.

TABLE 4.1: List of NRENs and AS numbers hosting Ripe Atlas Probes

NREN	AS Number	Country
RENU	327687	Uganda
KENET	36914	Kenya
iRENALA	37054	Madagascar
SudREN	37197, 33788	Sudan
TENET	2018	South Africa
University of Cape Town	36982	South Africa
Rhodes University	37520	South Africa

4.1.2 Ripe Atlas Measurements

Considering that networking probing is costly to the networks in that it introduces additional traffic, an efficient measurement strategy should aim to minimise the number of packets required to complete the measurements. Based on this consideration, this research project spawned on off a parallel study to consider the question of *how to efficiently and reliably collect mapping data for the UbuntuNet topology*. This study was carried out by a student pursuing a Computer Science honours degree and was supervised by the author.

To undertake this study, Ripe Atlas Internet measurement infrastructure was used. The motivation for using Ripe was that it provided a flexible API that allows fine-grained configuration of measurement parameters, as well as allowing programmatic control of the measurements. Furthermore, Ripe Atlas provided the largest number of measurement vantage points within the UbuntuNet topology than the other available platforms.

As of October 2016, Ripe Atlas had over 100 active probes within the UbuntuNet member countries. Unfortunately, only 19 of these probes were located inside NRENs, and these were only in five member countries. To ensure that there was a uniform distribution of vantage points across the NRENs, a random sampling of probes done to avoid biasing the result towards NRENs that had higher numbers of probes. For instance, of the 19 active probes inside NRENs, 9 were in TENET, while most of the others NRENs had no more than two probes. Considering that the level of inter-connectivity and routing behaviour in TENET was more advanced than the rest of the NRENs, over sampling TENET would have caused the aggregate continental traceroute and latency results to be more influenced by the TENET's performance. For this reason, although some NRENs had more Atlas vantage points, a maximum of three were used from each NREN, and study was thus conducted using 14 Ripe Atlas vantage points located in five member NRENs of the UbuntuNet.

The NREN institutions and the AS numbers in which RIPE Atlas probes were selected are shown in Table 4.1.

The measurements were conducted from the selected vantage points to a set of 50 IP

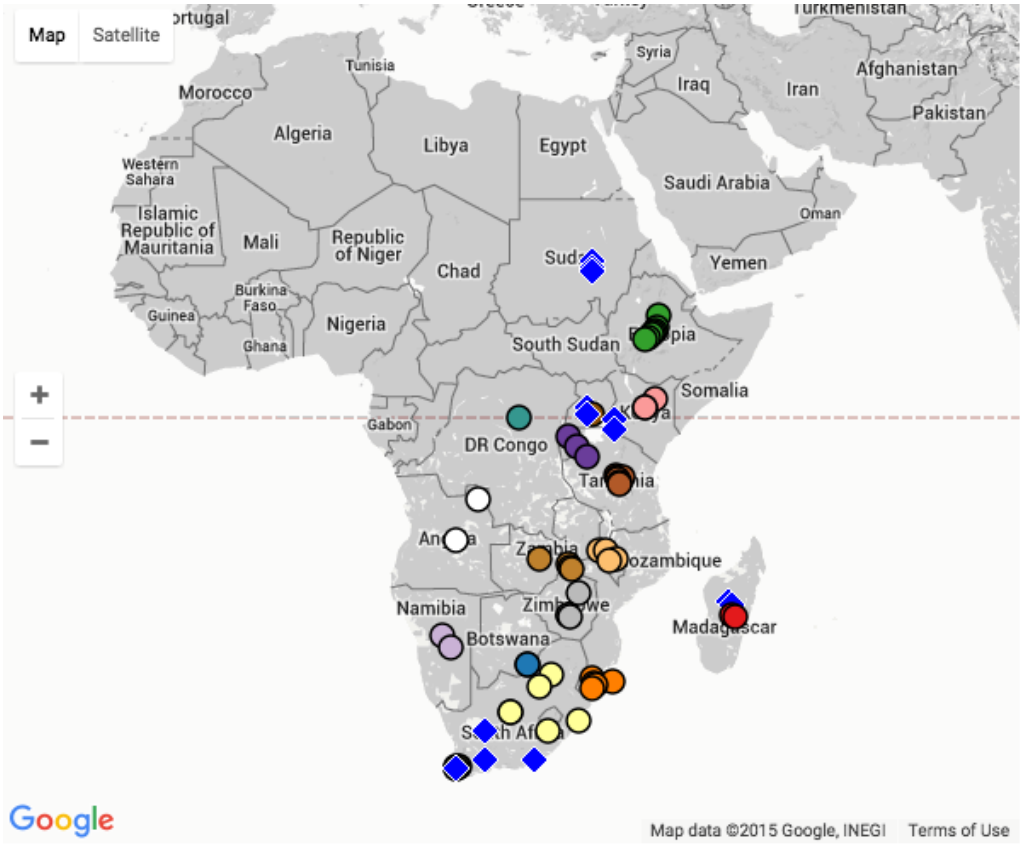


FIGURE 4.1: Colours and shapes represent Atlas probes in different UbuntuNet NRENs and research institutions (KENET, RENU, iRENALA, SudRen, TENET, UCT, Rhodes)

addresses, each one representing a university or a research institution within each NREN in the UbuntuNet Alliance. With 14 Atlas probes as vantage points and the 50 IP addresses as probe destinations, the experiment had in total 700 source-destination pairs.

Furthermore, a reliable measurement campaign needs to include enough redundancy and diversity in the probing to overcome the problem of blocked packets, and to be able to discover paths that may be hidden due to load balancing. For this reason, each measurement cycle was repeated with each of the three protocols: TCP; UDP; and ICMP.

The probes and the destinations used for the traceroute measurements are depicted in Figure 4.1. Probes are represented by diamonds while destinations are represented by circles. The colour of the circle indicates the AS in which the IP address is located.

In total, eighteen Paris-traceroute measurements were conducted for each source-destination pair. The measurements consisted of six ICMP-based measurements, six TCP-based measurements and six UDP-based measurements. As the results are collated, it is possible to piece together the end-to-end path metric for each measurement.

4.1.3 Efficient Ripe Atlas Measurements

The reason why traceroute measurements generate a lot of packets is that the mechanism is iterative in nature, transmitting from source, at each cycle, a probe packet with a higher TTL until the destination is reached. For a complete measurement, traceroute sends as many packets as the number of IP hops on the path from source host to the destination host. The exhaustive probing is required because measurements and topology inference are not coupled, as the process of topology inference is completely separate from the measurement process (Eriksson et al., 2012). Information from prior measurements is not used to inform and optimize subsequent measurements. In standard traceroute-based network discovery experiments, traceroute exhaustively probes every destination address, measuring every link from source to the destination. As a result, some of links get to be probed multiple times by packets that are destined for different destinations. This redundant probing of the sub-paths is wasteful as it does not necessarily provide any new information.

In order to reduce probing redundancy and enhance efficiency in performing measurements, this research employed the Sequential Topology Inference mechanism (Eriksson et al., 2010; Ni et al., 2010; Ni et al., 2008). The mechanism couples topology inference and measurement into one process by exploiting the accumulated knowledge of topology structure to guide subsequent probing. In the same manner, the approach employed in this research coupled topology inference and measurement, analysing the end-to-end paths in order to identify the shared infrastructure and minimize overlapping measurements. This process builds the logical tree structure and leverages the current estimated topology to determine how the next measurement should be performed.

Consider Figure 4.2 where vantage point $ProbeX$ probes multiple target hosts $H, I, J, K, ...$ Through traceroute probing, it is possible to observe similarity between end-hosts, such as in terms of overlapping sub-paths. The paths from vantage point $ProbeX$ to the set of destinations share the first two hops, up to hop G. For measurements undertaken at about the same time, it is of no benefit in having all traceroute measurements redundantly measure the first two links (ProbeX -> B and B -> G). It is sufficient in such a scenario to let only one of the measurements probe the first two links, and for the rest to only start measuring at hop G where the paths diverge.

In the same manner, in Figure 4.3, if a set of measurements from multiple vantage points M, O, P, Q to the same destination Y converge at some hop V, then it is not optimal for all measurements to probe beyond V, particularly if the measurements happen at about the same time. It is sufficient for just one of the measurements to probe all the way to the destination Y, while the rest can terminate at hop V.

With the shared infrastructure clustering and the overlapping matrix of source-target pairs, it was possible to configure the experiments such that certain hops are skipped in the probing. This has the potential to reduce the number of probe packets necessary to resolve

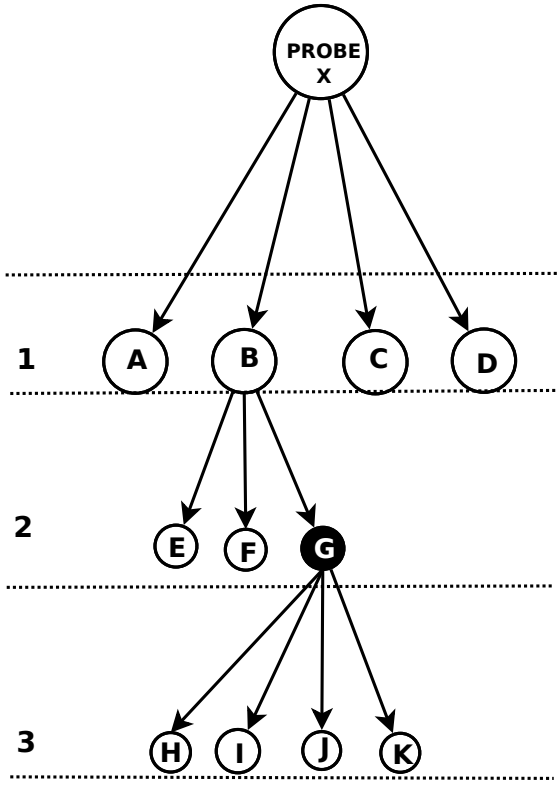


FIGURE 4.2: Example n-ary tree created to find overlapping paths at beginning of traceroute measurements

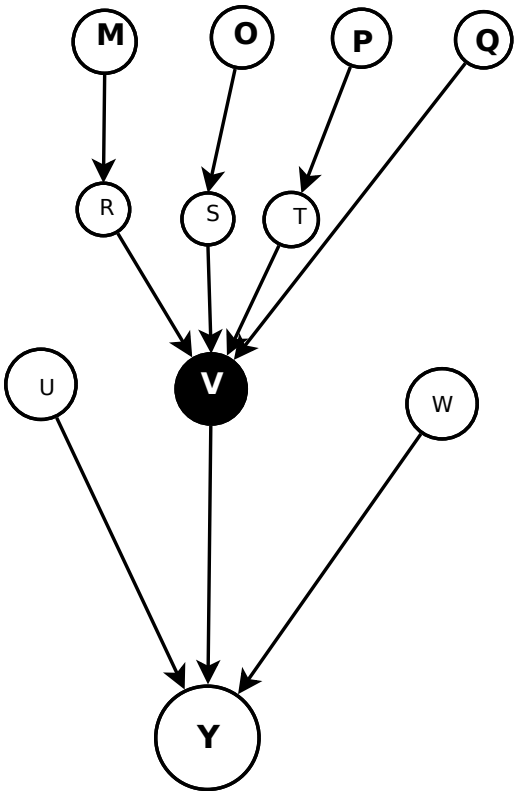


FIGURE 4.3: Example of overlapping paths at the end of two traceroute measurements

the end-to-end logical topology. In default traceroute, each measurement starts from the first hop (TTL as 1), and probes until the destination is reached.

In Figure 4.2, traceroute packets from $ProbeX$ going to target hosts H, I, J, K all go through J. Thus, (H, I, J, K) constitutes elements of a $ProbeX$ cluster, rooted at hop G. In each cluster, only one measurement probes the entire path from vantage point to destination. For the rest of the cluster elements, the first hop for the traceroute measurement was set so as to probe only beyond the cluster root. In the example, the initial TTL for subsequent traces was set to 3 so as to skip the first 2 hops - B and G. Each result set is tagged with a cluster identifier, which helps to indicate that the have the same path beginning and to allow for reconstruction of the path information. Ripe Atlas's traceroute implementation allows one to configure the first or last hop for each measurement, and in this experiment, the configuration for each traceroute's initial TTL was setup through a script for launching Atlas measurements.

Similarly, paths from different vantage points to the same target and converged at some hop were identified. In the example in Figure 4.3, measurements from M, O, P, Q going to destination Y go through V. In each cluster, only one measurement probed up to the destination. For the rest of the cluster members, the traceroute was configured to terminate at the point of convergence, ie hop V .

Each result record was tagged with a cluster identifier and was used for merging and reconstructing full path information for partial measurements. At the end of measurements, the skipped hops of each partial traceroute record were added into the record from the full traceroute, matching the full and partial records based on cluster identifiers. This meant that although partial traceroute only probed a subset of the path hops, the final path record would have all the hops from source to destination.

4.1.4 Dataset limitations

One of the challenges with the topology study was the high number of traceroute measurements that would not reach the destination. While all destinations were reachable via some TCP-based measurement, but only 70% of the targets could be reached by ICMP measurements and only 56% of the probe destinations could be reached via UDP measurements. This indicates a significant amount of blocking for network probing traffic within the African ISPs and NRENs, especially for ICMP and UDP based probes. For analysis, only the traces that reached the destination networks were considered. To determine if the destination network was reached, the last responding hop's autonomous system number (ASN) and city was compared with that of the target (i.e. both the City and AS numbers of the target IP and the last responding hop should be the same). In such cases, the RTT for each source-destination pair was taken as the RTT of the last responding hop inside the destination network.

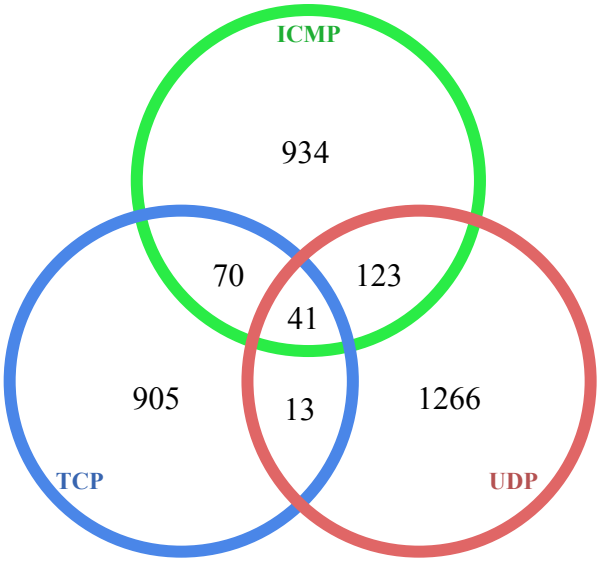


FIGURE 4.4: Venn diagram shows total number of end-to-end IP paths observed uniquely by each protocol, and the paths observed by multiple protocols

After running 18 traceroute measurements for each of the 700 source-destination pairs, a total of 3352 IP paths were observed. ICMP-based measurements traced 1168 paths; TCP-based measurements traced 1029 paths; and UDP-based measurements traced 1443 paths. While some of the paths were observed only by one of the three probing protocols, many others were common to the different protocols, as depicted in the Venn diagram in Figure 4.4. As a result of combining multiple probing protocols, there was an increase in total number of unique paths probed.

Combining the different probing protocols ensured that all of the destinations were probed from every vantage point, resulting in a more complete data set. This demonstrates the importance of not only using multiple vantage points, but also employing different probing protocols to ensure that more probe destinations are reached. Using only a single protocol would leave significant gaps in the topology discovered.

Another challenge is that although the topology measurements were carried out from multiple vantage points, the drawback is that the paths discovered are only forward paths from the vantage points to the destinations. This is the case because Internet traffic is not necessarily symmetric, ie, forward paths are usually not the same as reverse paths (Shavitt and Weinsberg, 2011). The targets in this study were university websites, and therefore very difficult to run reverse traceroute measurements. (This would be possible if one was tracing pair-wise between, say, Atlas vantage probes as they can traceroute to each other). Therefore, the maps obtained from outgoing traceroute measurements are still incomplete. A more complete picture can be obtained by increasing the number and distribution of vantage points (Shavitt and Shir, 2005).

4.1.5 IP Geolocation of Traceroute Hops

Of interest from each of the traceroute measurements was the geolocation of the IP hops in the paths, as well as the round-trip time for each source-destination pair. The IP location mapping and round-trip times were necessary to enable latency comparison for the traffic that was routed through inter-continental links in comparison to traffic that was being routed only within the continent. For this purpose, traces from each vantage point are grouped into two: inter-continental traffic originating in Africa and traversing routers outside Africa; and intra-Africa traffic that got routed within the continent. For the inter-continental traffic, a further interest was to quantify the effect of the inter-continental links (i.e. latency from the vantage point to remote inter-continental gateway) on the overall RTT.

A major challenge on the analysis of the dataset is the inaccuracy of the geolocation information for the IP addresses. MaxMind's free geolocation database - GeoLite2 - is reported to have about 80% accuracy for IP to city resolution (within 40km) for most countries (Shavitt and Zilberman, 2011). For example, the accuracy level reported for South Africa's IP to city database is 71%, whereas for Kenya it is reported to be as low as 55%. However, the accuracy for IP to country resolution, on which route categorisation is based, is higher at 99.8%.

4.1.6 Efficiency in Traceroute Measurements

One objective for running Ripe Atlas traceroute measurements was to investigate an efficient mechanism for collecting topology data in the UbuntuNet Alliance. For continuous or prolonged active measurement, it is important that the measurement mechanisms are reliable and efficient. It is important to minimise the redundant measurement packets considering that active topology measurement mechanisms introduce additional traffic, which can be significant for prolonged monitoring. In addition, topology measurement infrastructures, such as Ripe Atlas, limit the number and/or rate of measurement packets that can be sent. It becomes necessary for researchers to be efficient in how they utilize their measurement traffic quotas. For this purpose, a measurement mechanism that avoids redundant probing of overlapping paths was implemented and tested. The mechanism minimised the number of hops traversed by traceroute packets by skipping hops that might already have been measured by other overlapping measurements.

The mechanism is detailed in Section 4.1.3. Each full traceroutes traverses all the hops of each path that can be traversed from source to destination. Partial traces on the other hand, were based on knowledge gained from prior measurements and were configured to skip the overlapping links at the beginning or end for some of the paths that intersect. Each partial measurement was matched with a full traceroute record based on cluster identifiers. This meant that in the end, each path record had the complete set of hops from source to destination.

Depending on the level of intersection, partial tracing should only probe a smaller overall number of IP hops. Figure 4.5 presents the cumulative distribution for the number of IP hops traversed during the full and partial traceroute measurements. Whereas full traces had a median IP hop count of 17, the partial traces had a media hop count of only 9 (Table 4.2). Results in Figure 4.5 and Table 4.2 indicate that the full measurements, in which path overlaps were not considered, traversed almost double the hops traversed by partial measurements. This also represents a 47% average reduction in the number of packets that had to be sent by an Atlas probe for each traceroute measurement.

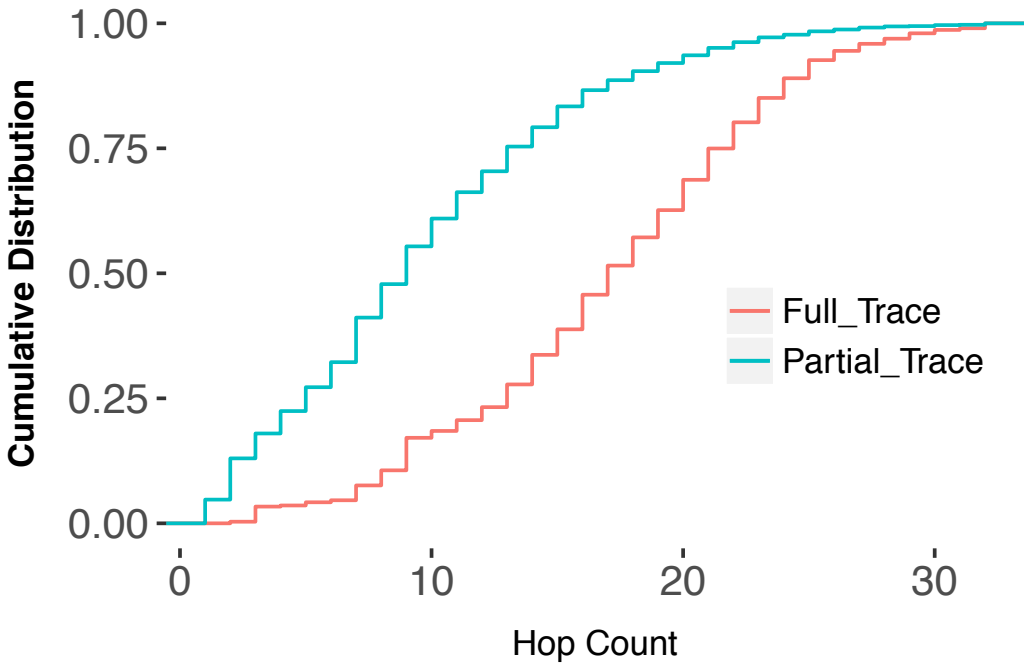


FIGURE 4.5: Hop Count Distribution

Hop Count	Full Measurements	Partial Measurements
Minimum	2	1
1st Quantile	13	5
Median	17	9
Mean	16	9
3rd Quantile	22	13
Maximal	32	32

TABLE 4.2: Hop count for Full and Partial traceroute measurements

The significant reduction in the number of hops confirms the presence of a significant amount of overlapping paths. This is not surprising considering that the IP addresses and the Atlas probes used in the experiments were all located in institutions that share Internet paths through their respective NRENs. For example, traffic from TENET institutions

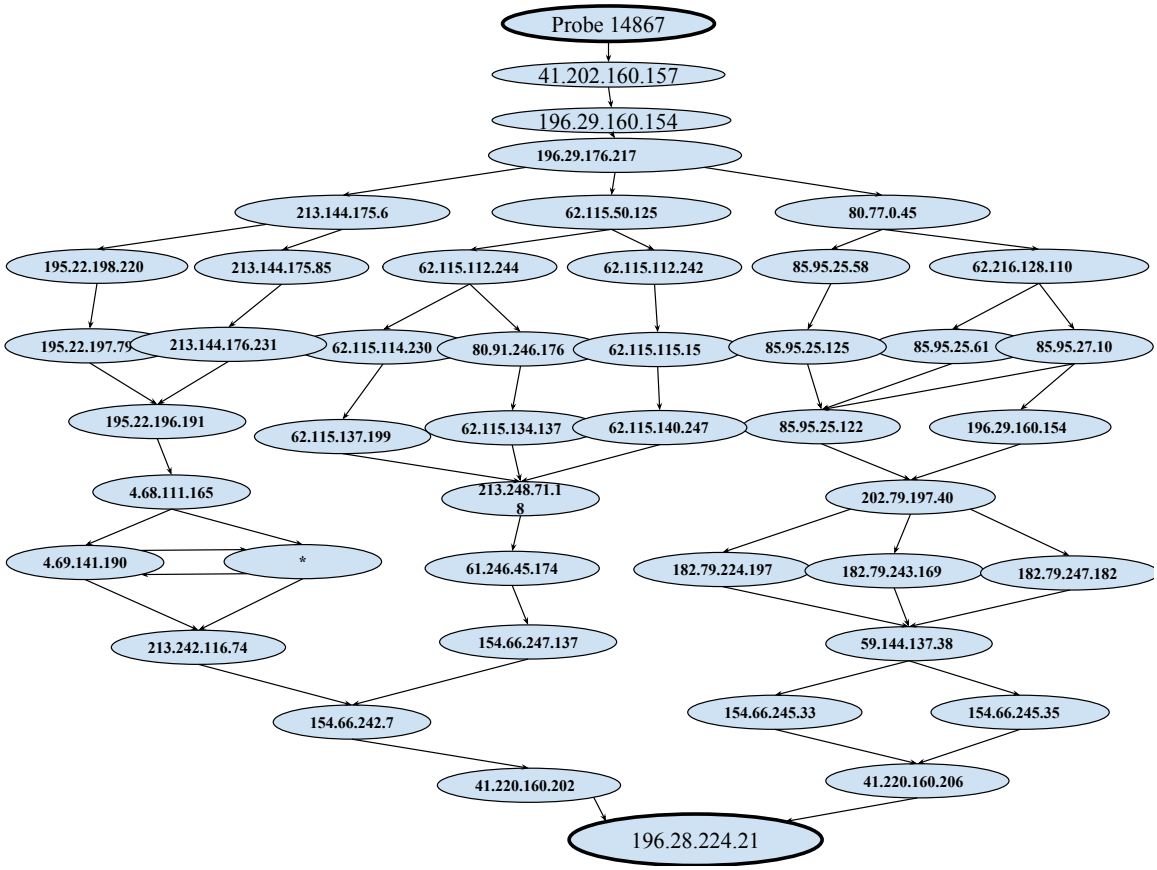


FIGURE 4.6: Example of IP path diversity between Atlas probe 14867 and IP address 196.28.224.21, with nine unique paths

in South Africa, to universities in Kenya's KENET, would generally follow the same path between TENET's gateway in Cape Town and KENET's gateway in Mombasa.

4.2 Pan-Africa NREN Logical Topology Analysis

4.2.1 Path Diversity

A network topology has path diversity if there are multiple end-to-end paths between a source and destination. Figure 4.6 shows for example, that there are nine unique paths from Atlas probe 14867 to destination address 196.28.224.21.

For an internetwork topology, path diversity can be expressed as the average number of unique paths between a source and a destination. Results from probing the African NRENs topology has shown, as is seen in Figure 4.7 and Figure 4.8, that about 70% of the source-destination pairs have more than one IP path; 50% of the source-destination pairs have more than two IP paths. Overall, there is an average path diversity of 3.

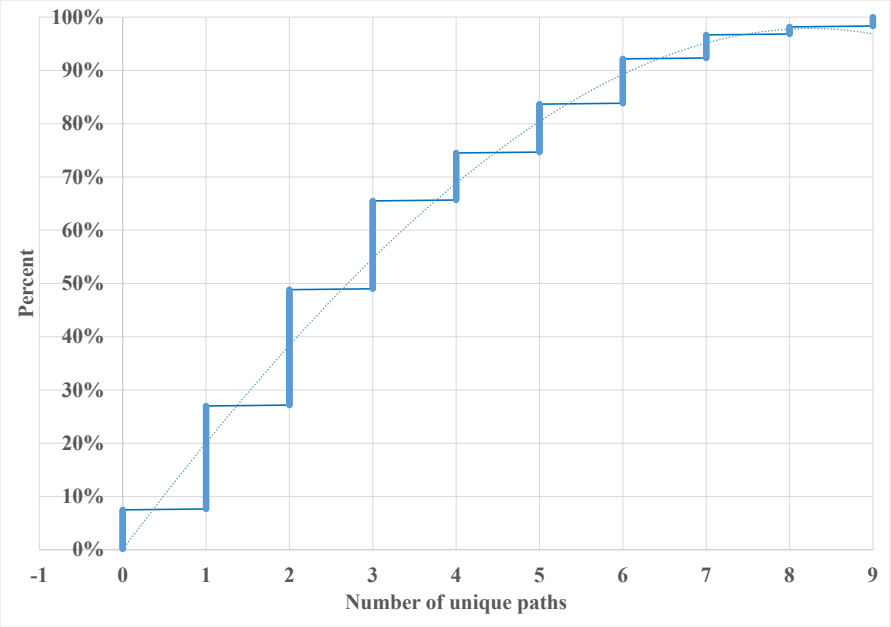


FIGURE 4.7: Distribution for the number of alternate IP paths observed between vantage points and destinations in the UbuntuNet Alliance

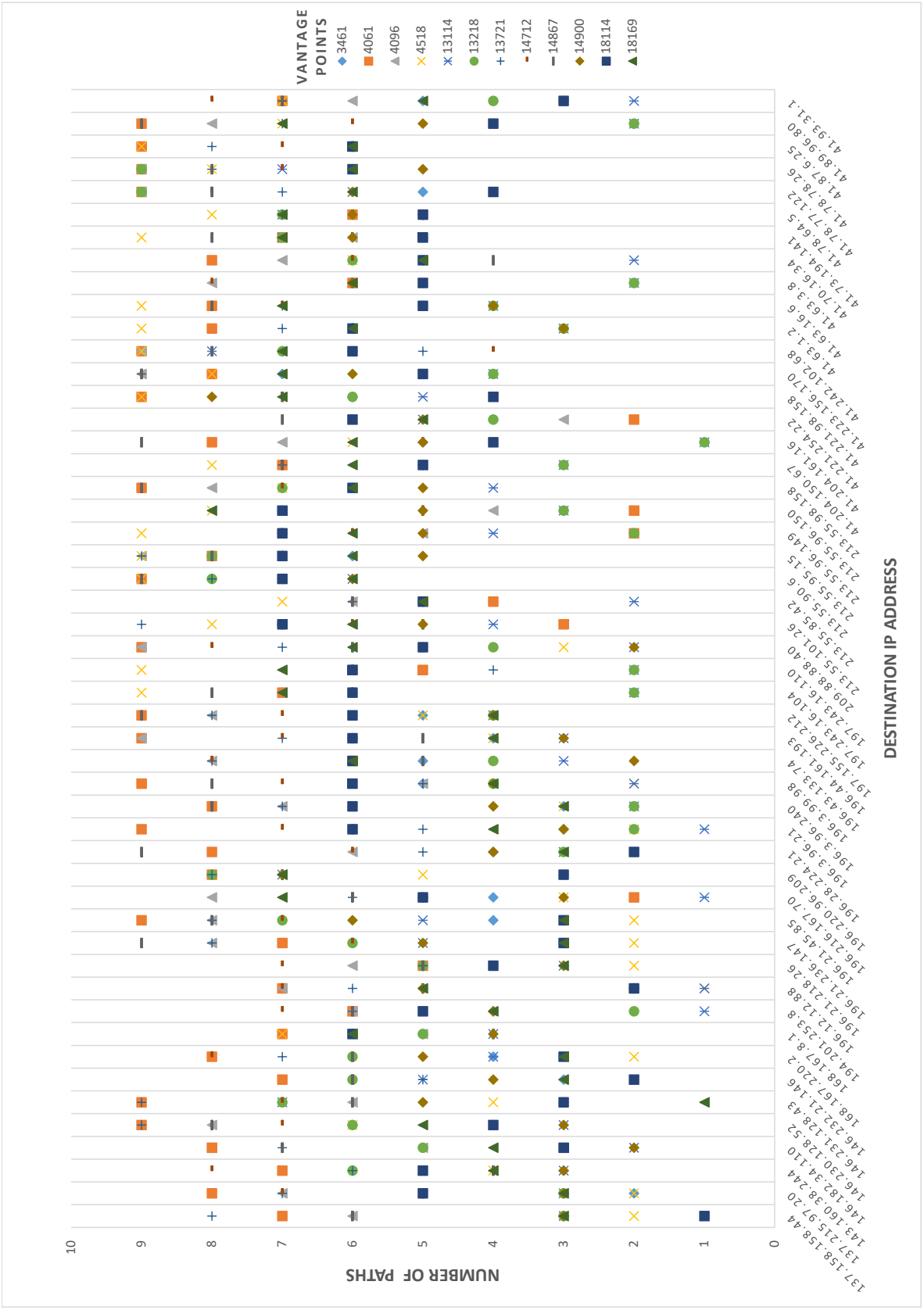


FIGURE 4.8: Number of IP paths observed between Ripe Atlas vantage points and destinations in the UbuntuNet Alliance

4.2.2 Inter-continental Routes

To observe the extent of inter-continental link utilization from each vantage point, traceroute measurements from CAIDA vantage points (Morocco, Gambia, Senegal, South Africa and Rwanda) were grouped by source and destination pairs. From each vantage point, the traces for each source-destination are analysed as follows: starting from the source (vantage point), the next hop and its corresponding RTT is extracted; using MaxMind's GeoIPLite database, each hop's geographical location (City and longitude/latitude coordinates) is obtained; for any hop that was mapped to a city outside Africa, a further manual verification was conducted by comparing the RTT from the vantage to the IP in question, with other IPs in the mapped city; where MaxMind's mapping was doubted, a manual lookup in other mapping databases was conducted, including Whois and OpenIPMap. Traceroutes traces that had at least one intermediate hop being located outside Africa were categorised as inter-continental. For purposes of quantifying the inter-continental RTT from the vantage points, the first hop outside Africa, together with its corresponding RTT, are recorded as the inter-continental RTT for the route.

A key observation from traceroute data was that a larger percentage of traffic originating in Africa and destined for African universities got routed through PoPs that are outside the continent. On average, 75% of the traces from African vantage points to African NRENs traversed inter-continental links through PoPs in Europe, such as Amsterdam, London, Lisbon, and Marseille.

However, a wide variation of percentages of inter-continental traffic were observed from the various geographical locations of the vantage points. For example, the vantage points along the north-west coast of Africa used inter-continental links for as much as 95% of the traces, whereas vantage points in central and southern Africa had a relatively lower usage of inter-continental links. From the Rwandan vantage point, 70% of the traceroute traffic used inter-continental links, while the South African vantage point had about 60% of the traffic traversing inter-continental links. The lower usage of inter-continental links by the South African vantage point can be attributed to the direct logical links observed between South Africa and some of its neighbouring countries, such as Mozambique, Zambia and Zimbabwe, as well as due to physical links to East Africa, such as the EASSY submarine fibre-optic cable.

Figure 4.9 shows the PoP-level connectivity map for traffic originating at the five vantage points to the addresses in the sample.

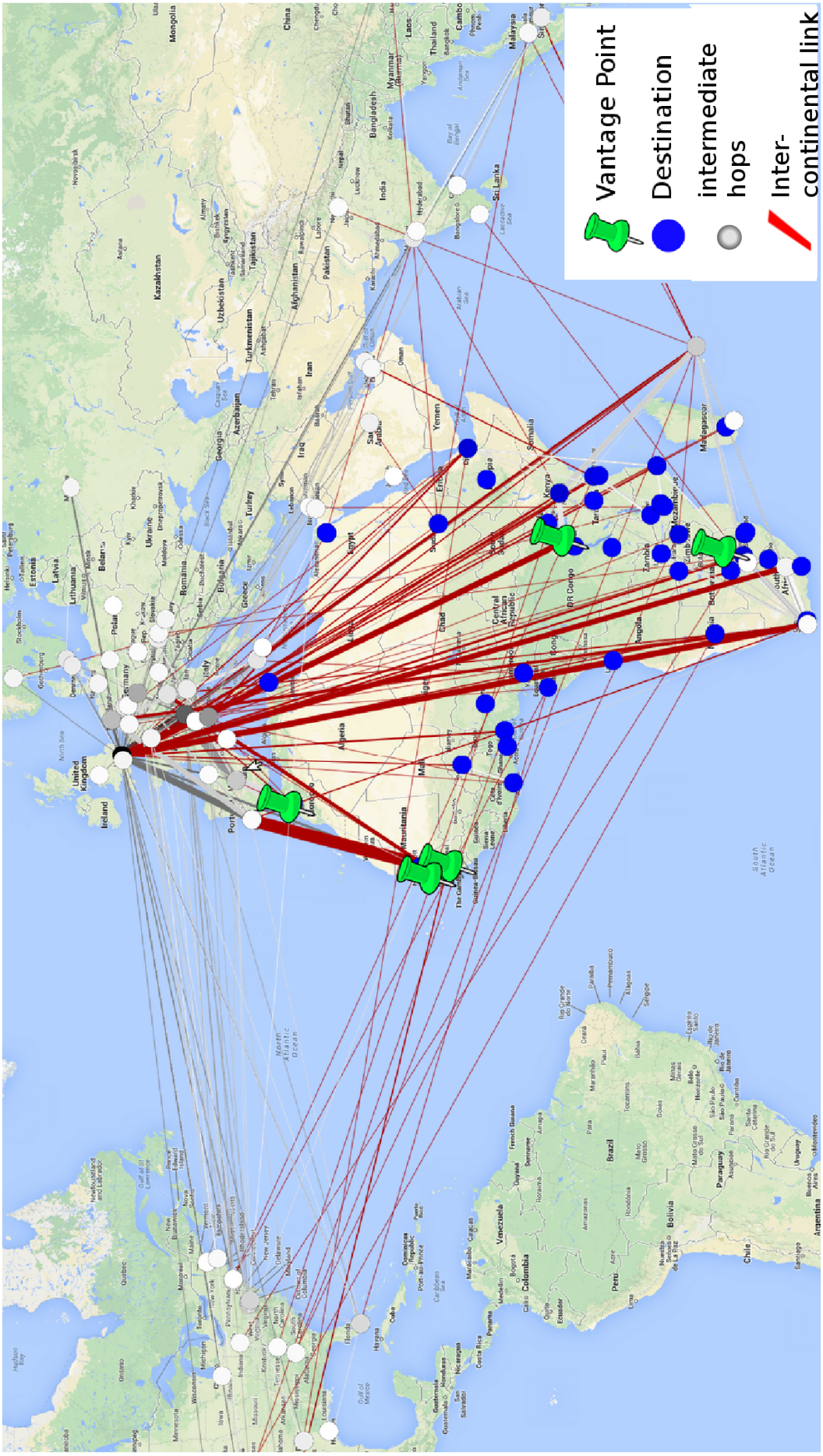


FIGURE 4.9: Logical paths for African traffic, showing logical links interconnecting in Europe and North America

4.2.3 Round-Trip Times

Results from the traceroute measurements show that inter-continental traffic from Africa to African universities experienced RTTs that on average are double those of intra-Africa traffic. As Table 4.3 and Figure 4.10 shows, inter-continental traffic experienced RTTs averaging 409 ms, in contrast to an RTT of 176 ms for traffic that did not leave the continent (intra-Africa). Half of the inter-continental traffic experienced RTTs ranging between 265 ms and 563 ms, whereas the intra-Africa traffic had RTTs ranging between 74 ms and 232 ms.

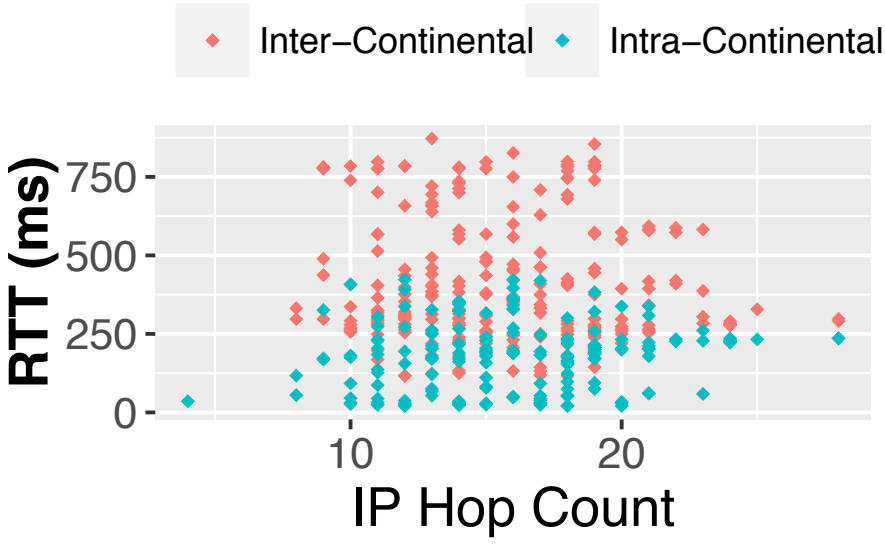


FIGURE 4.10: RTT for intra-Africa traffic and inter-continental traffic

	Intra-Africa RTTs (ms)	Inter-Continental RTTs (ms)
Minimum	20	116
1st Quantile	74	265
Median	188	332
Mean	**176**	**409**
3rd Quantile	232	563
Maximal	423	872

TABLE 4.3: Summary of RTTs for intra-Africa traffic and inter-continental traceroute traffic

The scatter plot in Figure 4.10 further shows there are clear overlaps in latencies experienced by the two sets of traffic. This indicates that, for certain source-destination pairs, better latencies are obtained when exchange points outside of Africa are used while, for other destinations, better performance is obtained when traffic is routed within the continent.

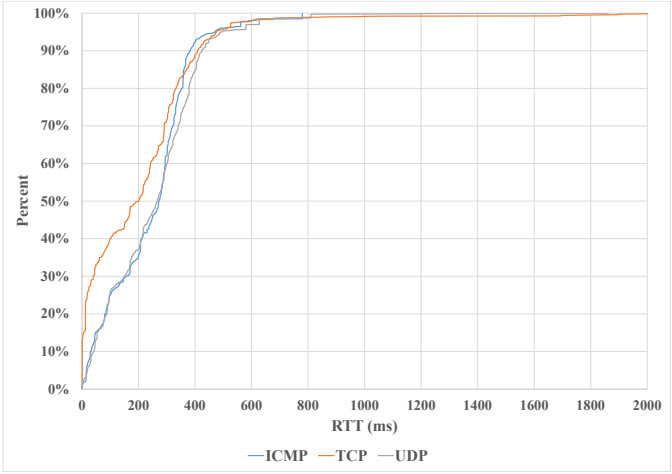


FIGURE 4.11: Distribution of UbuntuNet inter-NREN RTTs based on Ripe Atlas measurements using TCP, UDP, and ICMP

Figure 4.10 also shows that, in terms of IP hop count, both intra-Africa and inter-continental traffic traverses a similar number of IP hops, with half the traffic in either case traversing between 13 and 18 IP hops.

The Cumulative Distribution Frequency graphs in Figure 4.11 show the latencies observed between NRENs in the UbuntuNet. The latencies were measured with Ripe Atlas using TCP, UDP, and ICMP protocols, as described in Section 4.1.2. The cumulative frequency graphs generally follow the same curve for all three protocols. Figure 4.11 further shows that there are more low latencies for TCP-based measurements than for UDP and ICMP-based measurements. About 50% of the latencies for TCP-based measurements were less than 200 ms, compared to only about 35% of the ICMP and UDP-based measurements are less than 200 ms. The lower TCP latencies suggest that ICMP and UDP traceroute traffic is generally given lower priority compared to TCP in the topology. This observation is also analogous to the level of blocking observed for the three protocols (Section 4.1.4), where TCP probes are less likely to be blocked than ICMP and UDP.

About 60% of UDP and ICMP measurements recorded RTTs of less 300 ms (i.e about 40% registered RTTs of over 300 ms). For TCP traffic, 70% of the traffic experienced RTTs of less than 300 ms (i.e only about 30% experienced more than 300 ms). Round-trip times of over 300 ms are in the range of RTTs observed for intra-Africa traffic that is circuitously routed through Europe (Section 4.2.2). This does suggest that a substantial amount of inter-NREN traffic is still being circuitously routed. Some of the RTTs observed are quite high, such that for all the three protocols, at least 10% of the traffic experienced round-trip times of over 400 ms.

The high latencies observed between African NRENs suggests that there is still need for routing strategies would reduce inter-NREN latencies. Furthermore, given the topology's path diversity observed in Section 4.2.1, there is potential for implementing better traffic

engineering solutions that should be able to dynamically identify and configure low latency routes between UbuntuNet NRENs.

4.2.4 Impact of inter-continental latency

Round-trip times, as well as geographical distribution of routes for inter-NREN traffic in Africa reveals a lack of peering among the African service providers. Despite a growing number of national IXPs in Africa, there is still limited interconnection at regional IXPs to facilitate cross-border peering. As a result, inter-NREN traffic in sub-Saharan Africa follows circuitous inter-continental routes, resulting in much higher latencies compared to traffic exchanged within the continent.

When RTT is analysed from each vantage point, the latency of the inter-continental link connecting Africa to other continents appears to have a significant impact on the overall RTT. On average, the RTT from the vantage points to the remote gateways (ie the first hop outside Africa) is about 50% of overall RTT obtained for inter-continental traffic (Figures 4.12 and 4.13). The RTT to just the remote gateways is also about the same as the average total RTTs obtained for traffic exchanged within Africa.

Figure 4.12 shows that different vantage points experienced different degrees of delay on the inter-continental link, and this appeared to largely depend on each vantage point's proximity to European exchange points. For example, Morocco, which is closest to Europe among all the vantage points used in this study, had an average RTT of 60 ms to the remote gateways, whereas the South African vantage point had an average RTT of 250 ms to the European gateways.

In general, vantage points with a higher inter-continental link latency obtain higher overall RTT. The number of IP hops did not appear to be a distinguishing factor in the overall RTT. This result shows that the high latency for inter-continental traffic is significantly increased by the delay on the inter-continental link itself, rather than the number of hops that are traversed outside the continent. For example, traffic from southern Africa to southern Africa, routed via London, covers a distance of roughly 30,000km (the West Africa Cable System fibre-optic cable from Cape Town to London is about 14,530 km long). Given that the cable length from from Cape Town to London is roughly 15,000km, and about the same length from London to Nairobi, then one the way packet distance is about 30,000km. The round-trip distance (ie Cape Town > London > Nairobi > London > Cape Town) would be almost 60,000km. Given the light speed of 200km/ms in fibre cable, this translates to a theoretical RTT of about 300 ms. The observed RTT for this round-trip is around 370 ms, which suggests that about 80% of the RTT in this case is due to the distance factor alone. This shows that the physical length of the transmission medium used by the packets (linearised path) has a significant contribution to the RTT.

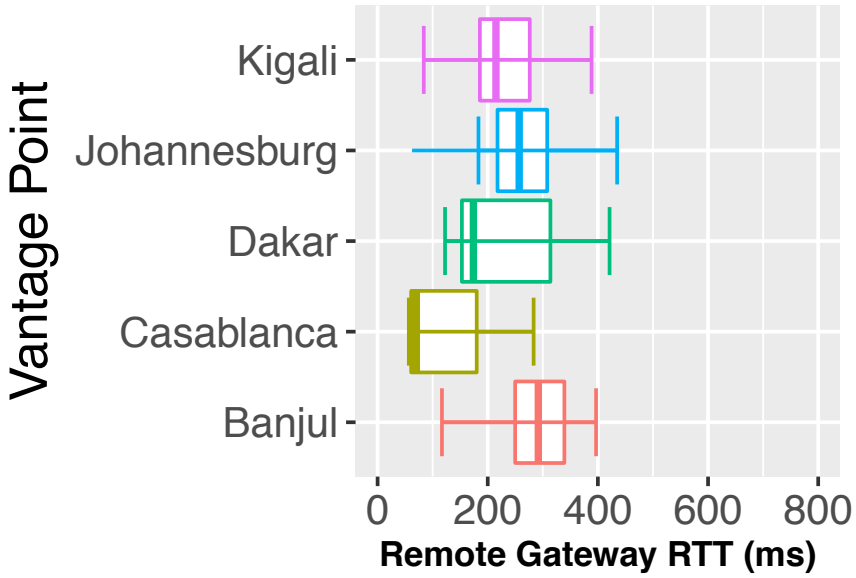


FIGURE 4.12: Round-trip times from African vantage points to remote inter-continental gateways

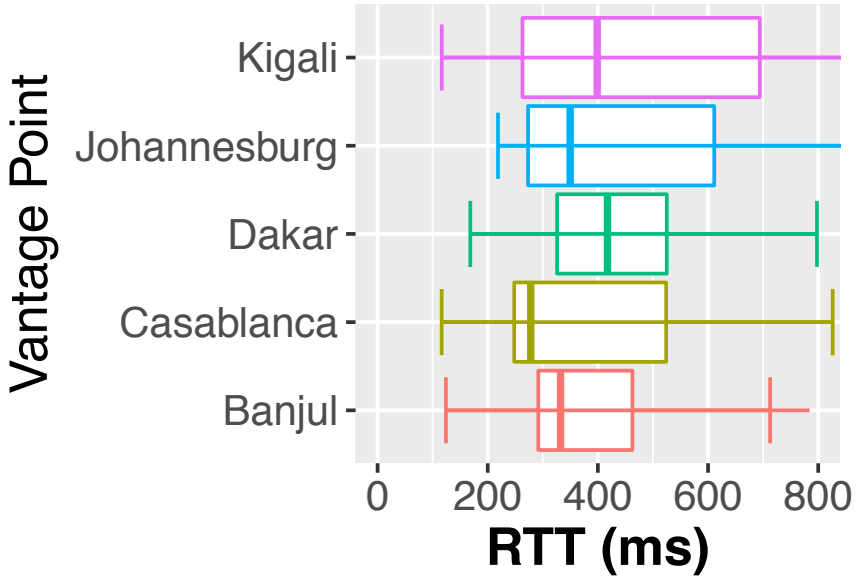


FIGURE 4.13: Total round-trip times for intra-Africa traffic that is routed through inter-continental gateways

The effect of inter-continental routing is demonstrated further in the case of universities within the same country that achieve remarkably different RTTs due to routing differences. For example, in Kenya, one university had its traffic from Johannesburg, South Africa routed via Amsterdam (AIMS-IX), then back to South Africa (CINX, Cape Town) before being forwarded to Kenya, and experienced an average RTT of about 400 ms. In comparison, another university in the same country (Kenya) had a direct logical link from Johannesburg to Kenya and achieved an RTT of only 80 ms. It was further observed that traffic to countries with no direct fibre connection from the vantage points experienced much higher latencies. For example, traffic from Johannesburg to Malawi was routed first through London, then Maputo, Mozambique, before being forwarded to Lilongwe, and experienced average RTT of around 380 ms.

In contrast, lower latencies were observed between countries that shared direct fibre cable system. For example, the Zambia NREN (ZamREN) had a direct connection to the Johannesburg IXP where it peered with the UbuntuNet alliance. Using the Johannesburg link, traffic from South Africa to ZamREN experienced a low latency of 55 ms. Other destinations that recorded low RTTs from the South African vantage point were in the neighbouring countries such as Mozambique (45 ms) and Namibia (80 ms). These countries had direct fibre links to South Africa. Futhermore, where functional NRENs were present and traffic was being routed locally within national IXPs, much lower latencies were observed. For example, within South Africa, members of South African National Research Network (SANReN) are linked though a fibre-optic backbone and exchange traffic locally at national IXPs - the Johannesburg Internet Exchange (JINX) and the Cape Town IXP (CINX). In the experiments, traffic from the vantage point located within the SanRen, to other SanRen members, achieved an average RTT of 20 ms.

4.2.5 Inter-NREN AS-Level Topology Analysis

Autonomous System Numbers (ASNs) are numbers that are used to uniquely identify routable networks on the Internet. These ASNs are administered by regional Internet registrars and are associated with specific countries and continents. Furthermore, in order to facilitate the exchange of routing information between autonomous systems, IP addresses are mapped to ASNs.

To understand the AS level interconnectivity of Africa's research and eduction institutions, IP hops in the traceroute data were mapped to their corresponding ASNs using the geolocation method as described in Section 4.1.5. Where a number of consecutive IP hops are mapped to the same ASN, they are represented in the graph by a single ASN node. Figure 4.16 is the ASN level graph, where each ASN is represented as a node and edges are

[0]http://afterfibre.net/

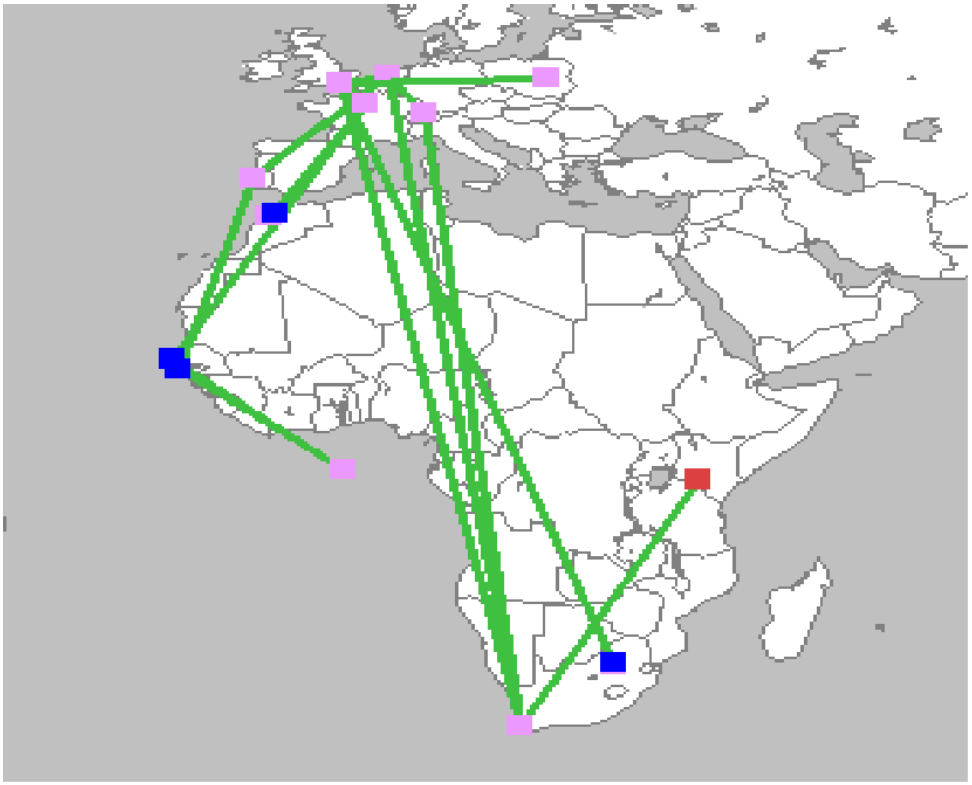


FIGURE 4.14: Traffic from all vantage points (blue squares) including from **Johannesburg, South Africa**, destined to a university in Nairobi, Kenya (red square), being routed through London, then to Cape Town, before being forwarded to Nairobi

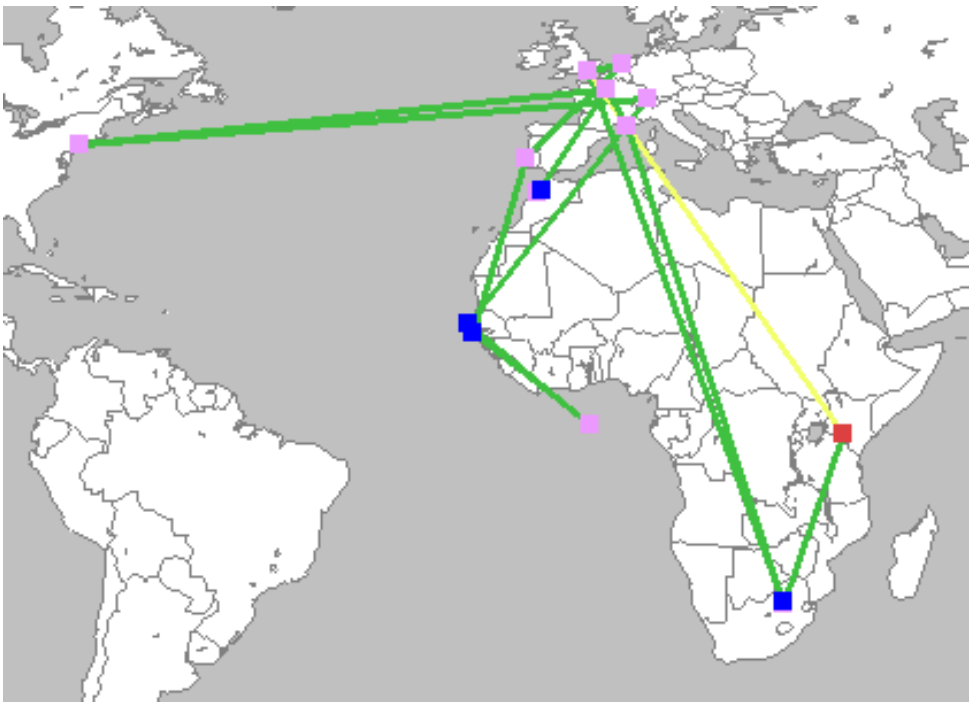


FIGURE 4.15: Traffic from four vantage points (blue squares) destined to a university in Nairobi, Kenya (red square), being routed through Amsterdam and London, then to Johannesburg, before being forwarded to Nairobi. Traffic from Johannesburg is forwarded directly to Nairobi

created if there is direct link between a pair of ASes. There was a total of 100 unique ASes in the topology: 65 registered in Africa; 23 in Europe; 6 in Asia; and 6 in USA. In the AS-level topology (Figure 4.16), the African ASes, represented by blue nodes, appear on the outer parts of the graph, while the European ASes, represented by the green nodes, appear at the centre of the topology.

The observed interconnection among the African networks is largely through European transit networks, such as Cogent Communications(ASN 174), TATA (ASN 6453), Level3 (ASN 3356) and Seacom (ASN 37100). According to information available in the Peering Database (PeeringDB)[1], these European ASes peer at global IXPs in Europe, including at the London Internet Exchange (LINX), Amsterdam Internet Exchange (AMS-IX), and Frankfurt Internet Exchange. There is also high interconnectivity through the South African Internet eXchange (ASN 5713) to the UbuntuNet Alliance (ASN 36944) peering at LINX and AMS-IX.

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

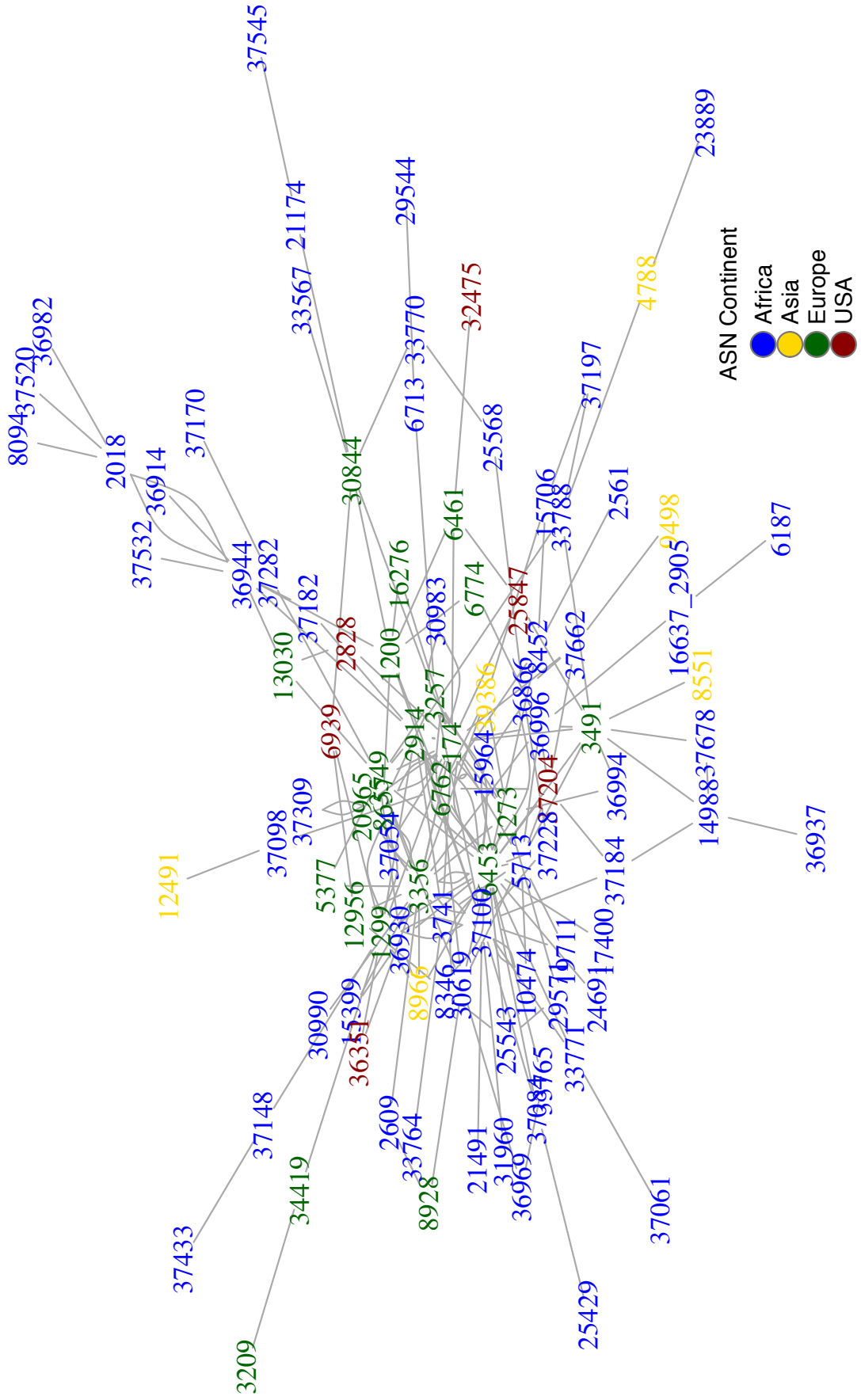


FIGURE 4.16: AS-level interconnectivity map of African vantage points and universities

It is important to note that the AS-level topology dataset described here constitutes both edge networks and upstream/transit providers. These different types of networks should exhibit different properties, such as in terms of observed node degrees and centrality. The African ASes in the dataset are mostly edge networks, being either sources or destinations of the measurement packets. On the other hand, many of the non-African ASes are transit providers. Since the traceroute measurements were conducted between African vantage points and destinations, the expectation would be that only African ASes would be on the edge of the AS-level map. However, it is noted that a few non African ASes are observed on the edge of the graph in Figure 4.16: ASN12491, IPPLANET from Israel; ASN8551, BEZEQ-INTERNATIONAL from Israel; ASN9498 Bharti Airtel from India; and ASN3209, Vodafone from Germany. These foreign ASes are noted to provide IP connected to some African countries through either satellite or cellular networks.

AS-level Node Degree

The AS-level graph showed that there was minimal peering within Africa, as most of the networks had direct AS paths to transit ASes in Europe. This is further indicated by the higher node degrees for the European ASes, in comparison to the African ASes. Figure 4.17 shows the overall degree distribution of the AS-level topology, and Figure 4.18 and Table 4.4 depict the distribution of node degrees for the ASes grouped by the continents. The European ASes had the highest average node degree of 8.65 and an inter-quartile range from 8 to 11, and a maximal node degree of 32. In comparison, the African ASes had a lower average node degree of 2.69, and an inter-quartile rage from 2 to 3, and a maximal node degree of 18. Among the African ASes, the highest node degree of 18 was registered by SEACOM (ASN 37100), followed by the South African Internet eXchange(ASN 5713) with node degree of 9, and UbuntuNet Alliance (ASN 36944) with a node degree of 8.

AS-level Node Centrality

Another way to compare the ASes in the AS-level topology is to consider the centrality of each of the AS nodes (Bonacich, 2007). For this purpose, the *eigenvector* centrality measures (Csardi and Nepusz, 2006) were computed to measure the influence of each node in the network.

Figure 4.19 and Table 4.5 show the eigenvector centrality measures for the AS nodes in the topology, grouped by continents. The European ASes registered the highest average eigenvector centrality measure of 0.28 and an inter-quartile range from 0.17 to 0.43. In comparison, the African ASes had an average eigenvector centrality measure of 0.08, and an inter-quartile range from 0.01 to 0.11. SEACOM (ASN 37100) was the most central among the African ASes, with a eigenvector centrality measure 0.59, followed by the South African

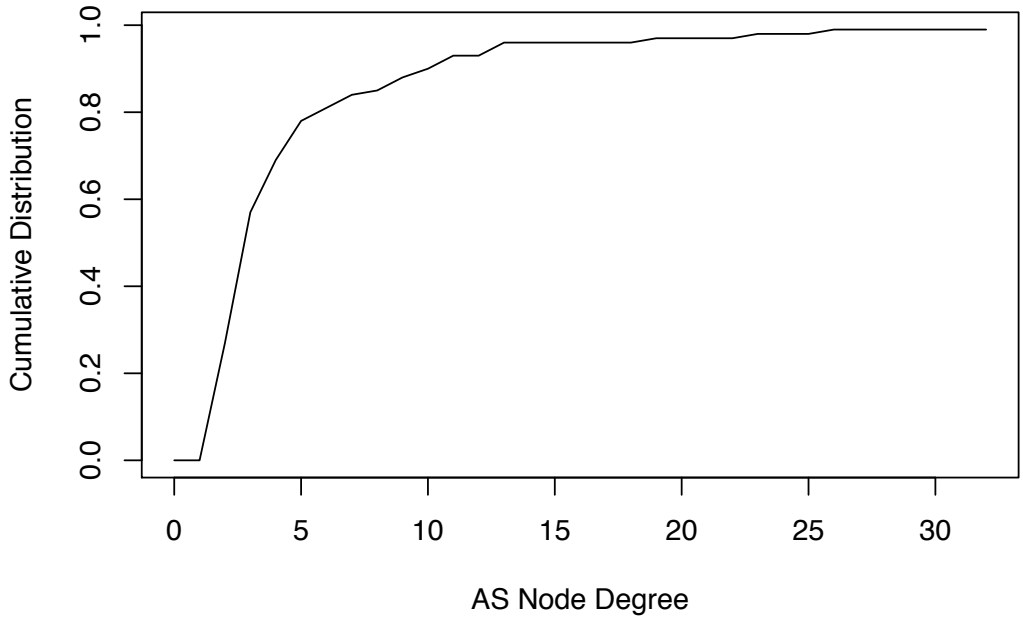


FIGURE 4.17: AS-level node degree distribution

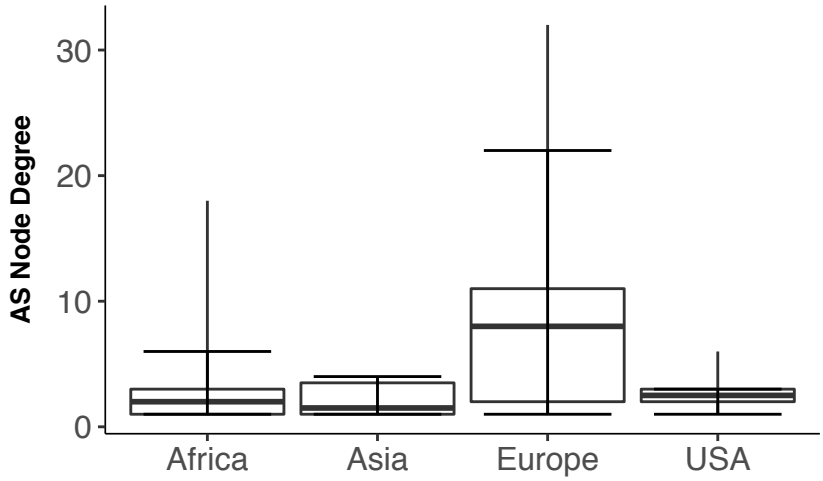


FIGURE 4.18: AS-level node degree distribution

Node Degree	Africa	Asia	Europe	USA
Minimum	1	1	1	1
1st Quantile	1	1	2	2
Median	2	1.5	8	2.5
Mean	**2.69**	2.16	**8.65**	2.83
3rd Quantile	3	3.5	11	3
Maximal	18	4	32	6

TABLE 4.4: AS Node Degree

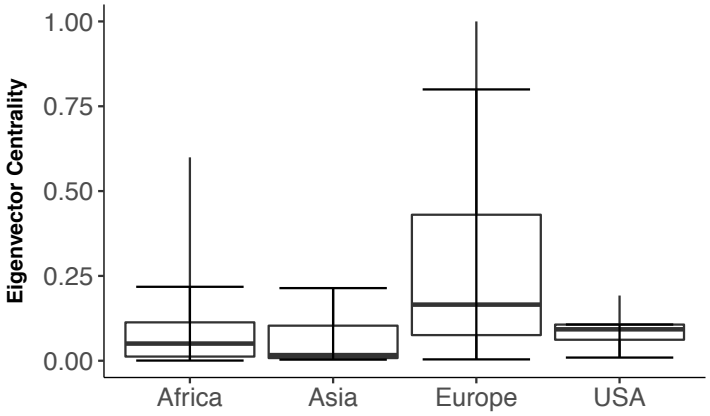


FIGURE 4.19: AS-level EigenVector Centrality

EigenVector Centrality	Africa	Asia	Europe	USA
1st Quantile	0.01	0.00	0.06	0.06
Median	0.05	0.02	0.17	0.09
Mean	**0.08**	0.06	**0.28**	0.09
3rd Quantile	0.11	0.10	0.43	0.11
Maximal	0.59	0.21	1.00	0.19

TABLE 4.5: AS-Level EigenVector centrality measures per continent

Internet eXchange(ASN 5713), which had a centrality measure of 0.28. Overall, TATA communications (ASN 6453) had the highest centrality measure of 1.

The AS-level graph in Figure 4.16 has an overall network diameter of 8, an average AS path length of 3.37, and an average clustering coefficient of 0.180. Whereas an AS path length of 3.37 would suggest a densely interconnected AS-level topology, a low clustering coefficient of 0.180 suggests that, overall, the AS-level topology is sparse. These findings are consistent with the previously reported low peering among Africa based ASes and a high connectivity to the global IXPs (Gupta et al., 2014).

4.3 Traceroute Visualization

Another question that needed addressing while collecting topology data was how the NRENs topology can be presented to enable easier evaluation and understanding by stakeholders involved with NRENs in Africa. This prompted a related study on how best to visualize the UbuntuNet NRENs topology. This strand of research was carried out in collaboration with a student pursuing an honours degree as part of their dissertation project and was supervised by this author. This section describes the design of the visualization tool, and presents the main findings from the study.

The approach undertaken was to design and implement a geospatial visualisation with the aim to study how a topology visualisation can be useful for NREN stakeholders in identifying possible network traffic routes between NRENs, and helping NREN managers to identify incorrect or suboptimal routing and aid peering decision-making processes.

An information visualisation (InfoVis) is the representation of the abstract data on an interactive visual interface (Keim, 2002; Wassink et al., 2009). By presenting topology information in a visual and interactive manner, through an information visualisation (InfoVis), gaps, anomalies, clusters or patterns in the data can be identified (Becker, Eick, and Wilks, 1995; Carr, 1999; Keim, 2002; Shneiderman, 1996). For example, GTrace (Periakaruppan and Nemeth, 1999), a Graphical traceroute tool, uses the InfoVis approach to help in discovering routing loops and in deciding route implementations. Using heuristics, the GTrace system determines the location of a node as the traceroute is executed before displaying the route on a world map as series of nodes (hops) and links. A table displays more detailed information about the traceroute, including the hop number, hop IP Address and hop hostname.This research implemented and tested the GTrace approach to evaluate users' interaction with topology data of the African NRENs.

One of the earliest attempts to visualize Africa's Internet topology was carried out by Gilmore, Huysamen, and Krzesinski (2007), who generated router and Autonomous System (AS) level maps of the African Internet using data collected from traceroutes sent to selected

IP addresses from a vantage point in South Africa. At the router level, a Java-based tool, Terrapix, was created, where 2D and 3D visualisations mapped nodes and links to geographic locations. For the AS level, CAIDA's Walrus tool was used to generate logical node-link graph visualisations in a 3D space. Using these visualisations, a "picture" of the African Internet was developed. One limitation with the map produced by Gilmore, Huysamen, and Krzesinski (2007) was that traceroute measurements were conducted from only a single vantage point, and were thus biased (Shavitt and Weinsberg, 2011).

The main research question that this study attempted to answer was if a *geospatial visualisation does effectively and accurately communicate the network topology of African NRENs*. To answer this question, the study considered whether the potential users of the visualization tool would be able to do the following:

- identify networks (physical and logical);
- see where networks interconnect (location of source, destination and intermediate hops of traceroutes); and
- determine what route network traffic traverses between NRENs (intra-continental vs intercontinental).

4.3.1 Design and Implementation

A User-Centred Design (UCD) approach describes a process in which users are involved throughout the design cycle, including when the needs and goals of users are determined (Wassink et al., 2009; Abras, Maloney-Krichmar, and Preece, 2004). This implies first gaining an understanding of the visual queries (an information need addressed by a visualisation) that potential users may have. Users of the visualisation are envisaged to be NREN managers and network engineers of tertiary education institutions that are part of the UbuntuNet Alliance. In order to obtain user tasks and goals, network specialists and managers working within UbuntuNet Alliance NRENs and research institutions were invited to an online survey.

The online requirements gathering option was preferred due to the difficulty of physically interacting with the potential users who, due to the speciality of their networks, are quite few and geographically dispersed. The survey enquired about current network management operational tools used in by NRENs and metrics of interest. A "lack of comprehensive routing information" was stated as a network management limitation encountered, and network down-time and congestion were cited as common network problems experienced.

Two UCD iterations were executed to analyse, design and evaluate the effectiveness and accuracy of the visualisation interface at communicating the network topology. Each iteration consisted of three phases: the early envisioning phase, the global specification phase

and the detailed specification phase (Abras, Maloney-Krichmar, and Preece, 2004). Each iteration in the design cycle was considered concluded once a specific criteria was reached, such as if users are able to adequately answer visual queries (Wassink et al., 2009).

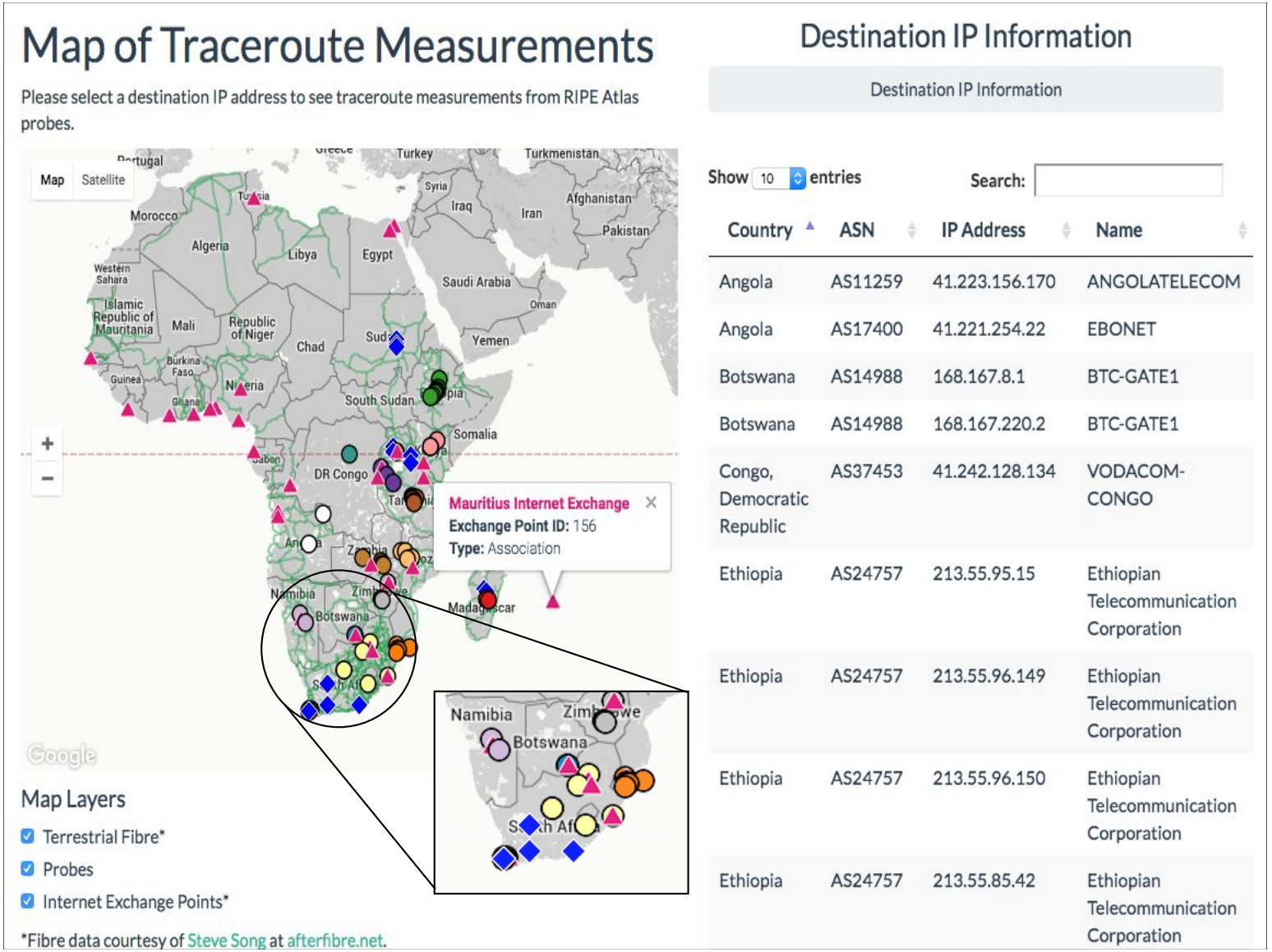


FIGURE 4.20: Visualization prototype capable of displaying layers to include terrestrial fibre, Atlas probes, IXPs, and measurement targets

The design of the visualisation was based on the existing traceroute visualisation tool - OpenIPMap (RIPE, 2015). In addition to the Google Maps Javascript API, the DataTable plugin for jQuery was used. Design of the interface and visualisation flow followed Schneiderman's Visual Information Seeking Mantra of overview, zoom, filter and details on demand (Shneiderman, 1996). The visualisation in Figures 4.20 and 4.23 was developed using the Google Maps Javascript API and tested with traceroute data collected from the Ripe Atlas platform (Atlas, 2015).

The visualisation displays several dimensions of data, including the location of active Internet Exchange Points in Africa, physical fibre cables and collected traceroute information. IXP data was obtained from Packet Clearing House's Internet Exchange Point Directory (House, 2014) while fibre data was obtained from the AfTerFibre Project (the African Terrestrial Fibre Optic Cable Mapping Project) (Song, 2011).

Different symbols were used to represent various types of data on the map, and these features differed on two channels of colour and shape to make items distinguishable from each other. Destination IPs are represented by different coloured circles on the map and colour-coded by country so that users could more easily locate a point of interest (Figure 4.20). Probes were shown on the map as blue diamonds, and target IP addresses were marked with blue circles. Intermediate hops of the traceroute were marked as small green circles and IXPs as pink triangles, as seen in Figure 4.20. A route was illustrated on the map connecting the probe, hops and destination IP markers as a line with an animated arrow indicating the direction of the traceroute.

Users were first presented with an overview of all the target IP addresses on a map (Figure 4.21). When the user hovers the mouse pointer over an icon on the map, an info window with information related to that point of interest is displayed. After clicking a chosen destination IP address icon on the map, multiple traceroute measurements are shown on the map from all available probes (various vantage points) to that particular IP address.

Clicking a row in the table zooms into the related icon on the map. Selecting the checkboxes allows different layers (terrestrial fibre, probes, Internet exchange points) to be added to or removed from the map (Figures 4.21 and 4.22). A search box can be used to filter results in the table and more easily locate points of interest.

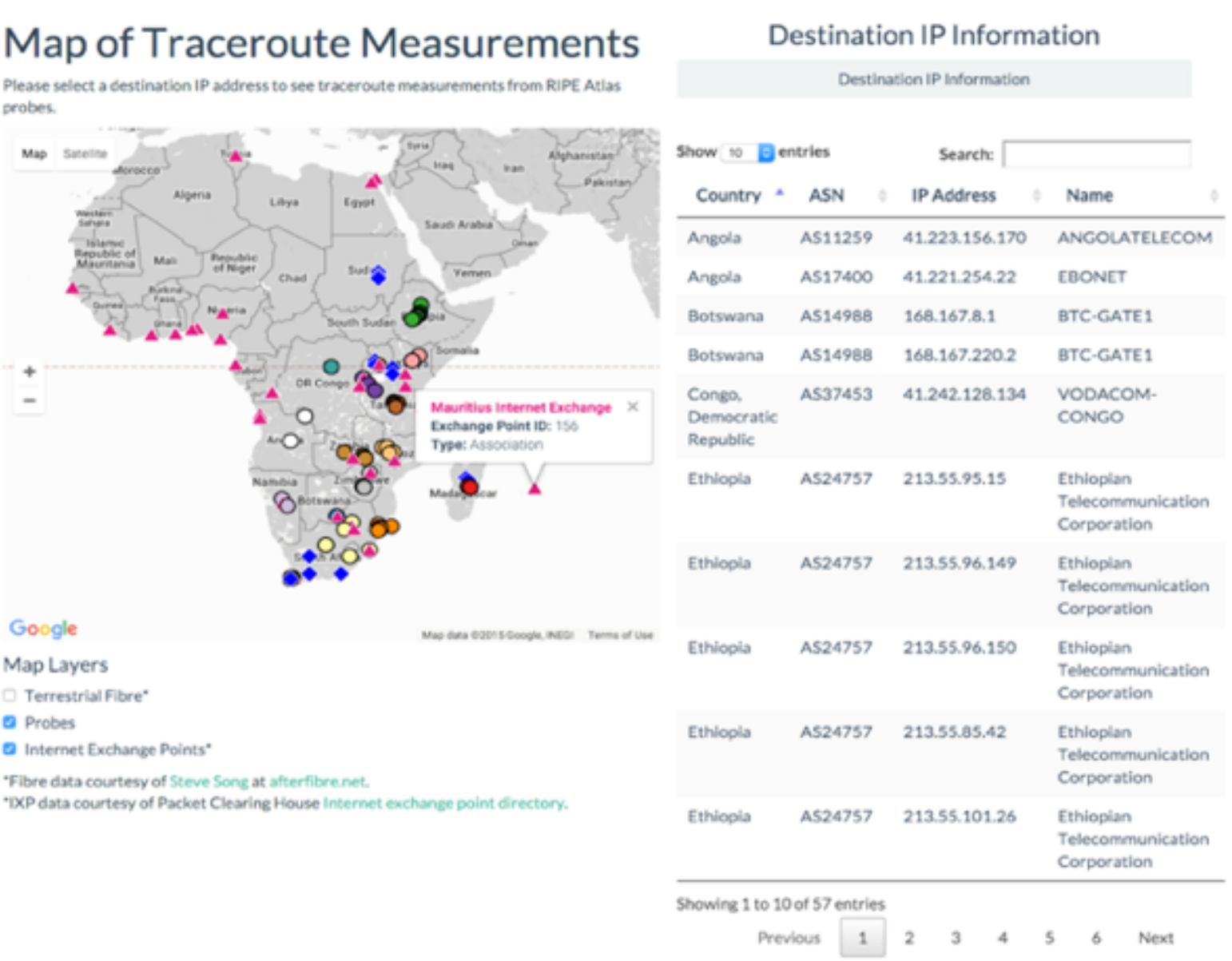


FIGURE 4.21: Initial overview screen with Internet Exchange Point and Probe layers displayed

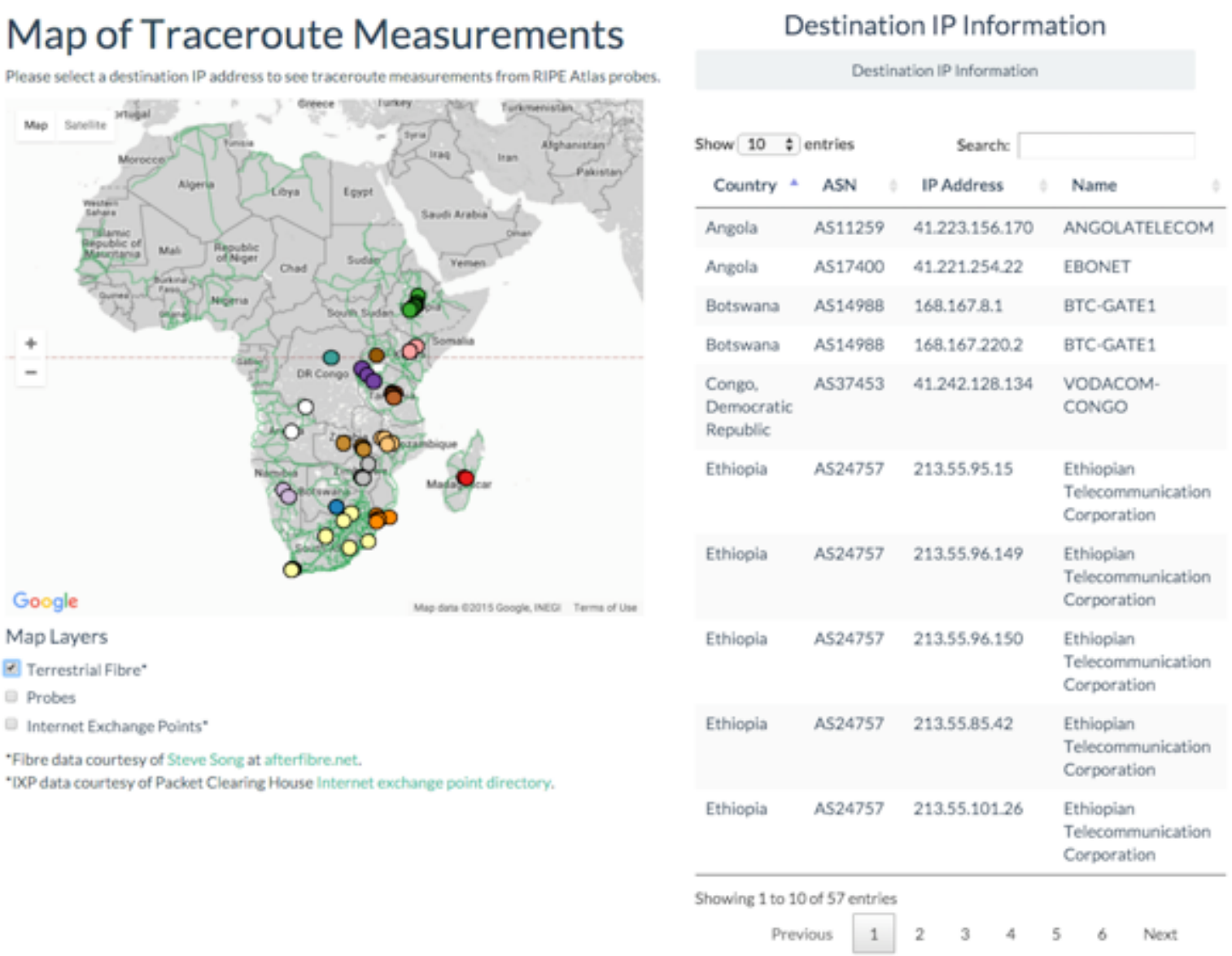


FIGURE 4.22: Checkbox enabled to display the terrestrial fibre cables

MaxMind GeoLite City Database was used to map IP addresses to city-level coordinates. The city level coordinates were then used to position map markers for probes and destination IP addresses. Hops were illustrated on the map as small green circles connected by lines from source to destination, indicating an end-to-end path. An animated arrow moves along these lines to show the direction of the packets (Figure 4.23). Clicking on a destination node allows for viewing multiple traceroutes sent from various vantage points to be seen on the map all at once. Clicking on a specific path allowed the user to view a single traceroute as a red line with animated red arrow. Detailed hop information is displayed in table(Figure 4.24).

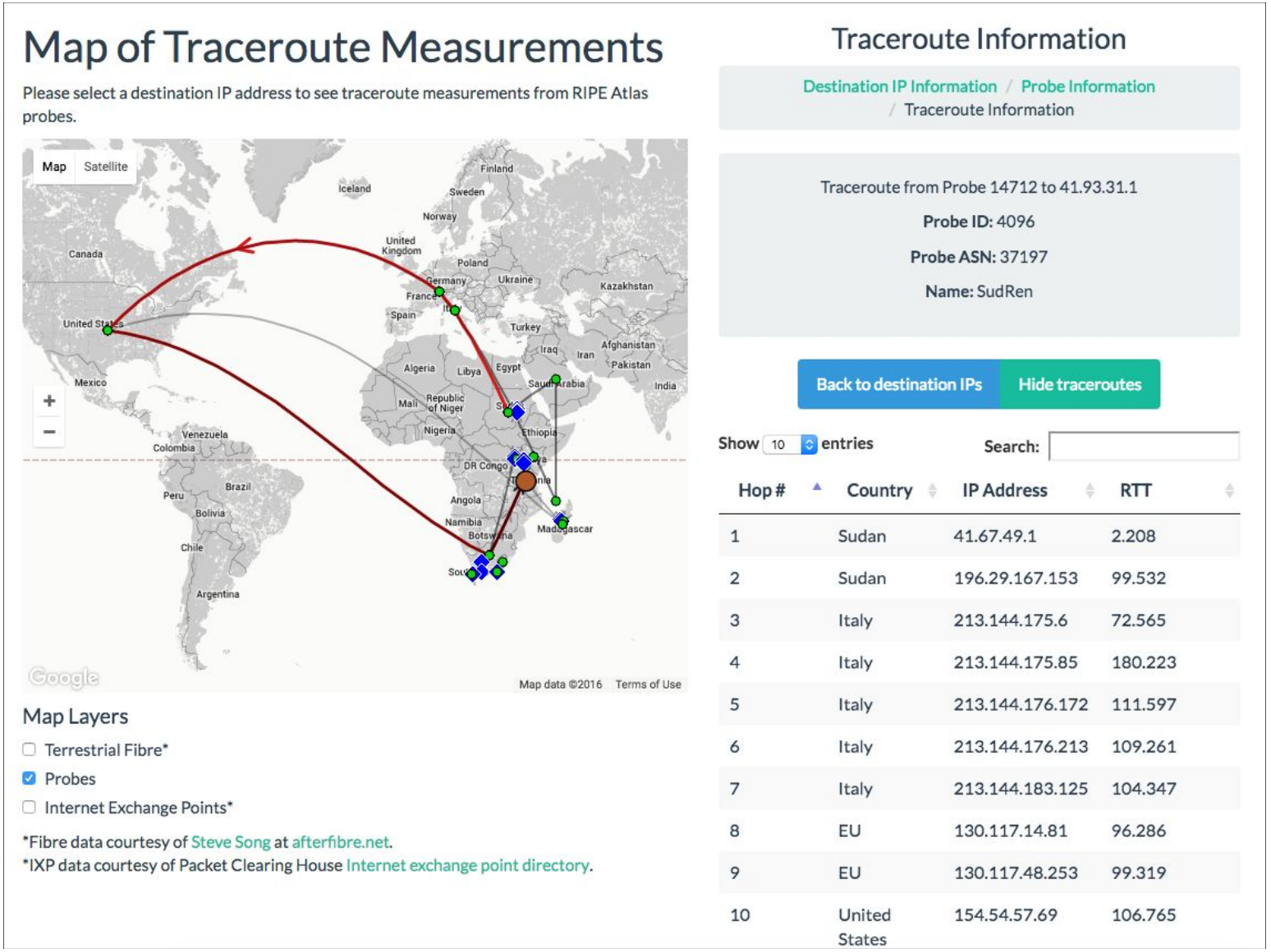
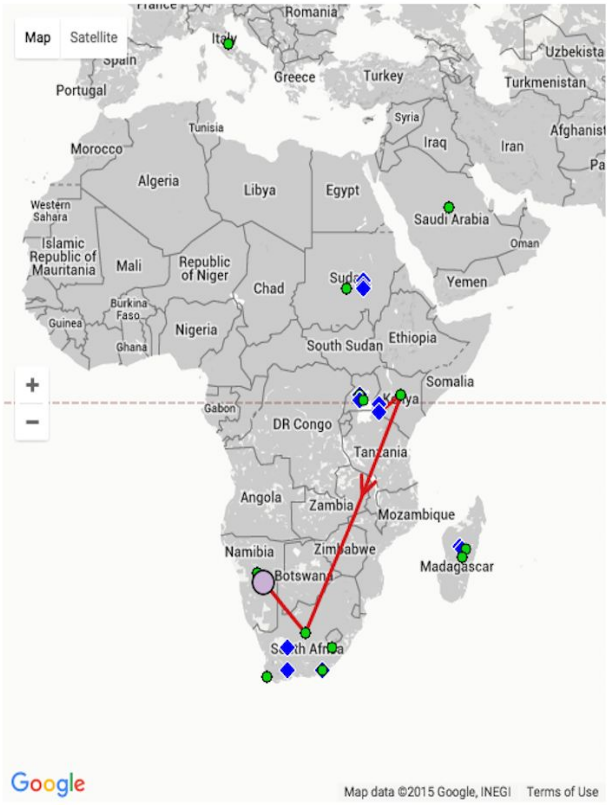


FIGURE 4.23: View multiple traceroutes to selected destination IP

Map of Traceroute Measurements

Please select a destination IP address to see traceroute measurements from RIPE Atlas probes.



Map Layers

- [ ] Terrestrial Fibre*
- [x] Probes

*Fibre data courtesy of Steve Song at afterfibre.net.

Traceroute Information

Probe ID: 13114

Probe ASN: 36914

Name: KENET

Show all traceroutes | Back to destination IPs

Show 10 entries Search:

Hop #	Country	IP Address	RTT
1	Kenya	41.89.1.1	0.5733333333333334
2	Kenya	41.204.176.121	0.5296666666666666
3	South Africa	196.32.209.85	168.706
4	South Africa	196.32.210.29	199.585
5	South Africa	196.32.210.250	199.636
6	*	*	*
7	Namibia	41.205.155.241	409.42366666666663
8	Namibia	197.188.0.5	466.38966666666664
9	*	*	*
10	*	*	*

Showing 1 to 10 of 14 entries Previous 1 2 Next

FIGURE 4.24: Animated traceroute

4.3.2 User Sampling

Two main cycles of evaluation were conducted, first with a small focus group of networking experts, and finally with a larger group of Computer Science students. For the initial evaluation, the prototype was presented to an expert user group. The expert user group consisted of domain experts in networking from the department of Computer Science: two postgraduate students, two technical staff members and a networks lecturer. For final evaluation, a usability tests were conducted with a group of 23 Computer Science students to assess the effectiveness, accuracy and usability of the visualisation platform. The participants consisted of a mixture of undergraduate university students in 2nd, 3rd and 4th year.

4.3.3 Evaluation Metrics

The evaluation experiments assessed potential use of the visualisation (Lam et al., 2011). Users' subjective feedback and opinions of the visualisation tool were taken into account and used to determine the effectiveness, accuracy and usability of the visualisation (Lam et al., 2011). Effectiveness refers to a tool's functionality and examines a user's performance when performing tasks (Freitas et al., 2002) with it. Accuracy refers to whether users can accomplish tasks and obtain the correct answers to a set of questions related to visual queries. Usability describes the quality of use of an application by a user (ease of use, satisfaction, efficiency) and is, therefore, an important aspect of the visualisation (Freitas et al., 2002; Valiati, Pimenta, and Freitas, 2006). Effectiveness and accuracy were measured using the metric of successful task completion while usability was measured using the System Usability Scale (SUS) questionnaire (Nielsen, 2001; Brooke, 1996; Koua and Kraak, 2004).

4.3.4 Usability Tests

During initial evaluation, users had differing opinions on the inclusion, on the visualization, of the undersea fibre overlay surrounding Africa. Some users thought the multiple lines and colours in the overlay added too much noise to the visualisation and needed to be removed while another thought it was useful for deducing which cables were used by traceroutes to destinations. Furthermore, it was noted that users were interested in understanding the role of physical cables, as in whether high latencies occurred due to a lack of physical infrastructure (fibre cable), or the result of routing protocols (logical topology) that needed to be changed. The expert focus group participants were also interested in identifying problems on a per link basis between traceroute hops. Others suggested that, rather than showing each IP hop, hops should be aggregated at country or city level and that the animation of the arrow of traceroute links could vary by speed based on link quality.

TABLE 4.6: Visual Queries and Related Research Theme

Visual Query	Visual Query Type	Research Theme
Task 1: Which country on the African continent has the most fibre?	Physical Network: Most Fibre Cable Network topology	Network topology; identification of physical network
Task 2: What is the route for the traceroute between institution A and institution B?	Route of traceroute: Country level	Identification of potential routes of traffic traversal; where networks connect (source, destination and intermediate hops)
Task 3: Does the route for a traceroute from institution A to institution B travel intracontinentally (within the continent)?	Route of traceroute: Intracontinental Level	Identification of potential routes of traffic traversal; where networks connect (source, destination and intermediate hops)
Task 4: Does the route for a tracereoute from institution A to institution B travel intercontinentally (to a different continent)?	Route of traceroute: Intercontinental Level	Identification of potential routes of traffic traversal; where networks connect (source, destination and intermediate hops)

The final evaluation tests were conducted in an uncontrolled environment of a computer laboratory, where participants accessed a webpage that contained the visualisation. Tasks for the usability test were designed with the objective of establishing if users could use the interactive functionality of the visualisation to answer visual queries. The tests lasted approximately 30-40 minutes.

Users performed tasks that served as a self-guided walk-through of the available functionality of the visualisation. This allowed users to familiarise themselves with the interface before they attempted to answer a set of 10 questions. The visual queries formulated were questions related to identifying networks (physical and logical), where they connect (location of source, destination, intermediate hops) and potential routes of traffic traversal (traceroute paths at a country and continental/intercontinental level) (Table 4.6). Thus, the 10 questions/tasks in the usability test were based on answering the visual queries and completing relevant tasks.

In addition to the four tasks in Table 4.6, six more tasks, involving locating various feature on the topology maps, were presented to users for the purpose of gauging successful completion of high-level subtasks of overview, zoom, filter, details on demand, relate, history and extract, as described by Shneiderman (1996). The six tasks were: locate IXP IDs; locate geolocation (country) of IP address; locate number of hops in a traceroute; locate RTT of specific IP in traceroute; locate hop number of specific IP address; and locate highest RTTs.

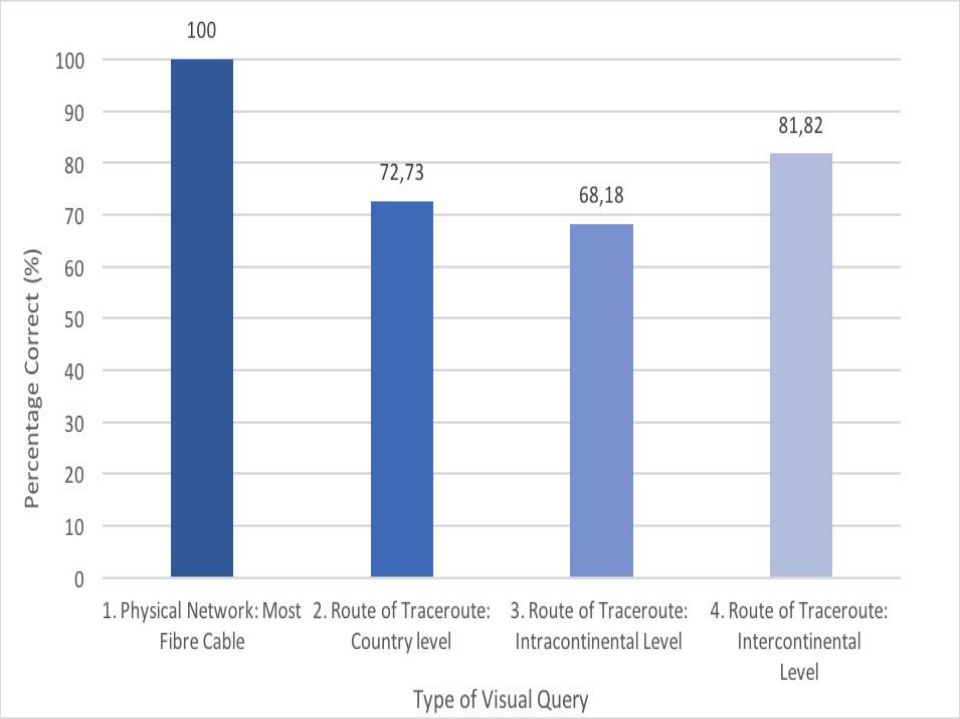


FIGURE 4.25: Percentage of users successfully completing Tasks 1-4

On completion of the tasks, users were asked to complete a system usability scale survey for the visualization tool.

4.3.5 Visualization Results

Successful task completion is characterised by the ability of a participant to obtain specific data when carrying out a task (Sauro, 2011b). If a question was answered correctly by a participant in the question set, then the task was deemed to have been successfully completed. This same definition applied to the accuracy of answering visual queries.

Figure 4.25 presents results for the four primary visual queries listed in Table 4.6. With regards to Task 1, participants were able to correctly determine the country with the most fibre cables with 100% accuracy. Identification of a traceroute's path on a country level (Task 2) had 72.73% accuracy; there was 68.18% accuracy in identifying intracontinental (within the continent) paths (Task 3); and 81.82% accuracy for identifying intercontinental (between continents) paths (Task 4).

With regards to the six locate tasks, Figure 4.26 shows the percentage successfully or partially completed tasks during the usability test.

Both the task of locating the ID of an IXP and the task of locating the country of a particular IP address had 100% successful task completion. In comparison, the rest of the locate tasks had lower successful task completion rates (Figure 4.26). This is understandable as these tasks required more complex subtasks to be performed, thus increasing the likelihood

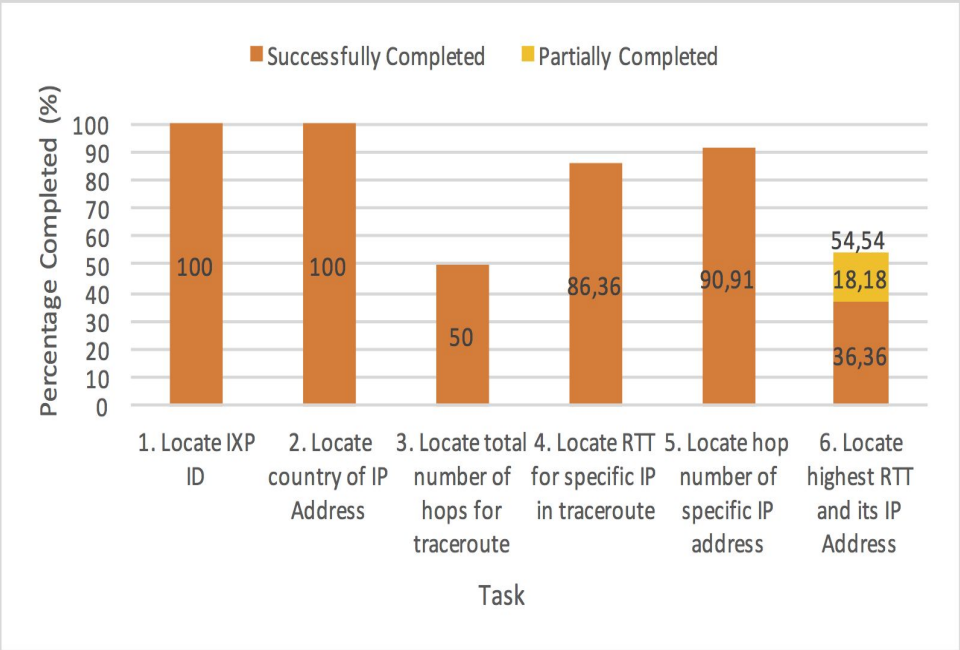


FIGURE 4.26: Correct Visual Queries Answered

of an error. It is possible that these errors may have resulted from users' lack of familiarity with traceroute.

The success rate of all 10 questions (4 visual queries and 6 tasks) across all participants was calculated to be 79.55%, where successfully completed tasks (correctly answered questions) were allocated 1 point and partially answered questions were allocated 0.5 points (Sauro, 2011b). According to Sauro (2011), who conducted an analysis of 1200 usability tasks, the average task-completion rate is 78%, which means the task completion rate for the visualisation is just above average.

Figure 4.27 presents the frequency of System Usability Scores by category for 24 users who participated in the usability tests.

The average SUS score was then calculated to be 67.8, which is about the same average score reported from an analysis of 500 studies making use of the SUS, where the average score was 68 (Sauro, 2011b).

4.4 Summary

Understanding the logical topology is an important first step in solving the latency problem. For this reason, this chapter looked at active topology measurements that were undertaken to quantify the extent of circuitous routing, as well as to determine its effects on Africa's

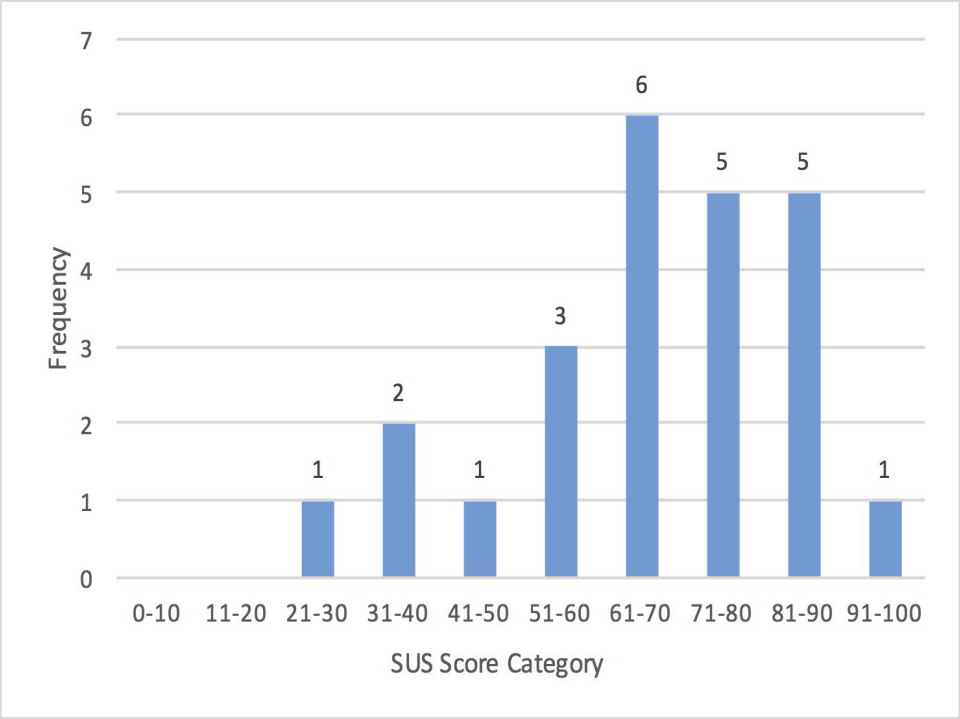


FIGURE 4.27: Frequency of individual SUS scores by category

NRENs traffic. CAIDA's Archipelago and Ripe Atlas were used to carry out active Internet topology measurements that were used to generate PoP-level and AS-level maps of the topology. Topology analysis revealed that a high percentage of traffic originating within Africa and destined for African NRENs was being exchanged through Europe. On average, 75% of the traffic originating in Africa and destined for African universities traverse links outside the continent. A logical topology map in this regard has been presented in Section 4.2.2.

This chapter also presented a probing mechanism that analysed overlapping paths to reduce redundancy during measurements. By skipping overlapping parts of the paths in traceroute probes, the number of probe packets needed to probe the topology was reduced by about 47% (Section 4.1.6). This result highlights an important mechanism for reducing probe packets in long-term topology measurement experiments, especially in multi-path networks.

In the final analysis, traffic that was routed on inter-continental links experienced latencies that averaged more than double that of traffic routed within the continent (intra-Africa traffic). On average, inter-continental traffic experienced RTTs of about 409 ms, whereas intra-Africa traffic had average RTTs of 176 ms. The details of latency performance were presented in Section 4.2.3. This result highlighted the huge performance impact of routing traffic via Europe for many NRENs. Regions with a higher inter-continental link latency, such as Southern Africa, obtained higher overall RTT for inter-continental traffic. There were, however, some African countries, such as Morocco, that are topologically very close

to exchange points in Europe and, for these, inter-continental routing did not significantly increase latencies (Section 4.2.4).

Lastly, this chapter presented the design of a geo-spatial visualisation. Evaluation of the tool confirmed its effectiveness in communicating the logical topology. Participants were able to correctly determine the country-level identification of a traceroute's path with a 72.73% accuracy, and 68.18% accuracy for identifying intra-continental from the intercontinental traffic. Details of these results are presented in Section 4.3.

Chapter 5

SDN/LISP Based Traffic Engineering for UbuntuNet Alliance

The previous chapter (Chapter 5) highlighted the presence of high latencies between African NRENs. The chapter also showed that despite high path redundancy and multipath between NRENs in Africa, a high percentage of inter-NREN traffic across Africa traverses inter-continental routes through Europe. Dealing with the latency problem between Africa's research and education institutions requires designing cross-border traffic engineering mechanisms that can identify and use optimal low latency intra-continental paths. This chapter discusses the potential for reducing end-to-end latencies in the UbuntuNet Alliance by employing, at the NRENs' gateways, a traffic engineering mechanism that is based on gateway ranking. Using a Software Defined Network (SDN) topology, and a LISP mapping system, the chapter examines the potential for dynamically ranking egress and ingress paths between multi-homed NRENs, based on end-to-end path metrics.

5.1 Introduction

Recently, the UbuntuNet Alliance embarked on the AfricaConnect project to increase the intra-Africa interconnectivity of Africa's NRENs. As a result of the project, the UbuntuNet topology now has at least eight Points of Presence (PoPs) interconnected with intra-Africa terrestrial fibre optic cable. The establishment of such multiple PoPs as well as multiple intra-Africa and transcontinental links in the UbuntuNet topology provides new opportunities for improving performance of traffic exchange among Africa's NRENs. However, the inability of traditional protocols to fully take advantage of the available path diversity remains a challenge. For this reason, African NRENs need to, apart from implementing the physical interconnectivity, consider appropriate traffic engineering mechanisms to allow NRENs to discover and use optimal inter-NREN paths.

Given the opportunities for traffic engineering provided by the multiple intra-continental and transcontinental links provided by the AfricaConnect network, one way of improving

the performance and optimization of traffic exchange across African NRENs is to enable dynamic selection of optimal traffic exchange routes based on application QoS needs. In this regard, NRENs could implement mechanisms to enable them to announce and exchange traffic through multiple Internet attachment points. For example, path selection for delay sensitive applications could be made based on prevailing end-to-end latencies through multiple intercontinental and intra-Africa links.

As discussed in Section 2.4, SDN provides NRENs with new opportunities for traffic engineering to improve network management through the ability to centrally manage heterogeneous devices from different vendors, using a standardized Application Programming Interface (API). The separation of the control plane from the data plane allows for flexible traffic engineering mechanisms through fine grained control of flow-based packet forwarding at various levels of granularity, including session, user, device, and application. SDN also eliminates the need for manual reconfiguration of devices when there are changes in policy or network structure. Similarly, LISP, introduced in Section 3.2.3, provides new opportunities for NRENs to announce multiple gateways so that alternate routes more visible and accessible (Secci, Liu, and Jabbari, 2013; Saucez et al., 2012). NRENs can discover each other's multiple gateways, which enhances path diversity and allows the networks to exchange traffic through multiple gateways.

To achieve optimal end-to-end performance using LISP, there is need for the source and destination networks to be able to evaluate alternate locators and to dynamically select the source network's ingress locator and destination network's egress locator. This would also ensure that packet forwarding responds to networks' dynamic conditions, such as changing traffic volumes and link failures. However, relying on pre-configured locator preferences may not result in the best end-to-end performance. This is because, in its standard form, LISP does not adjust locator priorities and weights in response to prevailing network conditions. LISP does, however, define a probing mechanism to allow locators to probe each other, thus being able to detect unreachable locators. From a traffic engineering point of view, enabling locator probing would also provide latency estimates between a pair of locators, which can be useful in identifying low delay gateways.

The overarching purpose of this study was to evaluate the extent to which LISP and SDN can support dynamic selection of end-to-end paths between multi-homed edge networks. LISP and SDN were used in a complementary fashion in this study. On one hand, LISP enables source networks to proactively select the destination's egress gateway, and allows the destination networks to influence the selection of incoming paths by ranking their multiple egress gateways. On the other hand, SDN, through the use a central controller, is used for configuration and management of paths between the LISP selected source and destination gateways. The SDN's centralized mechanism allows for global and dynamic

analysis and regulation of the forwarding behaviour of path hops. Experiments were conducted to evaluate the utility of using active measurements to aid dynamic adjustment of LISP locators' priorities at the network edge. Each of the edge gateways performed end-to-end active measurements so as to consider end-to-end latency, jitter and packet loss in determination of gateway ranks. In the subsequent experiments presented in the next chapter (Chapter 6), passive measurement metrics, such as packet count and link utilization were also used in path ranking. Metrics obtained through locator probing, as well as network statistics/events data available in LISP routers were used to adjust locator priorities. The objective was therefore to assess network performance in a topology that employs a dynamic performance-based path selection, versus static default path routing. A note should be made that while SDN is generally implemented in intra-domain contexts, the federated nature of NRENs makes it possible for a centralized controller to apply across the NRENs.

The chapter makes the following contributions:

- Framework for performance-based gateway ranking in LISP and integrating LISP-based traffic engineering in SDN topology.
- LISP/SDN traffic engineering architecture aimed at minimising latency between LISP-based NRENs.
- An evaluation of LISP traffic engineering capabilities for UbuntuNet NRENs.

5.2 A Model for Performance-based Path Selection

The proposed traffic engineering framework presented in this chapter incorporates LISP and SDN as depicted in Figure 5.2. The framework imagines a scenario in which traffic source gateways have the ability to select the destination's ingress gateway based on metrics of the edge-to-edge path, as well as for a controller to set up end-to-end paths between selected source and destination gateway. This requires that the source gateway has a mechanism for learning the destination's multiple gateways that, for LISP-based networks, can be inferred from the EID-RLOC mappings that can be obtained from the mapping server.

The availability of multiple locators for the same destination increases path diversity as the source networks are able to forward traffic for a particular destination through multiple remote locators (gateways). Multihomed edge networks are therefore able to achieve some degree of path diversity, and as a result, additional end-to-end performance gains can be achieved with the ability to dynamically select the ingress link at the destination network. Achieving optimal end-to-end performance in such environments requires that the source and destination networks should be able to evaluate the alternate paths, and to dynamically select both the source network's egress link and the destination network's ingress link. In

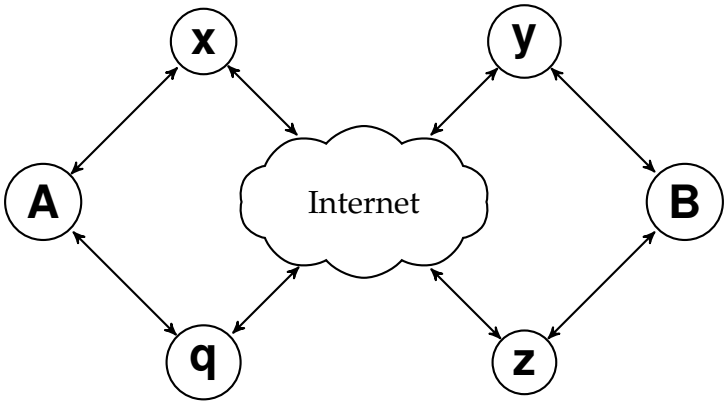


FIGURE 5.1: Multihomed Networks A and B, multihomed through providers (x,q) and (y,z) respectively

particular, the source network needs a way of discovering and evaluating end-to-end links through alternate egress and ingress links.

In Figure 5.1, for example, edge networks A and B are multi-homed to networks x,q and y,z respectively. Depending on how the routing is done in the Internet core, the choice of the egress link by network A, i.e. (A,x or A,q), has the potential to influence selection of ingress link towards B i.e. (y,B or z,B). Since each end-to-end path has its own unique path metrics in terms of bandwidth, delay, and loss, selection of particular egress and ingress links at A and B impacts the overall quality of the end-to-end path.

5.2.1 Performance-based Locator Selection

For multihomed networks, end-to-end performance gains can be achieved with the ability to dynamically select the ingress gateway at the source network, and the egress link at the destination network. Although the interdomain paths between LISP locators are determined by the BGP protocol, the multiple potential paths still exist between any two LISP-enabled networks if one of them is multihomed. It is up to the multihomed networks to determine how to leverage the multiple paths for managing traffic flow between them. For example, multihomed edge networks may attempt to influence incoming traffic by dynamically changing the EID/Locator mappings and locator priorities. For outgoing traffic, edge networks may leverage the multipath to improve performance by directing traffic based on performance of the multiple locators. A detailed description of LISP is given in Section 2.3 and a review of LISP's traffic engineering implementations is given in Section 3.2.3

To achieve performance based gateway selection, the LISP gateway routers in this framework have two additional key components: a measurements-based RLOC ranking module; and an SDN module (Circuit Pusher) for setting up end-to-end paths through an SDN controller. Figure 5.2 indicates the components of the locator ranking and selection model.

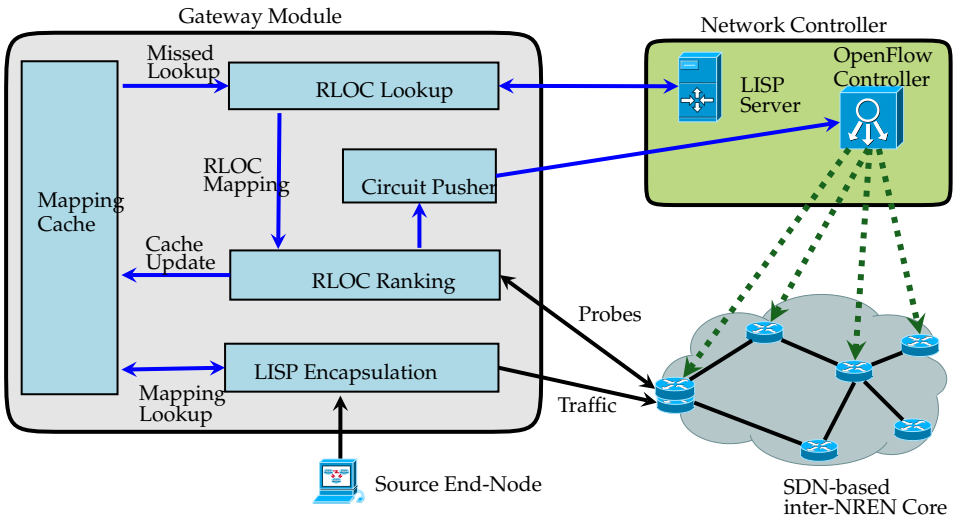


FIGURE 5.2: LISP/SDN Multihome Traffic Engineering Model

In summary, the model works in the following manner: For each new traffic flow from a local source node to some remote network, the source gateway queries a local cache to obtain the remote locator associated with the destination EID address. If no mapping exists in the mapping cache for a destination network, the *Mapping* module is invoked to perform the lookup, querying the LISP mapping system through an RLOC lookup API to obtain the destination network's gateway locators. The mapping records then stored in a local mapping cache. A *Locator Ranking* module in each gateway is responsible for periodically performing active measurements towards remote locators that are listed in the local mapping cache. The *Locator Ranking* module uses the network metrics obtained from the active measurements to rank the local and remote gateways, after which it updates the local mapping cache. The source network selects the source and destination gateways for outgoing packets based on the rankings in the mapping cache. The *Circuit Pusher* module is invoked to configure, through an OpenFlow SDN controller, the fastest path between every pair of source and destination locators.

5.2.2 Implementation

To evaluate the performance based LISP gateway selection, an SDN emulated topology, consisting of an OpenFlow network controller, SDN switches, Linux hosts, and a LISP mapping server, was constructed. The topology was built in a virtual environment, using the Mininet network emulator (Heller et al., 2012). Mininet is a lightweight process-based virtualisation and network namespaces emulation system. Process based virtualisation makes possible the emulation of networks comprising hundreds of nodes and dozens of switches, on a single computer. Mininet integrates with SDN (OpenFlow) network controllers, Openflow switches, Linux hosts and network links. The software switches implemented in Mininet

provide the same semantics as the hardware-based OpenFlow switches, making it possible for controllers and applications developed and tested in the emulated environment to be deployable in real world OpenFlow-enabled networks without modification. Furthermore, Mininet's hosts run a standard Linux kernel and network stack, and can therefore be used to test real network applications.

LISP Mapping Server

The emulated topology was built to use the LISP mapping system and LISP gateway routers. The LISP topology was based on an implementation of LISP called lispers.net[1], a closed source implementation that is written in the Python programming language. lispers.net provides a flexible API that simplifies the process of registration of EID-RLOC mapping records, and allows dynamic runtime re-configuration of the local mapping cache. This flexibility was very useful for achieving dynamic ranking of RLOCs depending on run time performance measurements. However, some modifications had to be made to make it work on Mininet virtual hosts. This author worked with the lispers.net developer to modify the software to enable it to work with any type of interface device to be used as the RLOC interface. This was an important modification, as it allows new operating systems, with different interface naming conventions, such as virtual systems, to be supported. This made usage of lispers.net possible for researchers who use Mininet for topology emulation and experimentations.

Central to a LISP-based topology is a Mapping Server, a database system responsible for receiving registrations of EID-RLOC mappings from LISP sites. The mapping server was configured to accept registrations from all LISP sites with valid authentication.

Gateway Locators

In this emulated topology, each LISP site was connected to two ISPs (Figure 5.3). Each LISP gateway router was therefore configured with two external network interfaces, and thus two locators, as well as one internal EID interface. The EID interface is used as the end hosts' default gateway for out going traffic.

Each gateway's configuration includes the IP address of the LISP mapping server, as well as the required authentication details. The mapping registration process is conducted from the LISP gateway router by announcing to the mapping server, their networks' EID prefixes and respective locator IP addresses. An EID prefix may be associated with multiple locators, in which case the EID network is said to be multihomed. In this topology for example, each site's EID prefix was mapped to both the site's locators such that end hosts could be reached by remote hosts through either of the locators.

[1] www.lispers.net

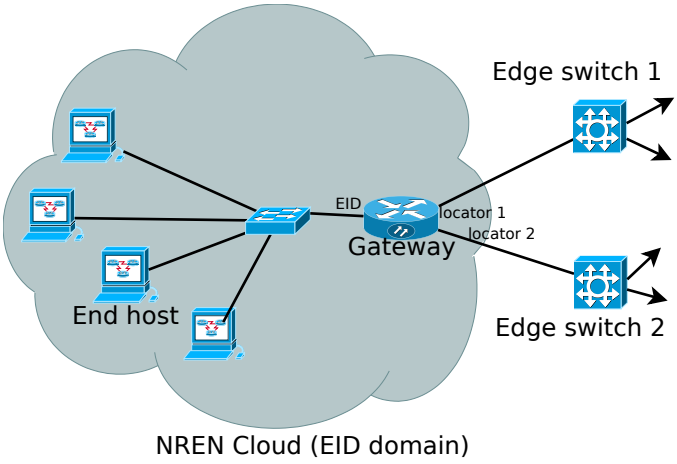


FIGURE 5.3: Multihomed LISP gateway

During topology initialization, each LISP gateway registers a database mapping for each of its locator interfaces. The mapping includes the network's EID prefix and subnet mask, the locator IP address, the locator's priority value, and the weight value. In LISP, selection of the remote network's egress locator (ETR) is ordinarily determined by priority and weight values recorded in locator records retrieved from the mapping system and stored in a local cache. If multiple locators for the same destination exist, the priority values, ranging from 0 to 255, are used to select the destination host's locator. Lower values for priority indicate that the locator should be preferred by the source LISP gateway when sending traffic to a remote network. If the priority values are the same for multiple RLOCs to the same EID prefix, then the weight value is used to determine the ratios for balancing unicast traffic between the RLOCs. In this experimental setting, both locators were assigned the priority value of 1, although this was inconsequential given that the actual ranking/priority for each of the locators was being computed dynamically by the source gateway.

Locator Ranking and Mapping Cache Updates

During network runtime, as end hosts start flows to remote hosts, the LISP gateway queries the mapping server to obtain locator addresses for destination EIDs. If the destination EID is registered in the mapping server, the result of the query is set of mappings, with each entry being a tuple consisting of at least the EID prefix, locator IP address, locator priority and weight. The gateway maintains a separate mapping cache for each of its locator interfaces, and every cache entry has a configurable expiry period (120 seconds was used in experiment). For each new traffic flow, the source gateway checks the local mapping cache to determine if a remote EID's locator address is already available and, if multiple exist, to select the highest ranked locator. If the mapping is not available in the local cache, the gateway queries the LISP mapping system to obtain the destination network's locator IP

addresses. While the local mapping cache remains valid, all subsequent flows to the same EID prefix use the cached locator mapping, subject to the locator priorities that are dynamically adjusted by the gateway. After the expiry of the mapping cache, new flows trigger the gateway to query mapping server again.

In the proposed framework, the locator ranking module was used to evaluate end-to-end performance towards the destination locators. The RLOC ranking module is incorporated with a network measurement mechanism that is installed in the gateways. The measurement module generates probe packets to measure the locator-to-locator latency, jitter and packet loss, which are used for updating the locator priority values stored in the cache. The probes are sent from both the locator interfaces (Figure 5.3), considering that the paths from the two local locators to either of the destination's locator could be different and exhibit different end-to-end latencies. Every 30 seconds, the LISP gateway sends out a ping probe through each of its locator interfaces to all the remote locators that are listed in their respective mapping caches. To convert measured path delays into locator priority values (ranking), the latencies are scaled into a ratio of a configurable maximum expected path delay in the topology (1000 ms was used in this case). For example, a path delay of 100 ms is scaled to the value 0.1, whereas path delays of equal to or greater than 1000 ms are scaled to 1.

The routing cost for a locator-to-locator path can thus be modelled as a vector comprising the measured performance metrics and the network RLOC preferences (Secci, Liu, and Jabbari, 2013). Let $P(A_{xy}), P(B_{yx})$ be the performance cost vectors for two edge networks A and B, with respect to forwarding traffic through their access links x and y respectively. The performance cost P from each edge network comprises a set of end-to-end path metrics:

$$P(A_{xy}) = \sum_{i=1}^{n} \Lambda_i \cdot K_{xy_i}$$

where K_{xy} = *(latency, jitter, packet loss)* is a set of path metrics from locator x to locator y. A weighting variable Λ is used to aggregate the path metrics in K into a cost value $P = \sum_{i=1}^{n} \Lambda_i K_i$; where n is the number of metrics used, $\sum_{i=1}^{n} \Lambda_i = 1$. The Λ_i associated with each metric i depends on the optimisation objective and, in these experiments, Λ values used were 0.6, 0.2, and 0.2 for latency, jitter and packet loss, respectively.

OpenFlow Controller Path Enforcement

The Ryu controller is used for computing the fastest paths between every pair of nodes in the topology and configuring necessary packet forwarding rules. The computation for shortest path is done using the Dijkstra's algorithm. To be able to compute the shortest paths between nodes, the controller maintains a global view of the SDN topology, including availability of of switches and links. This global knowledge is obtained by the controller through the Link

Layer Discovery Protocol (LLDP), a layer-2 protocol used by network devices to advertise their identity and capabilities. In Ryu, this feature is activated by the *-observe-links* when launching the controller, and triggers network discovery callbacks with the SDN switches. These call backs include *EventLinkAdd, EventLinkDelete, EventSwitchEnter, EventSwitchLeave, EventPortAdd and EventPortDelete*, and the controller uses the callback messages exchanged with switches to also monitor network changes and failures.

The controller uses the Dijkstra algorithm to compute shortest paths between every pair of edge networks (NREN). Every NREN in the topology has two edge switches, which are also connected to each of the NRENs LISP gateways. Each edge switch is further connected to an ISP switch. Every switch in the topology is uniquely identified by an integer value. As the topology is initialized, the controller computes end-to-end paths using the Dijkstra algorithm and stores the paths in a table, with each path being uniquely identified by the source and destination switches. Furthermore, each NREN manually registers their gateway IP addresses (locators) to the SDN controller through the REST API, mapping the locator IP address to the specific edge switch. For example, the request "*'http://controller-ip-address:8080/mapLocators/1/1/10.0.0.1/255.255.255.255'* tells the controller that the locator IP address "*10.0.0.1*" is connected to port 1 on switch 1.

In this proposed SDN/LISP framework, each edge network gateway has a Circuit Pusher: an SDN module for setting up end-to-end circuits between locators. After the source gateway has selected both the local and remote locators, it invokes the SDN module to configure, through an SDN network controller, a unidirectional end-to-end circuit between locators. This is achieved by installing flow entries on all switches that are part of the fastest path between two selected locators. The installed path is unidirectional because each source gateway independently performs RLOC lookups and path ranking, and sets up a circuit toward the remote RLOC.

The circuit pusher uses a REST API to request path configuration between a specific local locator (source) and the destination locator. The OpenFlow controller checks the paths table and locators table to select a single end-to-end path for the specific tuple (source switch, input port, destination switch, and output port), where input and output ports are, respectively, the attachment ports for the source and destination locators. The controller then configures the end-to-end path by installing the appropriate IP-based forwarding rules in each of the path's switches.

5.3 Experimental Evaluation

Experiments were conducted to evaluate the extent to which LISP locator ranking could reduce end-to-end latencies in the network. The aim was to compare the performance of two traffic engineering mechanisms: one in which source networks periodically rank the

destinations' multiple gateways based on end-to-end latencies; and the other in which static paths are pre-configured based on the destinations' locator priorities. The static path is selected without regard to prevailing network conditions. In the static set up, the source networks' LISP gateways would obtain a destination's locators from the mapping system and forward traffic to the highest ranked locator. Inside the SDN topology, a single fastest path is configured between every source and destination locator.

The experiments were set up using three server machines that were provided by the University of Cape Town's High Performance Computing (HPC) facility[2]. All the three machines run the server version of Ubuntu 14.04.5 LTS operating system. The first machine, with 32 gigabytes of memory and eight CPU cores (Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz), was used as the Mininet server responsible for creating and running the SDN topologies. The other two machines, each with 8 gigabytes of memory and four CPU cores (Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz), were respectively used as SDN controller and as LISP mapping server.

5.3.1 Topology

For this experiment, the network was designed as a Fat Tree topology (Al-Fares, Loukissas, and Vahdat, 2008), with each edge switch representing an NREN (Figure 5.4). To implement LISP routing, the network model was incorporated with a LISP mapping server and each NREN had two LISP gateways connected to two different ISPs. The ISPs were modelled as SDN switches, which were further interconnected through two core SDN switches. In total, the model consisted of 13 edge networks representing the UbuntuNet Alliance NRENs.

More specifically, the network was designed with the following features:

- There are 13 edge networks representing 13 NRENs in the UbuntuNet Alliance. Each edge network was attached to between 5 and 10 hosts representing university networks. In total, there were 100 end hosts acting as sources and destinations of traffic in the topology.
- To simulate multi-homing, each edge network was connected to two cable operators. The first connection models an intra-Africa link and has generally lower latency (20 ms to 100 ms), while the second connection represents an intercontinental link that has higher latency (100 ms to 400 ms). This setup models latencies observed for traffic exchanged between African NRENs presented in Section 4.2.3.
- Each access link between edge networks and the cable operators is configured with bandwidth value randomly selected between 2 Mbps and 4 Mbps.

[2]http://hex.uct.ac.za/

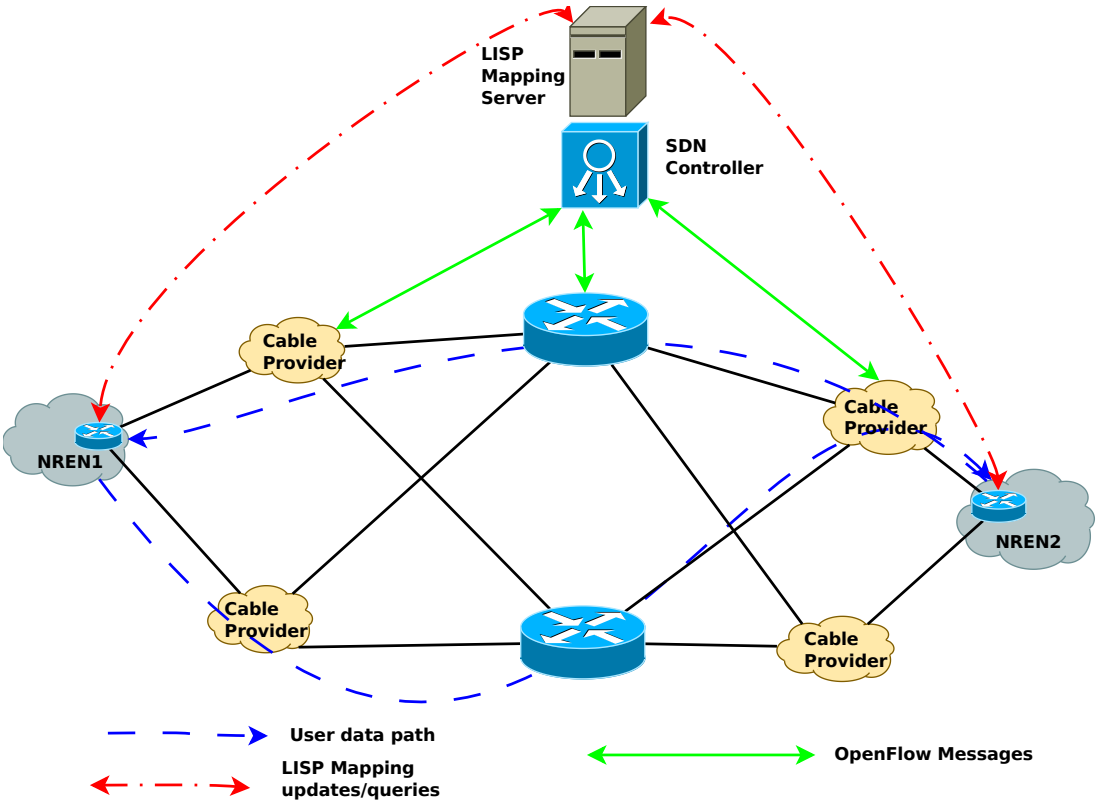


FIGURE 5.4: LISP/SDN multipath emulation

- Each cable operator is further connected upstream to the topology by connecting to 2 core switches that represent a global IXP fabric. The links between cable operators and the IXP are equally provisioned with bandwidth of 100 Mbps.

Figure 5.4 is a schematic view of the topology implementation showing just two NRENs. The actual emulation was made up of 13 NRENs.

5.3.2 Test Traffic

To evaluate the traffic engineering framework, one needs a network traffic model that realistically emulates the characteristics of the traffic in the actual the topology. This entails using traffic that characteristically resembles the pattern in the emulated networks. This is particularly important in this work considering the fact that a major scalability issue with centralized network architectures, such as the OpenFlow controllers systems, hinges on the ability of the controller and forwarding devices to cope with the traffic characteristics in the network (Benson, Akella, and Maltz, 2010).

Researchers have characterized Internet traffic based on flow metrics such as byte volume, packet volume, flow duration, and flow inter-arrival time (Quan and Heidemann, 2010; Zhang et al., 2009). The length of data flows impacts the relative latency introduced at the controller and, furthermore, the number of active flows has implications for the size of

the forwarding tables maintained at each OpenFlow forwarding device. Flows within local area networks are predominantly longer (Quan and Heidemann, 2010). For example, Quan and Heidemann (2010) characterised traffic inside a campus network and established that 21.4% of the traffic was carried by flows longer than 10 minutes, 12.6% by flows longer than 20 minutes, and nearly 2% was carried by flows longer than 100 minutes.

On the other hand, Internet wide IP traffic is characterised by a large percentage of short flows. Brownlee (2005) showed that at least 45% of Internet streams were short flows lasting less than 2 seconds, and that 98% of all the streams lasted no more than 15 minutes. Short flows are bursty and have flow speeds ranging from 1 Bps to over 10 kBps, while longer flows are slower, averaging around 50 Bps (Quan and Heidemann, 2010).

In terms of protocols, Internet traffic is largely dominated by TCP and UDP traffic. Over the years, Internet traffic has been dominated by TCP in terms of number of packets and bytes. However, in recent years, the percentage of UDP traffic has increased with the advent of UDP-based P2P applications and streaming multimedia that transport large volumes of data(Zhang et al., 2009). UDP is now responsible for the highest percentage of flows on the Internet, as shown by a 2013 Internet traffic analysis done by the CAIDA [3], which showed that the ratio of UDP to TCP traffic was almost 0.21 in terms of packet numbers, 0.11 in terms of byte count, and 3.09 in terms of flows (McCreary and Claffy, 2000).

Notwithstanding the characteristics of the traffic in the topology, another important characteristic of the network environment is the level of bandwidth utilization. Utilization refers to the ratio of the traffic flowing through the network to the maximum bandwidth available in the network (Welzl, 2005). Congestion occurs when a network is carrying more data than it can efficiently handle, which results in reduced quality of service through factors such as increased queueing delays, packet loss and jitter (Welzl, 2005).

To ensure that performance measurements were conducted when the network was in a realistic state, two sets of traffic were used in the experiments: background traffic; and performance measurement traffic. Background traffic was generated for the purpose of creating a realistic network environment, and such traffic was continuously exchanged between end hosts in the topology. Background traffic was based on Internet wide characteristics (McCreary and Claffy, 2000; Brownlee, 2005) as follows:

1. **Protocol flow:** UDP to TCP ratio: 3:1 (McCreary and Claffy, 2000)
2. **Flow Duration:**
    - 0 - 2 sec : 45% of all the traffic
    - 2 sec - 15 mins : 55% of all the traffic.
3. **Flow rate:**

[3] https://www.caida.org/research/traffic-analysis/tcpudpratio/

- Short flows (0 - 2 sec, 45% of all the traffic) : 1 Bps - 10 kBps. Average flow rate: 5 kBps
- Medium flows (2 sec - 15 mins 55% of the traffic) : 50 Bps The total average flow rate was about 2.28 kBps (0.45*5 + 0.55*0.05) kBps

In the emulated topology, the total access bandwidth was about 78 Mbps, i.e each of the 13 NRENs had an average bandwidth of 3 Mbps on the access links connecting to the core topology. Given an average flow rate of 2.28 kBps (about 18 kbps) and a total access bandwidth of 78 Mbps (78000 kbps), a total of about 4,333 flows would entail 100% utilization of the access link's bandwidth. However, tests on the network indicated congestion and degraded performance at about 3,000 total flows, i.e at about 70% of the bandwidth capacity. The level of utilization was reached with each of the 100 end hosts in the network maintaining about 30 outbound flows at all times.

In the experiments, three levels of network utilization were used by appropriately setting the number the traffic flows maintained by each end host:

- **Low** utilization at 30% (1,300 total flows, or 13 flows per host);
- **Medium** utilization at 50% (2,100 total flows, or 21 flows per host); and
- **High** utilization at 70% (3,000 total flows, or 30 per host)

Some of the most widely used traffic generators include Iperf, PackETH, D-ITG, and Ostinato (Botta, Dainotti, and Pescapé, 2012; Kolahi et al., 2011). PackETH (Jemec, 2012) is a stateless packet generator designed for Ethernet networks, and supports a number of protocols including UDP, TCP and ICMP. Iperf(Tirumala et al., 2005) is mostly used for evaluating topology parameters such as bandwidth, delay, window size and packet loss, for both TCP and UDP traffic. Ostinato(Botta, Dainotti, and Pescapé, 2012) is a user level traffic generator tool that supports UDP and TCP protocols at multiple rates.

In the emulated topology, D-ITG (Distributed Internet Traffic Generator) (Avallone et al., 2004) was used for generation of background traffic. D-ITG was selected due to its fine-grained controls and ability to generate Internet traffic based on user defined packet parameters.

On the other hand, measurement traffic was aimed at characterizing end-to-end performance in terms of round-trip times and jitter, and was generated from and to designated end hosts in the network. Performance measurement traffic was generated using Iperf(Tirumala et al., 2005). To perform the measurements, a single designated measurement host in each and every network would randomly select another measurement host in a remote network and initiate a TCP Iperf transmission for a random length of time ranging from 1 sec to 300

seconds. After completion of a measurement flow, the end host would again randomly select another remote host and run the measurement again. All the measurement hosts looped through this process for at least 30 mins.

The experiment was conducted once for the each of the three levels of network utilization: Low; Medium, High.

Independent Variables:

The variables for the experiment include:

- Volume of traffic was adjusted to evaluate the system's response to different levels of network load.
- Duration of flows varied randomly between 1 sec and 300 sec.

Dependent Variables:

The experiment measured two aspects of network performance: latency (round-trip times) and jitter.

5.4 Results

A key objective of the RLOC ranking was to discover and direct traffic flows through lower latency paths towards multi-homed remote networks. The experimental evaluation was thus aimed at measuring the performance of TCP traffic when RLOC ranking and dynamic path configuration is employed. The evaluation focused on TCP traffic, considering that it is particularly impacted by network round-trip-times. The evaluation also considered how jitter is affected due to path ranking and circuit configuration.

5.4.1 Round Trip Times

Figure 5.5 and Figure 5.6 show the dispersion and mean of the RTTs for TCP traffic in a LISP/SDN topology, with each flow lasting between 1 sec and 300 sec.

The vertical lines represent the dispersion of flow RTTs over the time interval. The average RTT for all the flows, grouped by flow duration, is indicated by the thick line along the vertical lines. Figure 5.5 shows that RLOC ranking achieved overall average RTT of 0.50 Sec, in contrast to an overall average RTT of 0.63 Sec for non-ranked default gateway selection approach (Figure 5.6). In this case, the RLOC ranking mechanism achieved 20 % lower overall latency compared to the default gateway selection mechanism.

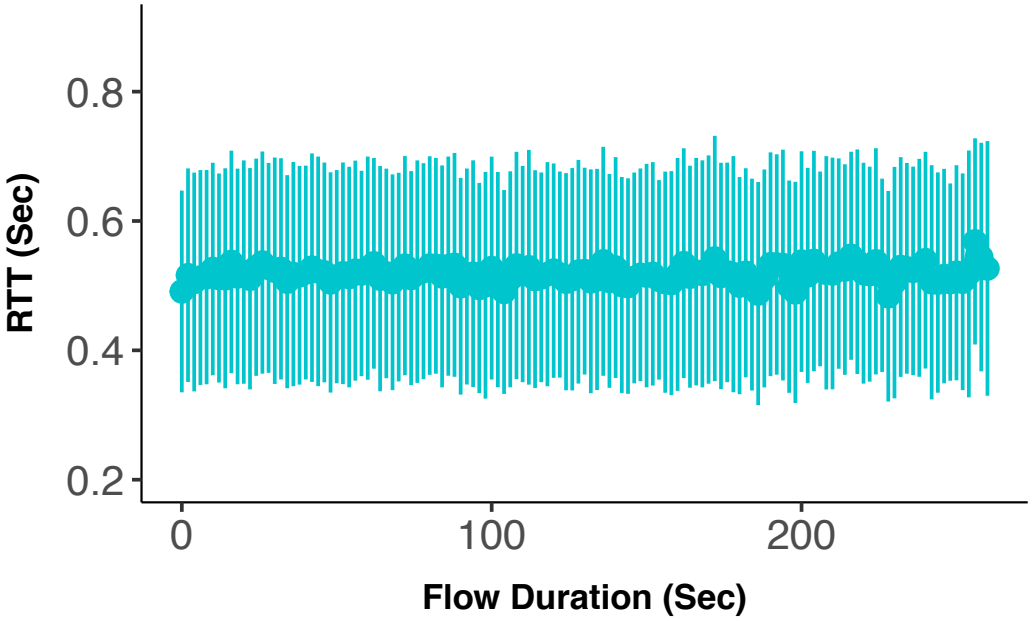


FIGURE 5.5: Dispersion and Mean RTTs for traffic flows in a network operating WITH LISP gateway ranking.

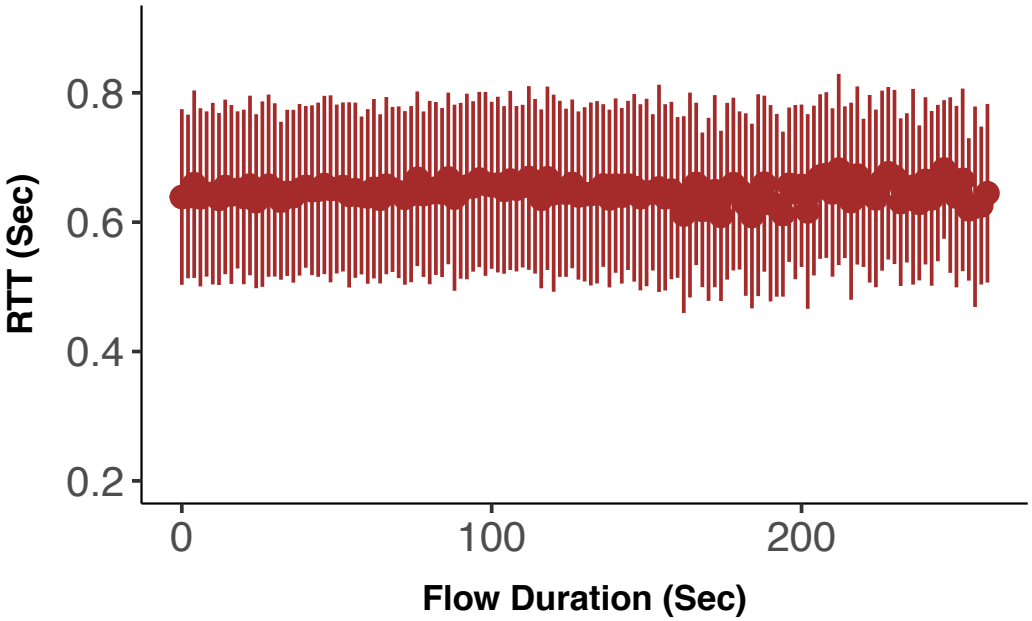


FIGURE 5.6: Dispersion Range and Mean RTTs for traffic flows in a network operating WITHOUT LISP gateway ranking.

The performance gains from RLOC ranking appear to diminish significantly with increased network load. To observe this, the experiments were conducted with low, medium and high network traffic, as described in Section 5.3.2. As the network gets more congested, the observable gain from RLOC ranking is significantly reduced. During *low* traffic experiments, the RLOC ranking had about 40 % performance advantage over the non-ranking mechanism in terms of RTTs (i.e 0.41 Sec vs 0.68 Sec RTT, Table 5.1). This advantage was reduced to 27 % and 10 % during medium (Table 5.2) and high network load (Table 5.3), respectively. Figures 5.7, 5.8, and 5.9 depict RTT differences for ranking and non-ranking approaches, measured during the different levels of network load.

Figure 5.10 and Figure 5.11 shows the dispersion and mean of the RTT when the network is nearly congested, i.e under **High** network utilization. In this condition, the same range of RTTs are obtained by traffic flows for both Ranking and Non-Ranking mechanisms. In general, centralized systems are vulnerable to performance bottlenecks under system overload. For instance, inter-arrival times of traffic flows have implications on the performance of centralized network controllers (Benson, Akella, and Maltz, 2010). In an OpenFlow network architecture, for example, a scalability challenge stems from the fact that the first packet of each flow is forwarded to a central controller, which is responsible for determining and configuring the forwarding path for the flow. Similarly, for LISP, the egress gateway performs a lookup from a mapping server to determine each new flow's destination network's RLOC. In either case, the flow inter-arrival time has a scalability impact on the network, as higher rates for new flows result in bottlenecks at the SDN and LISP controllers, thereby introducing latency and jitter. Although the scalability and performance bottleneck would affect both the ranking and normal LISP operations, the RLOC ranking mechanism would experience more severe impact as it is dependent on receiving replies from probe packets, which take longer when there is congestion. One way of dealing with RLOC ranking during congestion is to reduce the amount of probing required by using historical performance information to select the RLOCs. Also, the amount of probing needs to be reduced by performing ranking only for critical flows (eg. delay intolerant applications).

The RTT results show that the RLOC ranking mechanism barely produces performance gain under network congestion. The diminished performance gains of RLOC ranking can be explained with the fact that ranking entails that paths (gateways) with lower latency tend to initially carry most of the traffic between the networks, while the higher latency paths remain mostly idle. However, as the network links get more congested, the otherwise shorter links begin to exhibit equally higher latencies. This results in lowering of the rankings for previously high ranked paths' RLOCs. This lead to higher probability for the traffic to be forwarded via high latency paths. Furthermore, during congestion, there is higher chance of the probe packets timing out, such that the probe engine fails to discover any lower latency RLOCs and resorts to using default paths. This results in the source gateway forwarding the

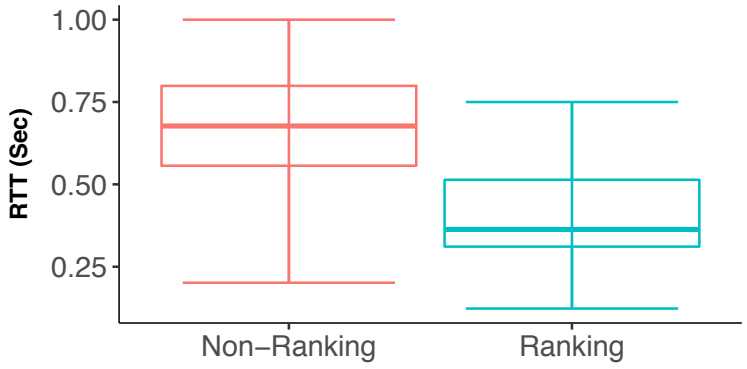


FIGURE 5.7: Ranking vs Non-Ranking in **LOW** network load

RTT(Sec)	Non-Ranking	Ranking
1st Quantile	0.56	0.31
Mean	**0.68**	**0.41**
3rd Quantile	0.79	0.51

TABLE 5.1: Ranking vs Non-Ranking in **LOW** network load

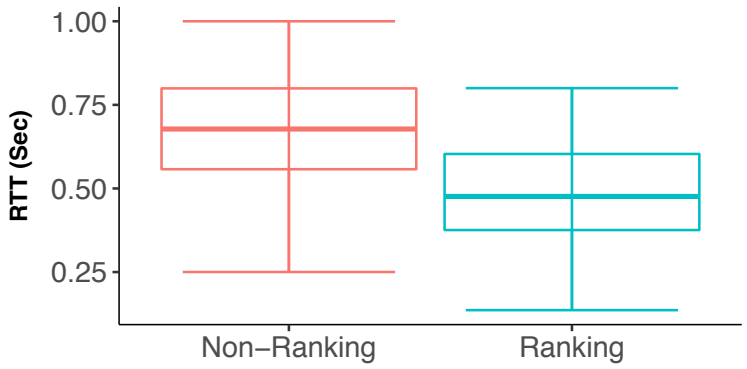


FIGURE 5.8: Ranking vs Non-Ranking in **MEDIUM** network load

RTT(Sec)	Non-Ranking	Ranking
1st Quantile	0.56	0.38
Mean	**0.68**	**0.49**
3rd Quantile	0.80	0.60

TABLE 5.2: Ranking vs Non-Ranking in **MEDIUM** network load

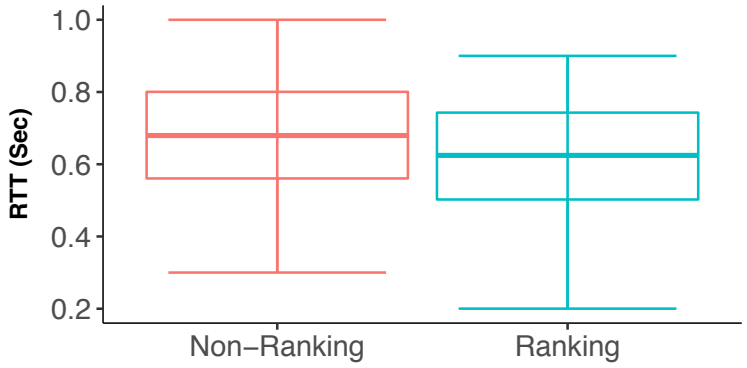


FIGURE 5.9: Ranking vs Non-Ranking in **HIGH** network load

RTT(Sec)	Non-Ranking	Ranking
1st Quantile	0.56	0.50
Mean	**0.69**	**0.62**
3rd Quantile	0.80	0.74

TABLE 5.3: Ranking vs Non-Ranking in **HIGH** network load

packets towards the destination's alternate and high latency RLOC.

5.4.2 Jitter

Apart from latency, jitter is another key metric that affects performance of interactive Internet applications. Some causes of Internet jitter include congestion in the core network as well as in the access links. Jitter may also be experienced when packets of the same flow traverse paths with different delays. In this work, the dynamic locator ranking meant that some of the traffic flows were being redirected at the source once the destination gateway rankings changed. This is a potential source of increased jitter for the traffic.

Results from the emulated network suggest that RLOC ranking and dynamic path configuration increased the overall jitter in the network. At low network load, the ranking approach had average jitter of 0.026 sec, against an average jitter of 0.023 sec for the non-ranking approach (Table 5.4). In this case, the ranking approach had 11% higher jitter than the non-ranking operation. At medium load, the ranking approach had an average jitter of 0.033 sec while the non-ranking approach had an average jitter of 0.027 sec (Table 5.5), meaning the ranking mechanism had 22% higher jitter than the normal operation.

As the network approaches congestion, both the ranking and non-ranking mechanisms experience similarly higher jitter. This is illustrated in Figure 5.14, where approaches have substantially increased jitter. At this point, the ranking approach had 0.039 sec jitter while

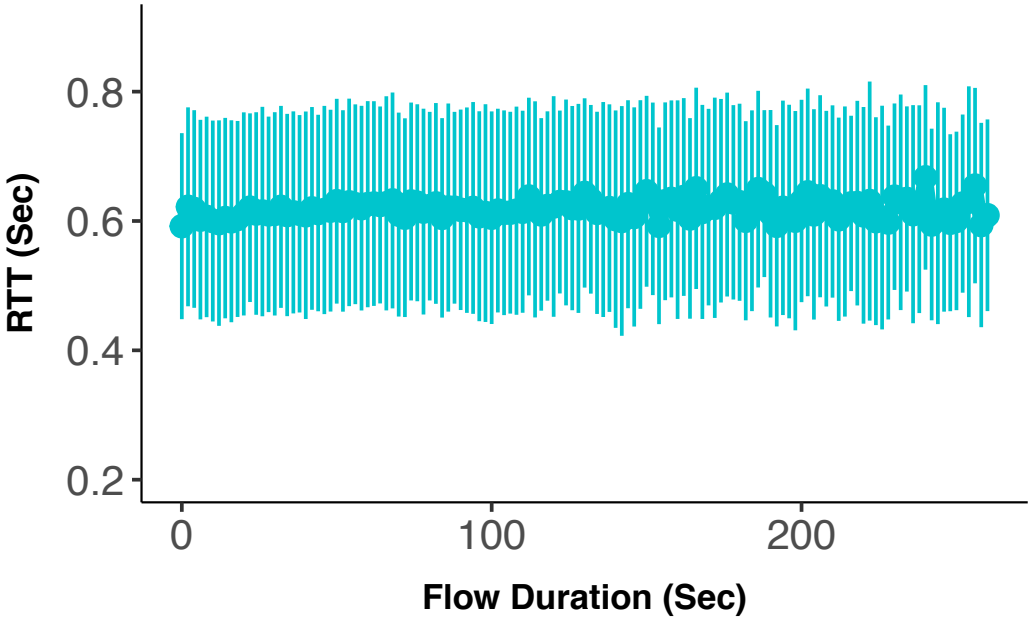


FIGURE 5.10: Dispersion and Mean RTTs for traffic flows in a Congested network operating WITH LISP gateway ranking

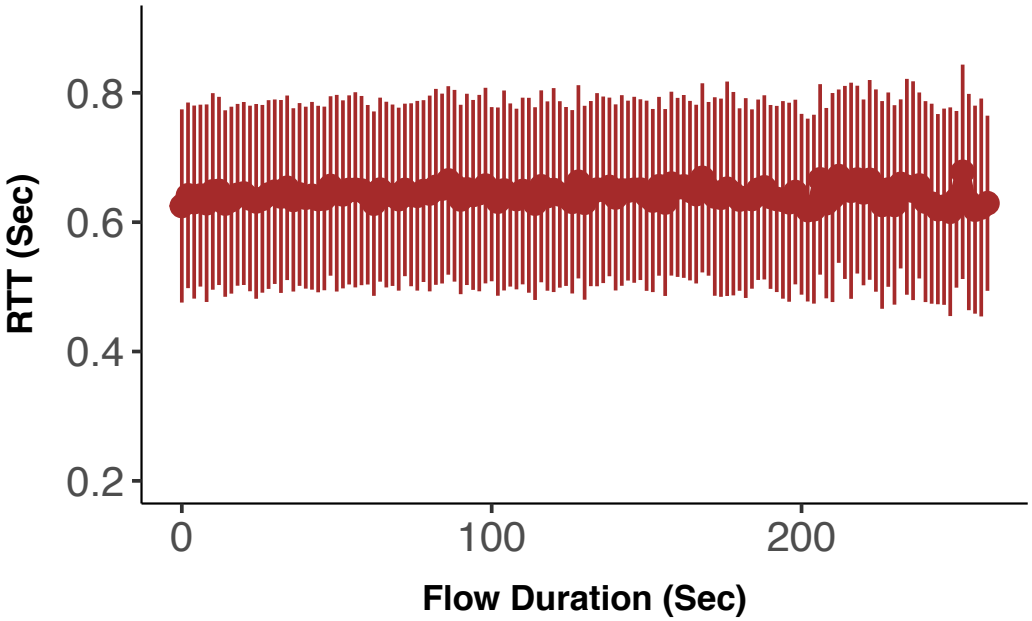


FIGURE 5.11: Dispersion Range and Mean RTTs for traffic flows in a Congested network operating WITHOUT LISP gateway ranking

the non-ranking approach had 0.037 sec jitter (Table 5.6). This means that the gateway ranking approach resulted in just about 5% higher jitter than the non-ranking approach.

Figures 5.12, 5.13, and 5.14 show the jitter for both ranking and non-ranking LISP operation, in low, medium and high network load.

The observed jitter in the experiments reveals that the process of RLOC ranking and path reconfiguration does increase the overall jitter in the network. In the presented model, a default end-to-end path is pre-installed through the SDN controller even before the initial path probing and RLOC ranking. However, the runtime ranking and re-ordering of the LISP gateways also means that the paths traversed by a flow's packets may change and, in the process, introduce jitter.

To deal with congestion problems at the source and destination LISP gateways, it could be helpful to have some kind of feedback channel, so that LISP routers can provide the source ITR with performance data and statistics. Another possibility would be to have the LISP locators set the explicit congestion notification (ECN) bits in the header of control messages exchanged between them. This would allow the source locator to adjust the load balancing metrics and reduce the amount of traffic sent through the congested destination locator. This mechanism would however require an extension/modification to the current LISP protocol.

5.4.3 Model Limitations

One challenge with the presented model is the assumption that NRENs are multi-homed. This is true to a large extent as, in general, for purposes of redundancy and resilience, NRENs will have more than one Internet attachment point. Another challenge is to do with independent selection of the remote RLOCs by the source network, which could result in violation of the destination network's preferences and policies. This could negatively impact on the destination's policies. For edge networks that have some form of cooperation, such as the case with NRENs within the UbuntuNet Alliance, a mutually beneficial approach would be to employ some level of coordination in selection of gateways. The concept of explicit traffic engineering coordination between networks, the NRENs would exchange information about their traffic and routing options, and all networks likely to be impacted by a routing change would negotiate the change (Mahajan, Wetherall, and Anderson, 2004). The mechanism for routing coordination might be implemented through coordination-based multi-agent reinforcement learning mechanisms (Zhang and Lesser, 2013; Guestrin, Lagoudakis, and Parr, 2002). In such a scenario, while each NREN would still aim to optimize its own traffic cost and QoS performance (latency), collectively, the NRENs could optimize the performance of some common preferred applications. Balancing the individual NREN's optimization objectives with those of the peering domains requires coordination and routing

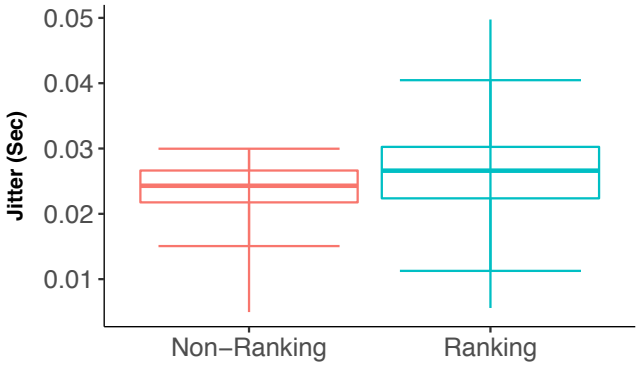


FIGURE 5.12: Lower Jitter for both ranking and non-ranking approaches during **LOW** load

Jitter(Sec)	**Non-Ranking**	**Ranking**
1st Quantile	0.021	0.022
Mean	**0.023**	**0.026**
3rd Quantile	0.027	0.030

TABLE 5.4: Jitter in ranking and non-ranking approaches during **LOW** network load

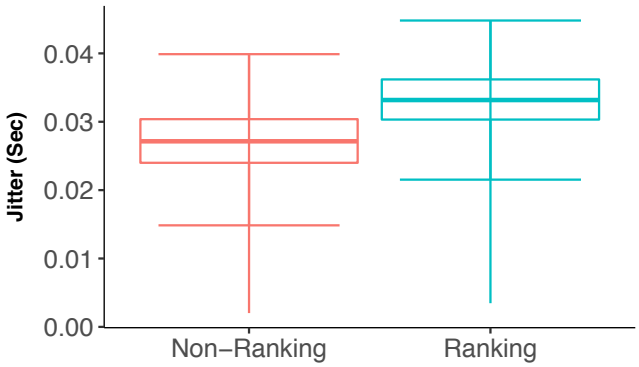


FIGURE 5.13: Ranking approach has higher jitter than non-ranking during **MEDIUM** load

Jitter(Sec)	**Non-Ranking**	**Ranking**
1st Quantile	0.024	0.030
Mean	**0.027**	**0.033**
3rd Quantile	0.030	0.036

TABLE 5.5: Jitter for ranking and non-ranking approaches during **MEDIUM** network load

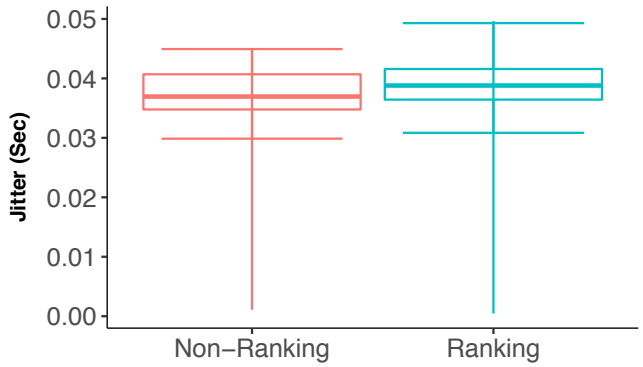


FIGURE 5.14: Both approaches have equally high jitter during **HIGH** load

Jitter(Sec)	Non-Ranking	Ranking
1st Quantile	0.036	0.035
Mean	**0.039**	**0.037**
3rd Quantile	0.042	0.040

TABLE 5.6: Jitter for ranking and non-ranking approaches during **HIGH** network load

cooperation among the peers. For the UbuntuNet Alliance, benefits from this level of cooperation could include better performance of the network applications, reduction in usage of intercontinental links, as well as reduction in the cost of inter-NREN traffic exchange.

5.5 Summary

This chapter has described a performance based traffic engineering mechanism that involves path measurement, gateway ranking, and SDN-based edge-to-edge path configuration. The first traffic engineering strategy presented was built with LISP and SDN to enable NRENs to dynamically rank gateways. Active measurements were periodically conducted from each NREN's LISP gateways to other remote LISP gateways. An SDN controller was used to set up end-to-end paths between every pair of NRENs, through lowest latency gateways. Results of the emulation-based evaluation of the strategy show that dynamic LISP path ranking was able to achieve 20 % lower latencies compared to the default LISP operation. These results confirm that dynamic LISP locator ranking does result in some performance gains in terms of reducing latency.

The LISP/SDN gateway ranking approach was unable to sustain performance gains (lower latencies) under high traffic load, and was ineffective when there was congestion in the core topology. This does suggest that just relying on edge networks' measurements of

end-to-end latency for ranking gateways may not always be effective. A possible improvement could be to employ a multi-dimensional ranking criteria, taking into consideration, for example, available bandwidth across the multihop path. Also, instead of a singular end-to-end path cost, it would be helpful to have a multi-hop cost of the underlay that can be adjusted dynamically during runtime. Such multi-hop performance costs would then have to be **employed in the inter-NREN core** to redirect traffic towards better paths.

Results from network emulation presented in this chapter pertain to the utility of using active measurements to rank destination gateway locators in situations where a destination has multiple locators. Results suggest that through dynamic ranking of the local and remote LISP locators, a source network can perform latency-based traffic engineering towards a destination without requiring direct cooperation of the destination network. Of course, the mechanism still requires the edge networks to cooperate in the sense of registering multiple gateways to the LISP mapping server. It is evident that by leveraging LISP capabilities through integration with SDN, there is potential for improving traffic exchange performance. For UbuntuNet NRENs, this would require that NRENs have control of the routing and traffic engineering across Internet exchange points, so as to be able to dynamically select routing paths among multiple ingress and egress links. This could provide important performance advantages for delay sensitive network applications between African NRENs.

Overall, the key results from chapter's experiments suggest that in under normal network conditions, and where the paths to RLOCs of multihomed edge network have significantly different RTTs, latency based ranking and selection of RLOCs does help to lower the overall latency in the network. There is need for further investigations into mechanisms that can enable African NRENs to perform collaborative performance based and application specific traffic engineering. A globally optimal solution requires coordination and collaboration among several domains that form part of the end-to-end path.

Chapter 6

Using SDN and Reinforcement Learning for Traffic Engineering

The previous chapter (Chapter 5) focused on how LISP and SDN could be used by the UbuntuNet Alliance NRENs to improve bandwidth utilization and reduce end-to-end inter-NREN latencies. Implementation of LISP at gateways was used to show how NRENs could select lower latency ingress and egress links, and to achieve generally lower end-to-end latencies. However, there was no mechanism for optimally distributing traffic inside the inter-NREN topology and, as a result, the mechanism was not able to offer any performance advantage when the topology was congested.

In this chapter, the SDN/LISP based traffic engineering framework, Reinforcement Learning is employed to guide distribution of traffic across multiple end-to-end paths in the inter-NREN topology. The focus is on the core topology, which links different NRENs through multiple PoPs and Open eXchange Points (OXPs). The mechanism makes use of network metrics to dynamically select topology links that have the potential to offer lower end-to-end latencies, as well as less congestion. The chapter looks at the utility of applying Reinforcement Learning to path selection, using network data obtained through an SDN controller. Results from network emulation show significant increases in total throughput when multipath routing is employed. Furthermore, emulation results show that where latency is the key metric for computing rewards, significantly lower latencies are achieved.

6.1 Introduction

It is not uncommon nowadays for networks to have multiple points of attachment and, in many cases, as in the case in the UbuntuNet topology, there are multiple disjoint or intersecting paths between a source and a destination. However, Internet protocols (TCP/IP and most transport-layer protocols) were built with the assumption of single-path forwarding only and do not utilize multiple routes for packet forwarding. Consequently, some traffic engineering research attention (Walton et al., 2016; Greene et al., 1991; Camacho et al., 2013;

Li, Wang, and Wang, 2011) is being given to approaches that leverage the high degree of redundancy in Internet topologies.

A study conducted on traffic routing between African NRENs, reported in Chapter 4, revealed that about 75% of the inter-NREN traffic traversed exchange points in Europe, resulting in much higher latencies. As shown in Section 4.2.3, intra-Africa traffic that traversed inter-continental links had a mean RTT of 409 ms, whereas traffic that was routed within the continent had a mean RTT of only 176 ms. Optimal path selection requires that the quality of links in the topology is continually evaluated to ensure that paths with better performance have a higher probability of being utilized (Desai and Patil, 2015). However, for large scale networks, the use of end-to-end active measurements for dynamic path ranking is neither efficient nor scalable (Jain and Pasquale, 2012).

The previous chapter showed that active measurements, conducted from edge networks, coupled with LISP gateway ranking, was not effective when the core topology was congested. This result further confirmed the assessment by Jain and Pasquale (2012) that, for large scale networks, the use of end-to-end active measurements for dynamic path ranking is neither efficient nor scalable. The experiments in this chapter are based on reinforcement learning, which was motivated by the fact that SDN controllers maintain a global view of the topology, and that they have at their disposal, a large volume and variety of network data.

A motivation for employing reinforcement learning inside the core topology was the notion that optimal path selection can be achieved when quality of links in the topology is continually evaluated so that paths with better performance are utilized more (Desai and Patil, 2015). This problem can be solved using reinforcement learning approaches, where experience gathered from iterative routing decisions can be used subsequently to select better paths. Some studies (Wolf et al., 2012; Rouskas et al., 2013) have shown that correlations learned from network controller data can be utilized to improve resource allocation and network performance. It is worthwhile therefore to investigate a data driven (Yin et al., 2014) approach where SDN nodes can use existing controllers' data.

This chapter discusses how the UbuntuNet can improve bandwidth utilization and reduce inter-NREN latencies by using Reinforcement Learning in an SDN-based core topology. This is done by implementing a reinforcement learning algorithm in the core SDN switches, applying network metrics to achieve dynamic selection of forwarding paths. More specifically, the chapter evaluates how the ability to discover alternate paths and dynamically configure routes based on path characteristics could help improve utilization and network performance of Africa's NRENs. Additionally, the chapter looks at the utility of applying Reinforcement Learning (RL) to path selection, using network data obtained through controller-based inter-switch probing. For evaluation, an SDN-based network was emulated in Mininet, applying the Q-learning Reinforcement Learning algorithm to distribute

traffic through multiple forwarding links. The principal aim of the strategy was to maximize throughput and reduce latency between NRENs.

6.1.1 Motivation for Centralized Multiagent Reinforcement Learning

The main idea for a centralized learning model is to move the learning logic away from the network nodes into a central controller that has more processing power than the individual nodes. Furthermore, since such as central controller is not directly responsible for packet forwarding, the processing strain on it does not directly affect the performance of the network. Additionally, a central controller has knowledge of the entire topology's traffic behaviour.

In standard Q-learning implementations, each agent learns a local forwarding policy by interacting with the immediate environment (Xu, Zuo, and Huang, 2014). The network nodes (agents) learn deterministic routing policies by monitoring network performance emanating from their forwarding decisions. For this reason, a common feature for existing Q-learning implementations is that every agent has mechanisms for monitoring network performance (Chun et al., 2014). Q-learning is dependent on the ability of learning agents being able to observe network traffic. As a result, traffic engineering implementations that employ Q-learning fail to build optimal routing policies when the traffic load is low, and are slow to adapt when network load reduces.

Largely, Q-learning routing decisions are computed locally at each agent. For example, the Q-routing algorithm (Choi and Yeung, 1996) requires that nodes exchange feedback on transmitted packets to calculate rewards and to make routing decisions locally. The agents iteratively build and locally adapt their packet forwarding policies. Additionally, each local agent needs to be able to exchange reward signals with other agents in the topology. The tasks of observing performance, exchanging rewards and computing routing policies at each node can easily become resource intensive and could potentially strain the forwarding devices, which can negatively impact the process of forwarding packets at the nodes.

The centralized reinforcement learning framework is based on a SDN paradigm, moving the logic for reinforcement learning away from the network agents into a central SDN controller. In an SDN topology, the SDN controller is expected to have superior computational power compared to network nodes, and is therefore more suited to handle the resource intensive role of parsing network data and applying it to a learning algorithm. Furthermore, an SDN controller has access to network-wide performance data, and therefore affords a better vantage point for reinforcement learning that quickly converge to an optimal network-wide policy.

The traffic engineering framework evaluated in this chapter consists of an SDN controller, a reinforcement learning engine and Q-learning agents, working together to dynamically configure optimal forwarding rules. The framework also includes a LISP mapping

system and locators that are used as gateways for the NRENs. The centralized SDN-based RL model is designed to address shortcomings of distributed RL agents, particularly the need for direct communication between the agents, the need for computational resources in each learning agent, and the need for agents to measure performance and provide feedback required for determining performance based rewards.

6.2 Structure of the Learning Framework

This section presents an SDN-based multipath traffic engineering mechanism to enable forwarding devices to adapt and improve network performance through learning from experience, through the application of the Q-learning Reinforcement Learning algorithm. The proposed centralized framework is designed to facilitate implementation of reinforcement learning in large scale networks without putting too much extra computational strain on network nodes. The design consists of an SDN controller, as well as a RL engine and Q-learning agents, working together to dynamically configure optimal forwarding rules.

The Q-learning agents, which run in the SDN nodes, are responsible for making packet forwarding decisions based on the Q-learning values. Each agent receives rewards from the central controller to update the Q-values, as well making packet forwarding decisions based on Q-values. The agent module uses the Q-values for each interface to compute the number of packets that should be forwarded through that link. In the experimental topology, each node (SDN switch) is modelled as a state s, and a next-hop switch as s'. Performance rewards are calculated based on packet delay between s and s' as well as the available capacity on the $s \leftrightarrow s'$ link.

The functions of the Q-learning controller (Figure 6.1) are divided into four different modules: (1) multipath computation module responsible for computing and keeping track of all paths in the topology; (2) a passive measurements module responsible for obtaining network statistics from SDN nodes across the topology and keeping track of the performance and capacity of the links; (3) an active measurement module responsible for injecting learning packets into the network and monitoring hop-by-hop performance; and (4) Q-learning module that combines the knowledge from passive monitoring and active measurements to compute and assign rewards to nodes. Both passive and active measurement techniques were employed so as to compute both utilization metrics, such as packet count and available bandwidth, as well as performance metrics, such as delay, jitter and packet loss.

In the proposed centralized Q-learning traffic engineering framework, the agents do not need to directly exchange performance rewards. Instead, a controller is responsible for injecting learning packets into the network, as well as retrieving usage statistics from network

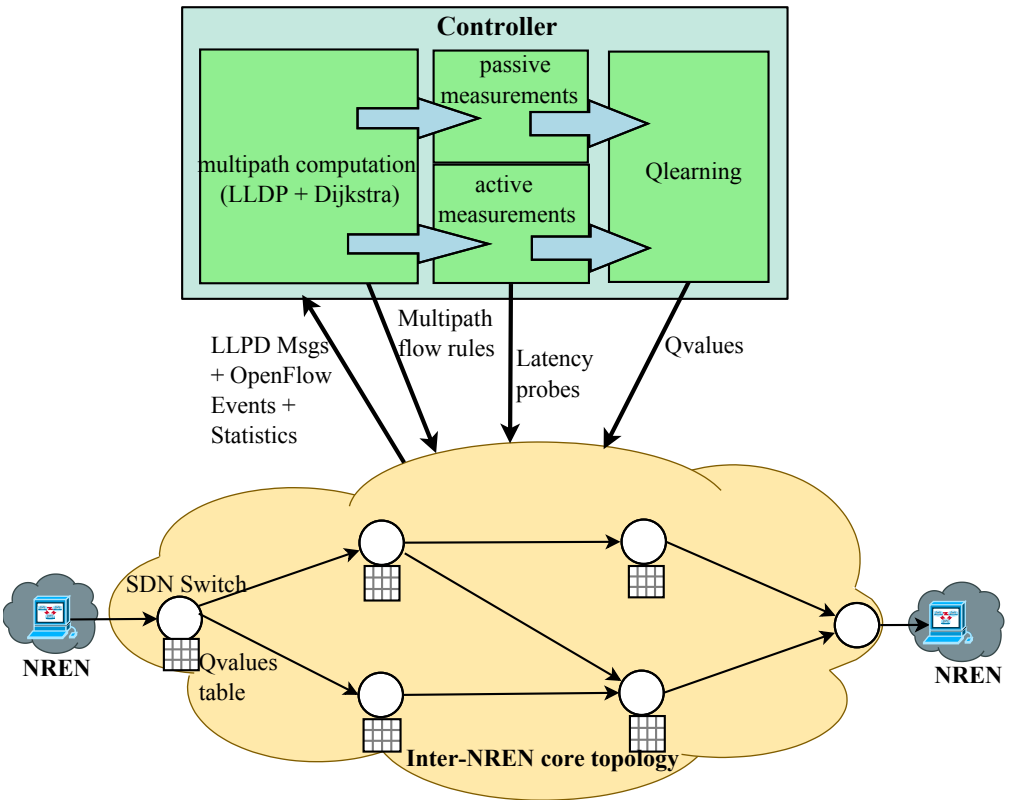


FIGURE 6.1: Q-learning based Traffic Engineering framework

nodes. The central Q-learning controller is responsible for monitoring performance of forwarding decisions across the topology. The controller has a Q-learning module that continually applies Q-learning on the network statistics and performance data to learn about the traffic engineering performance. The controller also calculates rewards for the exchange of traffic between agents, with rewards being given for each forwarding link/interface. The controller is also responsible for transmitting the rewards to network nodes as well as calculating a global routing solution based on Q-values of all nodes in every path.

Furthermore, the Link Layer Discovery Protocol (LLDP) (Wang, He, and Su, 2015) was used to obtain link and switch states from the SDN topology. All forwarding paths were stored and used as alternate routes for each source-destination switch pair. The use of LLDP helped to maintain a global view of network topology and traffic.

6.2.1 Learning and Path Selection

There are generally two modes of selecting actions in Q-learning implementations - exploration and exploitation. At the start of each Q-learning episode, a random value $(0 \leq x \leq 1)$ is generated and, if such a value is less than ϵ, then the learning is conducted in exploration mode. Otherwise, if the generated value is equal to or more than ϵ, the learning episode runs in exploitation mode.

In exploration mode, all the possible state-actions are randomly explored and their Q-values updated. This helps to ensure that the performance of all outgoing links are monitored and their corresponding Q-values updated, even if they do not provide the best performance. In exploitation mode, selection of actions is based on the learned Q-values, where the action associated with the highest Q-value is chosen. The transition between exploration and exploitation phase is controlled by a probability variable ϵ. With probability of $(1 - \epsilon)$, the action with the maximum Q-value is selected and, with probability ϵ, a random action is selected.

In this Q-learning framework, there are primarily two types of network traffic: the user/application traffic and learning packets' traffic. User traffic originates from the source node's application layer destined to another destination node. For this type of traffic, the Q-learning routing aims to achieve the best possible path based on the specific performance constraints, such as latency and bandwidth. To achieve the QoS intended by the Q-learning implementation, the forwarding decisions for user traffic are based on exploitation mode, in which the interfaces with the better prevailing Q-values are used to forward more traffic. The process of selecting the optimal end-to-end path for user traffic is thus achieved by iteratively selecting the forwarding link based on Q-value at each switch.

The other type of traffic - the learning packets - are initiated by the controller's active measurements module, and are sent between every adjacent pair of network nodes. These packets are used for actively measuring the hop-by-hop performance (latency, jitter, loss). For learning packets, an exploration mode is employed so that all the interfaces of the network nodes are measured.

6.2.2 Active Measurement Module

Active performance metrics, such as latency and packet loss, are important for characterising quality of Internet paths. In a single-domain or federated topologies, it would be possible to measure one-way delay through passive mechanisms (De Vito, Rapuano, and Tomaciello, 2008), such by sampling actual traffic packets and evaluating the time difference between the source and destination nodes. However, this would require the two nodes to have accurate time synchronisation, such as using the NTP (network time protocol). In centralized architectures, such as SDN, it is possible to use the controller to passively measure delay by calculating the time difference for a packet to move between two nodes. This can be done by sampling the actual traffic, or by monitoring controller generated measurement packets. In this proposed framework, the controller's active measurement module is responsible for measuring latency and packet loss between all the topology's adjacent SDN switches. First, the controller needs to monitor and regularly update the delay to each of the switches by sending probe packets. To measure performance between adjacent switches, the module

transmits a learning packet between every pair of adjacent switches in the topology. To measure the latency between Switch A and Switch B, the module transmits from the controller at time T_{AB0}, a learning packet $packet_{AB}$ through A destined for B. Switch A simply forwards the packet to Switch B. When the packet is received by Switch B, a $packet\_in$ OpenFlow control message, tagged with the $packet_{AB}$ id, is triggered and forwarded to the controller. The controller receives the $packet\_in$ packet at T_{ABt}, upon which the controller is able to determine the time delay between switches A and B based on the time difference from T_{AB0} to T_{ABt}. The measurement packet thus moves from the original source - the controller - through the source switch A, to the destination switch B, and then back to the controller. Since T_{AB0} and T_{ABt} are recorded at the controller when the learning packet, respectively, leaves and comes back to the controller, the time difference from T_{AB0} to T_{ABt} thus represents the round-trip time from the controller, through Switches A and B, back to the controller. The latency computation between Switch A and Switch B therefore needs to take into account the delay between the controller and each of the two switches. The delay between adjacent switches is thus computed as $T_{ABt} - T_{AB0} - \Delta_A - \Delta_B$; where Δ_A and Δ_B is the delay between the controller and switch A and switch B respectively. The packet loss measurement performed by the SDN controller are done at the IP packet level, without the service of transport layer packet reordering or retransmission.

6.2.3 Passive Measurements Module

Learning data in this framework is obtained through both active and passive measurements between all adjacent SDN switches. The statistics module is responsible for collecting passive measurement data from all the switch ports, thus being able to determine utilization (used bandwidth) in all the links. Capacity is measured in terms of the available link bandwidth relative to the number of flows and packets coming through each switch interface. The interface-level statistics (number of flows, packet count) are used together with performance data from learning packets to calculate a reward value that is used to update the Q-values.

6.3 Q-learning module

In this framework, each of the core topology's switches conduct active and passive measurements to monitor links to the next-hop neighbours. A network controller collects performance and utilization data from all the core switches and links, and employs Q-learning to dynamically determine the best forwarding links at each node. Performance rewards are calculated based on packet delay between s and s' as well as the available capacity on the $s \leftrightarrow s'$ link. The action represents the forwarding option taken, which is the node's interface/link

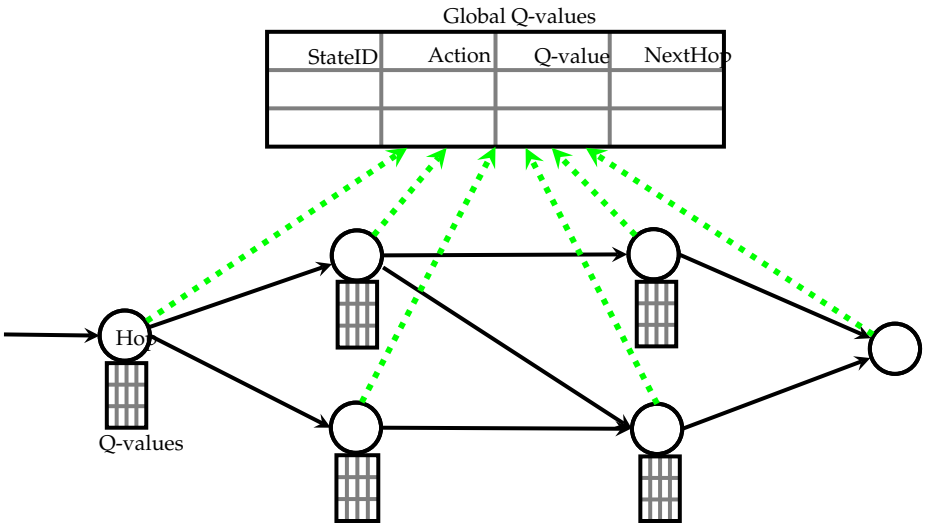


FIGURE 6.2: Local and global Q-values tables

used to reach the next hop. For each state-action, a Q-value record $< s, a, s', Q(s, a) >$ is maintained, consisting of the state identifier, action, a pointer to the next state for the action, and a Q-value associated with the interface (action).

6.3.1 Q-values Table

The Q-learning implementation consists of a local Q-values table at each network node (SDN switches and LISP locators), as well a global aggregation table managed by the SDN controller (Figure 6.2).

The controller uses data obtained through the active passive measurement module to calculate a reward value that it uses to update the Q-values. The Q-learning module is responsible for continuous adjustment of forwarding rules.

6.3.2 Q-learning Rewards

As the traffic flow commences between the source and destination hosts, a network controller commences learning episodes in which the controller performs active and passive measurements between all adjacent SDN switches. Active measurements are used to measure latency and bandwidth between switches, whereas passive measurements are used to obtain the topology's load and residual capacity. Capacity in this sense is measured in terms of available link bandwidth and switch throughput capacity (packets per second) relative to the number of flows and packets in each link and switch/interface.

After each learning iteration, the networks statistics and path metrics are applied to an aggregation function to calculate the reward value for each forwarding link, which is computed as a composite value of path metrics K. Each metric value is scaled as a ratio of the

maximum plausible value for the metric. For example, path delay is scaled as a fraction of the maximum possible delay for any link in the topology (1000 ms was used in the experiments). A measured path metric value that is equal to or greater than the maximum expected value is scaled to 1. For example, link delays of 100 ms and 500 ms are respectively scaled to the values 0.1 and 0.5, whereas link delays of equal to or greater than 1000 ms are scaled to 1. Better links are those whose scaled delay values are closer to zero.

In terms of bandwidth, two metrics were considered. First, the available bandwidth in a link is important as it determines how much traffic can be transmitted through the link. The link capacity metric is scaled as a ratio of the available link bandwidth versus some maximum topology link capacity (100 mbps was used in the experiments). Links with better capacity are therefore those with the scaled capacities closer to 1. Secondly, the level of utilization in a link is indicated by the ratio of a link's capacity that has already been utilized. Utilisation ratio can be used to gauge the potential of reaching congestion levels for a particular link. Link congestion is thus computed as a fraction of the link load versus the capacity of the link. For example, two links with 100 mbps and 80 mbps, with traffic load of 50 mbps and 8 mbps respectively, are computed to have congestion levels of 0.5 and 0.1, respectively. Lower utilization values result in better rewards.

A weighting variable Λ is used to aggregate the path metric in K into a reward value $r(s,a) = \sum_{i=1}^{n} \Lambda_i K_i$; where $\sum_{i=1}^{n} \Lambda_i = 1$, and K = {latency, available-bandwidth}. The Λ_i associated with each metric i depends on the optimisation objective and is described further with experiments' descriptions in Section 6.4.

The Q-values records consist of tuples with a state identifier, action, a pointer to the next state for each action, and reward value for the action. A state represents a hop in the network topology. Since each hop handles traffic going to different destinations, a state in this work is defined by the node name and a destination's IP prefix (Listing 6.1). This means each hop may have several states associated with it, one for each destination IP prefix for traffic going through it.

LISTING 6.1: Definition of State

```
class STATE:
      id = (nodeID + destPrefix)
      type = hostType // 0 if SDN switch, 1 if LISP router
      actions = []
```

The actions available at SDN switches comprise the outgoing links/interfaces (Listing 6.2).

On LISP-based states (LISP gateway locators), actions consist of a set of a destination's gateway locators obtained from the LISP mapping system. Furthermore, Q-values at LISP locators are translated into and used to update locator weight values as described in Section 5.2.2. The locator weights are used for determining rations of for splitting traffic towards destination gateways for outgoing traffic.

LISTING 6.2: Definition of Action

```
class ACTION:
      id = interfaceName or LISP routerName
      type = hostType // 0 if SDN switch, 1 if LISP router
      nextState = switch_out_port or LISP router IP
      reward = 0
```

After taking a state-action option and the reward r having been calculated, the state-action's Q-value is updated using the Q-learning equation $Q^*(s,a) = (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')$,
where s, a are the state and action respectively, s' is the resultant state after action a, $\alpha \in (0,1]$ models the rate of updating the Q-values, i.e how fast new information overrides previous information, and $\gamma \in (0,1]$ represents a discount factor that scales the importance of the immediate reward (obtained for the action at s) versus maximum reward obtainable for actions at the subsequent state s'.

A record $< s, a, s', Q(s,a) >$ is then written into the Q-values table. On SDN switches, Q-values are transformed into interface priorities that determine the next hops for each destination. The process of selecting the optimal end-to-end path for user traffic is achieved by iteratively selecting, at each router, the next hop that has the highest priority and weight.

The Q-learning settings were set with the following parameters: learning rate α = 0.5; and discount factor γ= 0.5 (Wang and Wang, 2006). The learning rate α models the rate of updating the Q-values once new information is applied, i.e how fast new information overrides the existing Q-values. On the other hand, γ scales the value of the reward at the current state versus the potential reward that might be obtained at the next possible state. A γ value of zero would mean a completely greedy approach in which the decision is based only on the current reward, with no regard to rewards further down each possible path.

6.3.3 Packet Blocks

The Q-learning framework is designed to take advantage of a multipath environment to achieve traffic engineering objectives, which in this case include minimising end-to-end latencies and increasing throughput. Flow-based multipath traffic engineering approaches force all packets belonging to a flow to follow the same path. Such approaches fail to aggregate the multipath bandwidth (Jo et al., 2002). Multipath TCP attempts to solve this problem

by setting up multiple Internet paths between a pair of hosts, while presenting a single TCP connection to the application layer(Mendiola et al., 2016a). The multipath TCP mechanism requires modification of the end hostss TCP/IP stack to support the establishment of TCP connections and the transmission and reception of data over multiple paths (Mendiola et al., 2016a). On the other hand, simultaneous use of multiple forwarding paths for a flow has the potential to aggregate link capacities and provide higher throughputs. For this reason, the Q-learning forwarding mechanism in this study was designed to forward packets over a set of paths based on Q-values assigned to the interfaces/links. For each forwarding node in the topology, the SDN controller maintains sets of outgoing interfaces that are part of the same source/destination multipath set. The solution also includes a packet reordering mechanism that resequences out-of-order packets before delivery to the destination end host.

In Q-learning, a single action is selected at each iteration based on a user-defined function $f(s)$ (Boyan and Littman, 1994). Usually, the action that is associated with highest Q-value is selected. In terms of packet forwarding, this would entail forwarding the packets through the interface that has highest Q-value. With this approach, contiguous packets are more likely to go through the same single path, and have a good chance of arriving at the destination in the same order in which they were transmitted. However, since only the best interface is utilized all the time, the approach does not ensure that all the bandwidth available on the multiple paths is bundled and utilized together.

Alternatively, a probabilistic approach could be implemented where the Q-values are treated as the odds with which actions should be selected. With this approach, an incoming packet can be forwarded through any particular interface with the probability given by the Q-value. Packets are thus distributed onto the multiple outgoing interfaces, such that higher Q-value interfaces have a higher chance of being used to carry the outgoing traffic. This approach would ensure that all the links in a multipath set are utilized, thereby increasing the potential throughput between a pair of end nodes. However, the approach increases the likelihood of contiguous packets taking different routes and therefore experiencing different amounts of delays. This has the negative consequence of packets arriving at the destination out of order, which generally results in the receiving node discarding some packets. This packet loss not only degrades the quality of service experienced by end nodes, but also wastes bandwidth through packet retransmissions that are required in connection-oriented communication.

The approach employed in this framework is a modification of the probability approach. Instead of probabilistically forwarding individual packets based on the Q-values, the proposed approach forwards bursts of contiguous packets (blocks). The main motivation for employing a block approach is to minimize the probability of packet reordering that results from contiguous packets taking different paths (Kandula et al., 2007). The size of the packet

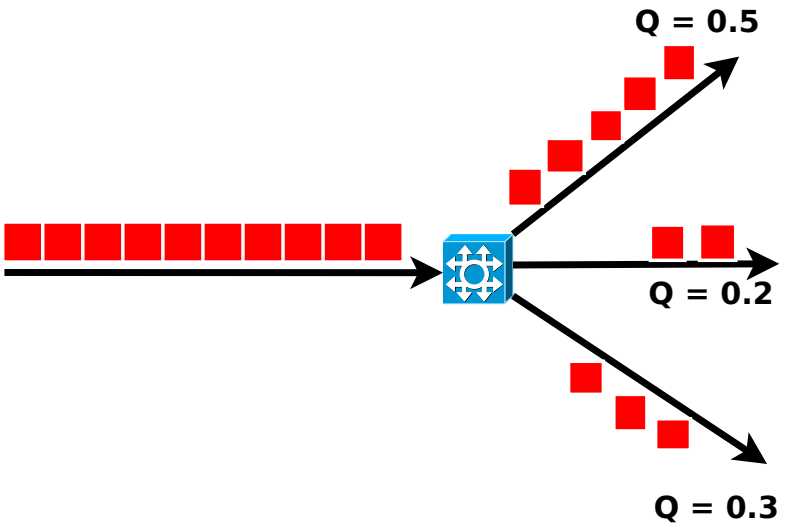


FIGURE 6.3: Burst packet splitting: 10 packets are split into the three outgoing interfaces commensurate with the respective Q-values of 0.5, 0.2, and 0.3

blocks corresponds to the Q-values associated with each outgoing interface/link in the multipath set. For each traffic flow coming through a forwarding node, each outgoing interface is assigned a block of packets proportional to its Q-value. In Figure 6.3, block sizes are obtained by multiplying the Q-values ($0 \leq Q \leq 1$) with a general block size of 10 packets.

In the experiments, the general block size was 200 packets, and each link would be assigned a ratio of the 200 packets depending on their respective Q-values. Also, for each source-destination pair, a maximum of three best paths were selected, and only the links that are part of those selected paths were assigned packet blocks.

6.3.4 Packet Reordering

Multipath forwarding at packet level entails that packets of the same flow potentially travel on different paths and experience varying delays. This raises the probability that contiguous packets would arrive at the destination with variable delays and possibly out of order. This has the negative consequence of increasing jitter and packet loss. Furthermore, contiguous packets arriving at the destination out of order necessitates buffering and packet reassembly, either within the network or at the receiver.

To minimize the impact of out-of-order packets on performance, multipath algorithms implement queues and reorder packets at the receiver node before passing them on to the application. The proposed multipath framework makes use of a packet reordering buffer implemented by Banfi et al. (2016). Implementing the corrective measure at the edge switch ensures that there is no need to change the networking function of receiving devices, thus making the multipath mechanism transparent to the end devices. The reordering buffer at the receiver edge maintains a record of the next expected packet sequence number, and

temporarily holds each incoming packet unless such packet's sequence number matches the expected one. The buffered packets are released in their proper order to the final destination node when either the buffer is full or the number of buffered packets for a flow exceeds a threshold. In the experiments, a buffer size of 200KB, and a threshold of 100 packets was used. The buffer memory for each flow is freed when a TCP FIN packet is received.

6.4 Experimental Evaluation

This section describes a set of experiments that were conducted to evaluate the simulated SDN topology, with reinforcement learning being used to adjust forwarding rules. The primary purpose of the experiments was to evaluate throughput and latency improvements that are achieved by the proposed traffic engineering solution. Furthermore, the evaluation investigated the impact of multipath traffic forwarding with regards to other QoS metrics such as jitter and packet loss.

The evaluation was performed in an emulated network built in Mininet (Heller et al., 2012). The emulated network was based the current and planned topology of the UbuntuNet Alliance network (UbuntuNetAlliance, 2016). A detailed description of Q-learning implementation in the topology is given in Section 6.3. The gateway of each NREN in the emulated topology was implemented with LISP, in the same manner as in Section 5.2.2 and Section 5.2.2. Some NRENs in the topology, such as KENET, have multiple attachment points to the UbuntuNet. For such NRENs, each LISP gateway router was therefore configured with two external network interfaces, and thus two locators, as well as one internal EID interface that is used as a default gateway for end hosts.

6.4.1 Emulating the UbuntuNet Topology

The 2016 state of UbuntuNet topology (depicted in Figure 6.4) forms a ring through the alliance's Points of Presence (PoPs) in Cape Town, Mtunzini, Maputo, Dar-es-salam, Nairobi, Amsterdam, London, and back to Cape Town (UbuntuNetAlliance, 2016). NRENs in landlocked countries are connected via terrestrial fiber optic cables to the coastal PoPs: from Lusaka to Cape Town and Dar-es-salam; Lilongwe to Lusaka; Luanda to Cape Town; and Kigali and Kampala to Nairobi.

This experimental evaluation builds on the SDN/LISP experimentation described earlier in Section 5.3, but now based on the UbuntuNet core inter-NREN topology, as well as the inclusion of reinforcement learning in network nodes. The UbuntuNet core topology was thus emulated with each of the PoPs in the Alliance represented with an SDN switch.

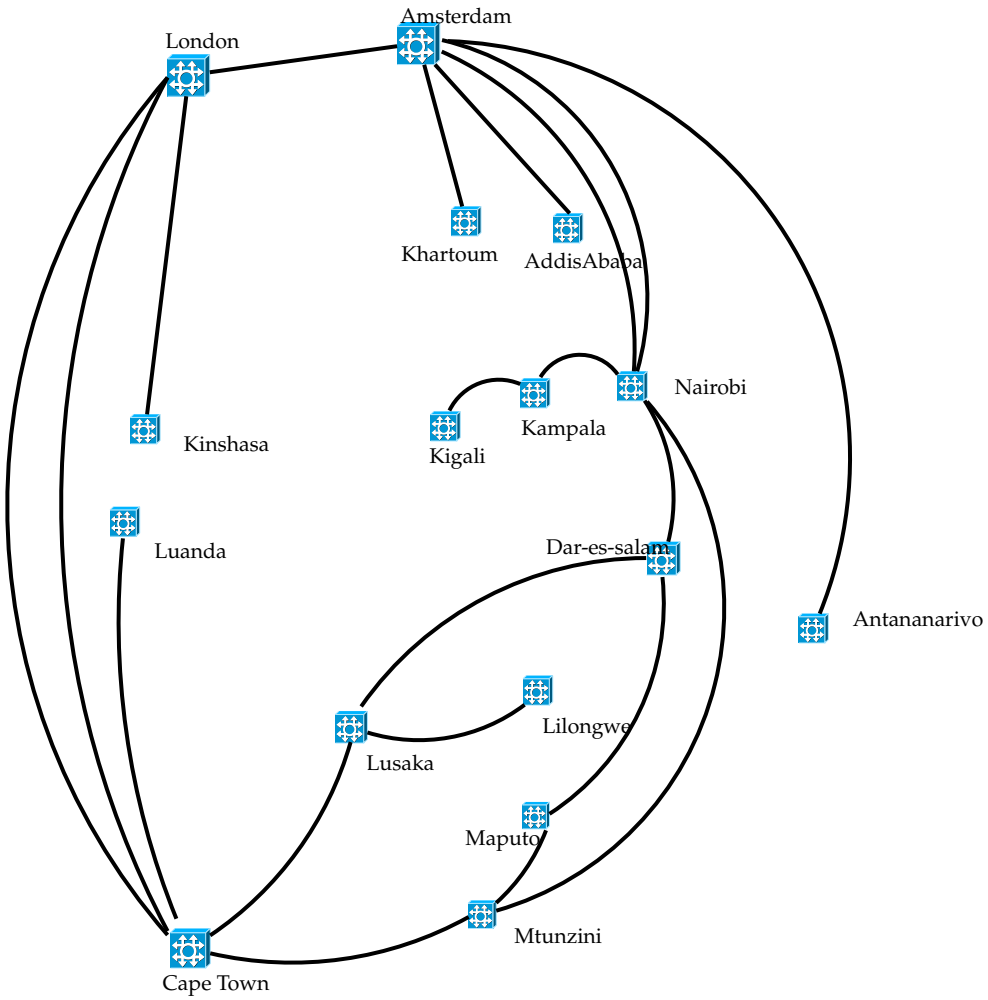


FIGURE 6.4: UbuntuNet Alliance Topology

NRENs were also modelled as switches connected the core topology switches. The UbuntuNet topology map indicates that four of the alliance's members - Sudan, Ethiopia, Madagascar and the Democratic Republic of Congo (DRC) - are connected to UbuntuNet through either London or Amsterdam PoPs. Traceroute measurements suggest DRC is directly connected to the London IXP, while the other three are directly connected to the Amsterdam IXP.

Network Latency and Link Capacities

There were two key metrics for the emulated experimental topology: link delays and link capacities. In terms of link capacities (bandwidth), the UbuntuNet Alliance has, as of June 2016, had a total link capacity of 2.18 Gbps linking the alliance's region to Europe (UbuntuNetAlliance, 2016). This capacity comprises 2 STM-4 links (2 X 622 Mbps) on the east cost of Africa, from Mtunzini to Amsterdam, with landing points in Maputo, Dar-es-Salam and Nairobi. On the west African cost, the capacity comprises of a single STM-4 (622 Mbps)

TABLE 6.1: Inter-NREN link distances and delays used in the experiment

Source	Dest	Sea(km)	Land(km)	Delay(ms)	Bw(mbps)
London	Amsterdam	386	45	2	256
London	Cape Town	11301	45	56	124
London	Kinshasa	8981	330	46	15
Amsterdam	Nairobi	11788	487	61	124
Amsterdam	Khartoum	7579	834	42	15
Amsterdam	Antananarivo	12836	485	66	15
Amsterdam	Addis-Ababa	8678	896	47	15
Cape Town	Luanda	3049	7	15	15
Cape Town	Lusaka	0	3133	15	62
Cape Town	Mtunzini	1526	132	8	62
Lusaka	Lilongwe	0	711	3	15
Lusaka	Dar-es-Salam	0	1942	9	62
Mtunzini	Maputo	583	224	4	62
Mtunzini	Nairobi	3237	700	19	62
Dar-es-Salam	Nairobi	325	485	4	62
Dar-es-Salam	Maputo	2598	20	13	62
Nairobi	Kampala	0	660	3	62
Kampala	Kigali	0	514	5	15

and 2 STM-1 links (2 X 155 Mbps), from Cape Town to London. The backbone between the UbuntuNet countries is made up of STM-4 links (622 Mbps): between Dar-es-Salam and Cape Town via Lusaka; between Mtunzini and Nairobi; and between Mtunzini/Maputo and Dar-es-Salam. There are also 2 STM-4 links (2 X 622 Mbps) between Nairobi and Kampala, and a single STM-4 between Kampala and Kigali.

Link delays between NRENs were estimated based on cable lengths, as calculated using 'road-trip' as well as port-to-port distances between the NRENs' PoPs. Terrestrial road trip distances between inland cities are obtained using Google Maps. Distances between sea port PoPs, as well as inter-continental links were obtained using PortDistance [1]. These distances were used to estimate link delays between PoPs, translating every 200km to 1 ms latency (Landa et al., 2013).

Table 6.1 lists the links in the topology with their terrestrial and marine distances, the estimated delays, and bandwidth used in the experiment. Figure 6.5 shows the topology, in which the link weights represent the link delays used in the experiments. Where multiple physical links exist between a pair of PoPs, a single aggregated link is used in the experiment. The overall link capacities were scaled down by a factor of 10 to cope with bandwidth limitations of the Mininet network emulator.

[1] https://www.searates.com/reference/portdistance/

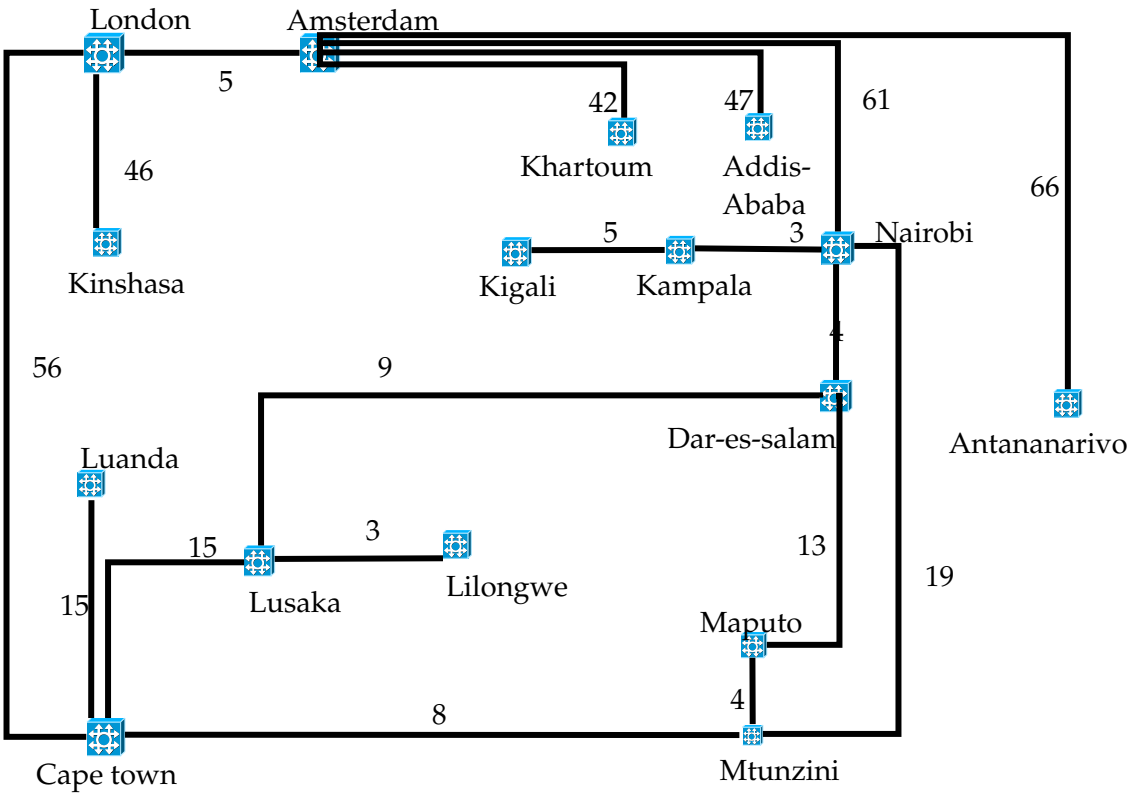


FIGURE 6.5: UbuntuNet Alliance Topology

Network Loops in SDN Topology

For the evaluation, the UbuntuNet Alliance core topology (Figure 6.5) was emulated as an SDN network, with each of the PoPs in the alliance represented with an SDN switch. An Ryu OpenFlow controller was connected to the Nairobi PoP chosen to host the SDN controller because it is the most central PoP in the topology. NRENs were also modelled as switches connected to the core topology switches.

As Figure 6.5 shows, the UbuntuNet has redundant links and loops in its topology. Dealing with network loops in a layer-2 topology requires the use of Spanning Tree Protocol (STP) to determine the loop-free paths (spanning-tree) that link every pair of switches in the network. The use of a spanning tree path between pairs of the SDN switches results in redundant links in the network being disabled. As a result, end-to-end communication gets restricted to single paths even though redundant and possibly better paths are available. STP eliminates the multipath capability that should otherwise be available in the topology. In the emulated topology, the Link Layer Discovery Protocol (LLDP) (Wang, He, and Su, 2015) was used to obtain link and switch states in the topology, and all paths were stored and used as alternate routes for each source-destination switch pair. The use of LLDP helped maintain a global view of network topology and retain a multipath environment.

6.4.2 LISP Gateways in UbuntuNet

Given the topology in Figure 6.5, if the NRENs were to implement a mechanism for dynamic multipath selection, as well being able to announce and discover multiple gateways, it would be possible for a pair of NRENs to exchange traffic through multiple gateways. In the emulated topology, NRENs connect to the UbuntuNet core topology through multiple LISP gateways, with each gateway facing a PoP. As an illustration, consider the traffic between the Kenyan NREN, KENET, and the South African NREN, TENET. KENET is attached to UbuntuNet PoP in Nairobi, whereas TENET is connected to two PoPs: in Cape Town and in Mtunzini. Thus, if the topology were to have a mechanism for dynamic multipath selection, traffic between KENET and TENET could flow through any of the four paths:
(1) $CapeTown \leftrightarrows London \leftrightarrows Amsterdam \leftrightarrows Nairobi$;
(2) $Mtunzini \leftrightarrows Nairobi$;
(3) $Mtunzini \leftrightarrows Maputo \leftrightarrows DarSalam \leftrightarrows Nairobi$; and
(4) $CapeTown \leftrightarrows Lusaka \leftrightarrows DarSalam \leftrightarrows Nairobi$.

In the experiments, NRENs gateways were implemented with LISP locators. It was thus possible for example, for KENET to announce two gateways, one reachable through Amsterdam and the other through Mtunzini (Figure 6.6). TENET was also able to announce two gateways, Cape Town and Mtunzini (Figure 6.7). Given this configuration, the NRENs would explicitly specify the source and destination gateways for traffic exchange. When setting up a flow, the source gateway selects one of the destination's gateways through which to encapsulate packets for the target NREN. The locator ranking approach employed in Section 5.2.2 was used in each of the gateways for selecting the destination gateway. The core topology, which in the emulation is based on SDN and Reinforcement Learning, is then only responsible for optimally routing the packets between the source and destination gateway.

6.4.3 Experiments

This section describes experiments that were designed to compare performance of single path forwarding, where all traffic flowing between a source and destination follow the same path, versus a multipath forwarding, where packets flowing between a source and destination take different paths depending on observed performance. Selection of paths in the multipath configuration was dynamic and was influenced by network performance and utilization, as well as through the use of the Q-learning algorithm to continuously modify packet forwarding rules.

The single path configuration was used as the control experiment for when static paths are pre-configured. In that case, the end-to-end paths do not depend on performance or prevailing network conditions. In the single path mode, edge-to-edge paths are pre-configured between each source and destination network at the start of the experiment, employing the

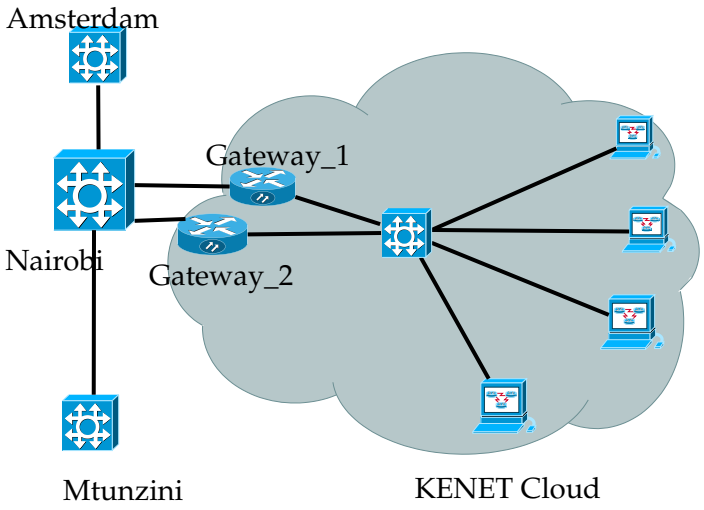


FIGURE 6.6: KENET dual-homed to UbuntuNet core topology through two LISP gateways facing Amsterdam and Mtunzini

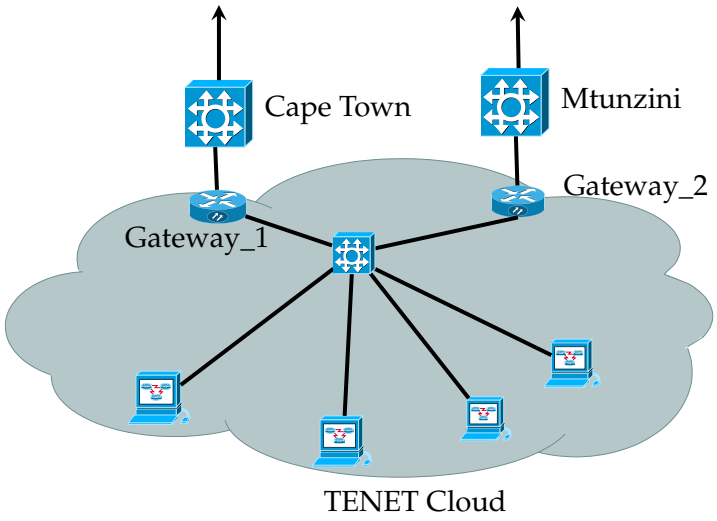


FIGURE 6.7: TENET topology dual-homed to UbuntuNet through two LISP gateways in Cape Town and Mtunzini

same methodology described in Section 5.2.2 where the SDN controller computes the shortest paths between every pair of NREN.

The following experiments were thus conducted to evaluate if and how performance of the UbuntuNet core topology would improve with implementation of SDN and Reinforcement Learning (Q-learning):

1. **Single lowest latency path forwarding:** This experiment was set up with the aim of evaluating performance when a single path is selected between each pair of NRENs. At the start of the experiment, the controller determined and configured the single lowest latency path between every pair of networks. This was implemented by using only delay between adjacent switches to compute the shortest path.

2. **Single highest capacity forwarding:** This experiment was set up with the aim of evaluating performance when a single path is selected for each flow. The controller determines and configures the single highest capacity path between every pair of NRENs in the topology. This was implemented by computing the cost of each link from the residual bandwidth, such that links with higher capacity are assigned a lower cost, as described in Section 6.3.2.

3. **Multipath forwarding based on latency and capacity:** This experiment was set up to evaluate the performance when the switches forward traffic through multiple paths towards the destination host. The rewards and Q-values are calculated based on delay between switches, as well as the residual capacities in the link. The switches then use the Q-values to split the flow packets probabilistically, in fixed size blocks, to the egress links' Q-values.

4. **Multipath forwarding based on latency:** In this experiment, multiple links paths were used for each flow, but the rewards and Q-values were influenced only by the path delay. The egress link that was part of the shortest delay path to the destination was awarded higher rewards, and thus carried more traffic to the destination.

5. **Multipath forwarding based on capacity:** In this setup, multiple paths were used, with the reward and Q-values being influenced solely by links' residual capacity. The egress link that is part of the path with the highest capacity receives higher rewards and Q-values, and therefore carries more traffic.

Each experiment measured four aspects of network performance: latency; throughput; jitter; and packet loss. The experiments were conducted in the environment described in Section 5.3. Latency and Jitter measurements were conducted in a network environment with medium background traffic as described in Section 5.3.2. Measurement traffic between end hosts in the network was generated using Iperf (Tirumala et al., 2005). This means that

the reported end-to-end throughput, delay and packet loss are based on the application level performance.

To conduct the measurements, each end host would randomly select a remote host and initiate a TCP-based iPerf transmission for a random length of time ranging from 1 sec to 300 seconds. The decision to have flows of varying durations was to emulate Internet IP traffic, which is characterised by both flows and long flows, with at least 45% of Internet streams being short flows lasting less than 2 seconds, but also that 98% of all the streams lasted no more than 15 minutes (Brownlee, 2005). After completion of a flow, the host would again randomly select another remote host and run the measurement again. All the measurement hosts looped through this process for at least 30 mins.

6.5 Results

The first section of the results (Section 6.5.1) presents the aggregate results for the entire topology, while the second section (Section 6.5.2) zooms into and looks at the performance between a single source-destination pair - Cape Town and Nairobi. These two PoPs have multiple routes between them and are analysed to highlight the impact of the multipath solution for individual source/destination networks. The results are represented in the form of boxplots, as well as using empirical cumulative distribution function $(ECDF)$. The function ECDF(X) represents the probability that a data value from the results set will be less than or equal to a value X.

6.5.1 Network wide performance

The first set of results are for the performance between all source/destination pairs in the topology.

Throughput

The evaluation compared the end-to-end throughput achieved for single path forwarding mechanism (Experiments 1 and 2) against a dynamic multipath forwarding mechanism (Experiments 3, 4 and 5). Table 6.2 provides a summary of throughput for the experiments 1 - 5.

The multipath experiments achieved significantly higher average throughput (Mbps) than the single path setup. Experiment 5, in which paths were ranked and selected based only on the available capacity in the forwarding links, achieved the highest average throughput of 32.66 Mbps, with an inter-quartile range (IQR) from 22.52 Mbps to 41.199 Mbps. It is not surprising that this configuration had the highest average throughput given that the paths with the highest capacity were chosen over lower capacity paths, regardless of other

TABLE 6.2: Throughput between Nairobi and Cape Town

(Mbps)	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Mean	13.02	13.64	23.23	19.98	**32.66**
1st Quantile	4.61	4.61	13.68	9.00	22.52
3rd Quantile	14.93	19.29	31.02	23.87	41.20

QoS parameters such as latency. Of the three multipath experiments, the lowest throughput was achieved for Experiment 4, which rewarded paths based on latency only, without regard to available bandwidth. Experiment 4 thus achieved mean throughput of 19.98 Mbps and inter-quartile range of 9 Mbps to 23.87 Mbps. Experiment 3 combines the path ranking mechanisms used in Experiments 4 and 5, such that path latency and capacity are equally weighted in determining the path rewards and ranking. Consequently, the approach in Experiment 3 achieves performance that falls between the two multipath mechanisms. Experiment 3 shows a mean throughput of 23.23 Mbps with IQR of 13.68 Mbps to 31.02 Mbps.

Between the single path forwarding experiments (1 and 2), Experiment 2 achieved a slightly higher average throughput of 13.64 Mbps compared to 13.02 Mbps for Experiment 1. Although the mean latencies for the Experiments 1 and 2 are almost the same, the Inter-Quartile Range in Table 6.2 and the distribution in Figure 6.8 indicates that Experiment 2 achieved a larger ratio of high throughput flows than Experiment 1. Experiment 2 had an IQR of 4.61 Mbps to 19.29 Mbps, which is higher than the IQR of 3.23 Mbps to 14.93 Mbps for Experiment 1. It is not surprising that Experiment 2 had better throughput than Experiment 1, considering that, for each pair of end nodes, the Experiment 2 mechanism selected the highest capacity path, whereas Experiment 1 selected the lowest latency path between every pair of nodes without regard to available bandwidth.

Latency

Reducing inter-NREN latency is one of the key motivating factors for this research. This section therefore evaluates performance of the proposed traffic engineering framework in terms of end-to-end latencies.

Figure 6.9 presents the average latencies per flow as measured using iPerf in Experiments 1-5. The recorded latencies are the average end-to-end packet delays for each iPerf flow between end nodes in the topology. From the graphs, it can be observed that the lowest mean latencies (mean of means) are from experiments 1 and 4, which had mean latencies of 53 ms and 69 ms respectively. Although experiment 1 is single path and experiment 4 is multipath, they share a similarity in that they both rank and select paths based on latencies. In experiment 1, packets traverse the single lowest latency path from source to the destination.

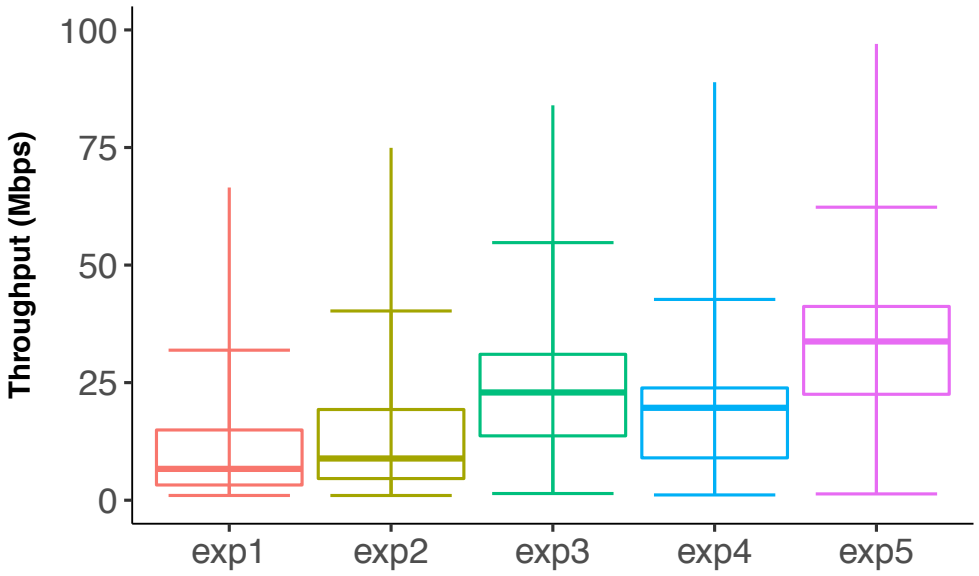


FIGURE 6.8: Distribution of throughput for data flows between end nodes in the topology for experiments 1-5

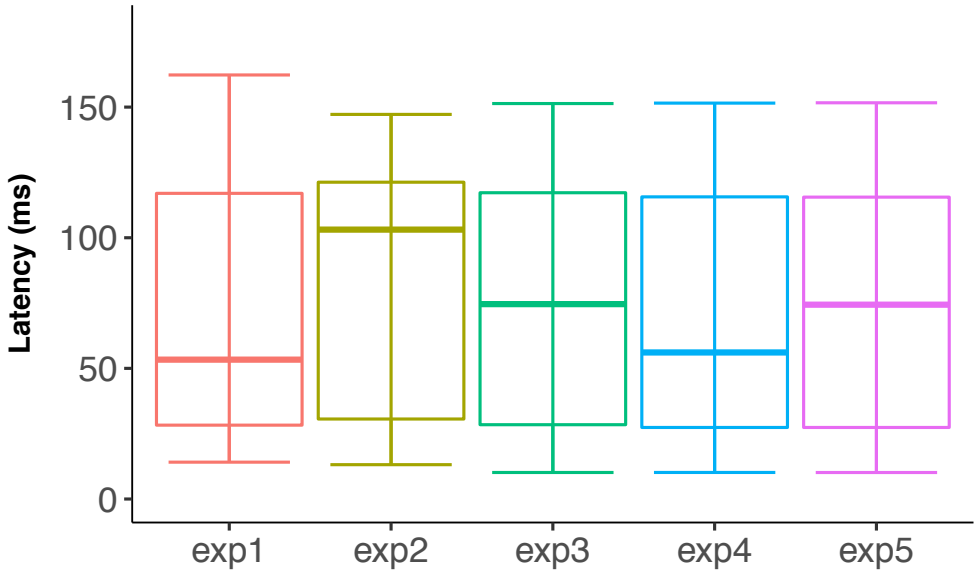


FIGURE 6.9: Distribution of end-to-end latencies for data flows between nodes in the topology for experiments 1-5

On the other hand, experiment 4 packets are distributed to multiple paths based on latency, with the lower latency paths carrying more packets.

The highest mean latency is recorded for experiment 2 and 5, both of which rank and select forwarding links based only on capacity. Experiment 2 has the highest mean latency of 103 ms, followed by experiment 5 with mean a latency of 75 ms. Given that the highest capacity paths in the UbuntuNet topology also have the higher latencies (Figure 6.4, inter-continental links have higher capacity than intra-continental), a traffic engineering mechanism that only considers capacity should indeed result in overall high latencies. Experiment 3, which ranks paths based on both capacity and latency, achieved a mean latency of 74 ms, almost the same as experiment 5 ranks only based on capacity.

Packet Loss

The packet-level multipath forwarding increases the probability for packets to have varying delays, experience higher jitter, and arrive out of order (Kandula et al., 2007). The corrective measure of buffering and packet reordering increases the potential for packet loss, as some packets time out and others get dropped when buffers get full. Also, flows traversing multiple paths that have markedly different end-to-end delays experience higher levels of delay variances and out of order arrivals. It should be expected therefore, that multipaths with higher delay imbalances should result in higher jitter and packet loss.

Each of the three multipath configurations (Experiments 3, 4, and 5) recorded some packet loss in at least 10 % to 15 % of the flows (Figure 6.10). In contrast, the single path experiments (1 and 2) experienced almost zero packet loss in all the flows. Of the flows that experienced packet loss (lossy flows), 20 % had loss of more than 1 % per flow (Figure 6.11).

The experimental results showed that for the three multipath experiments, the lowest packet loss was recorded in the multipath configuration that selected the paths based only on latencies (Experiment 4). As Table 6.3 shows, the lowest average packet loss of 0.76 % with IQR of 0.01 % to 1.98 % is recorded for Experiment 4 (where paths are rewarded and ranked based only on latencies). In contrast, Experiment 5, which rewards and ranks paths based on bandwidth alone, recorded a packet loss of 3.21 %, with an IQR of 0.03 % to 7.10 %. When latency is the only metric for selecting paths, there is a higher probability that a flow's packet take paths that are more similar in terms of end-to-end delay, and thus experience minimal delay imbalance, reordering, and loss.

Jitter

Jitter is caused by deviations in packet delay, and this can easily be aggravated when packets of the same flow follow different paths. Just like latency, multipath experiments (3,4,5) experienced much higher jitter compared to the single path configurations(1,2). As Figures

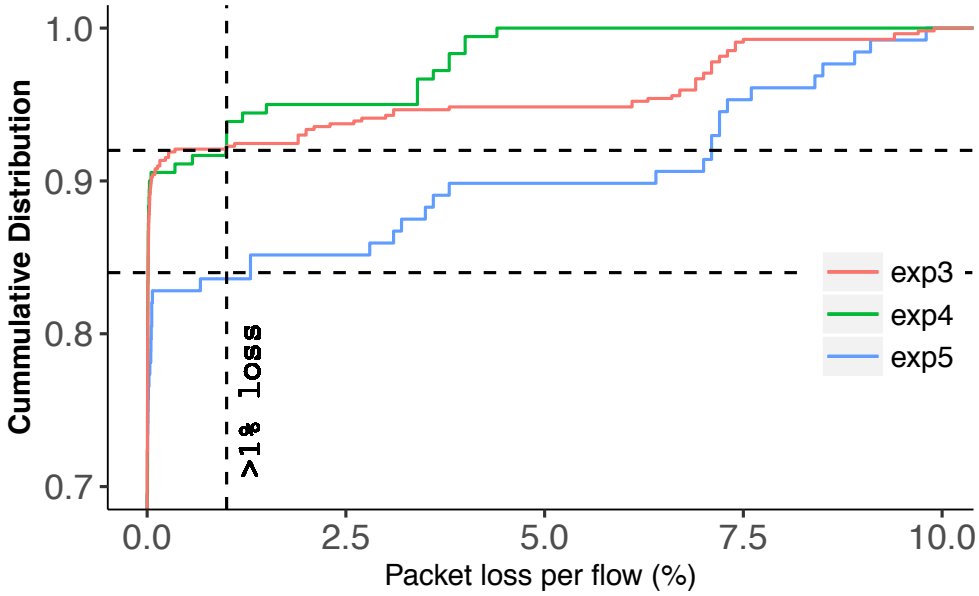


FIGURE 6.10: Percentage of significantly lossy flows (more than 1% packet loss per flow). About 84% of Experiment 5 flows, and 92% Experiment 3 and 4 flows had less than 1% packet loss, i.e about 26% of Experiment 5 flows, and 8% of Experiment 3 and 4 flows had more than 1% packet loss per flow.

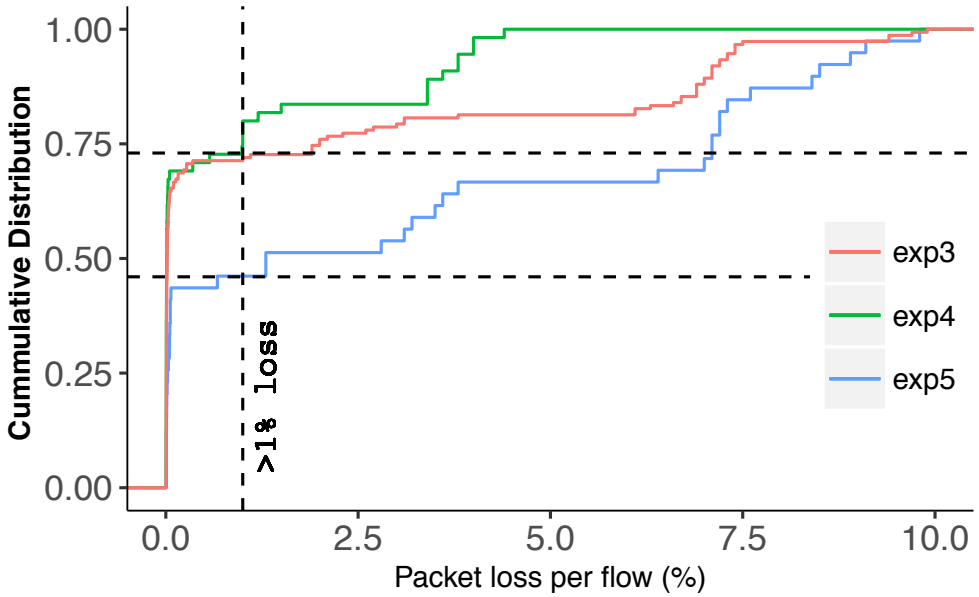


FIGURE 6.11: Magnitude of loss for lossy flows (excluding loss-less flows). About 55% of Experiment 5 flows, and about 25% of Experiment 3 and 4 flows had packet loss of more than 1% per flow.

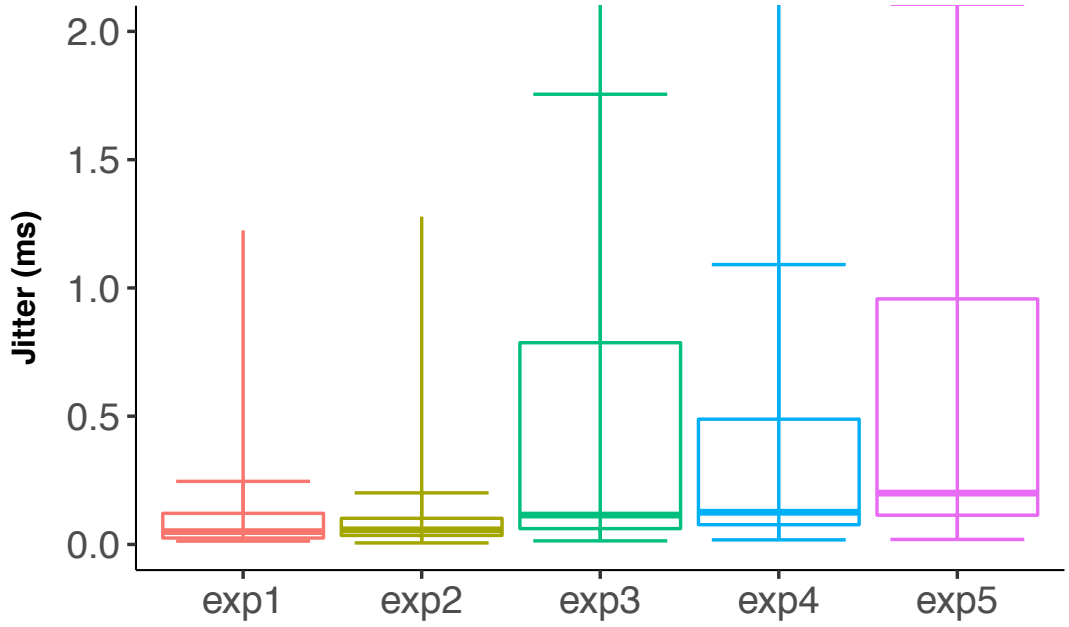


FIGURE 6.12: Experiments 1 and 2 (single path forwarding) had average jitter of 0.12 ms and 0.11 ms respectively. In contrast, Experiment 5 (multipath path selection with no regard to latency) recorded the highest average jitter values, averaging 0.88 ms.

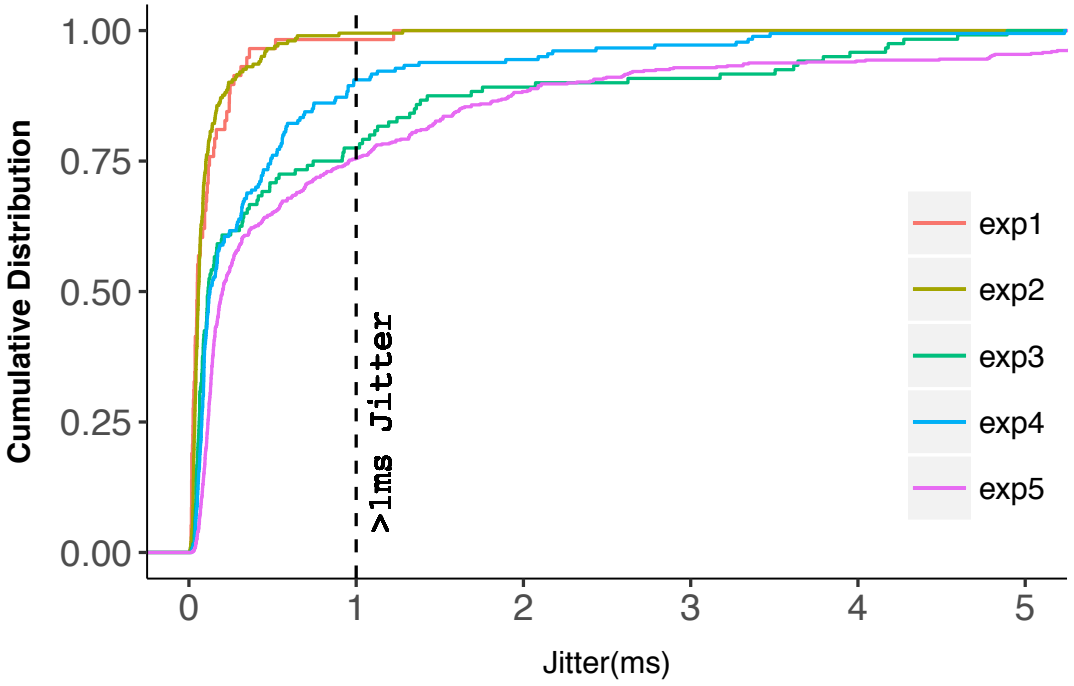


FIGURE 6.13: At least 25 % of the multipath flows (exp3,exp4,exp5) had jitter of more than 1 ms. In contrast, almost 100% of single path flows had less than 1 ms jitter.

TABLE 6.3: Packet loss per flow

Loss (%)	Exp 3	Exp 4	Exp 5
Mean	1.62	0.76	3.21
1st Quantile	0.01	0.01	0.03
3rd Quantile	1.98	1.00	7.10
Max.	9.90	4.40	9.80

TABLE 6.4: Jitter per flow

Jitter (ms)	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Mean	0.12	0.11	0.69	0.43	**0.88**
1st Quantile	0.03	0.04	0.06	0.08	0.11
3rd Quantile	0.12	0.10	0.77	0.49	0.96
Max.	1.22	1.28	4.89	5.23	9.94

6.12 and 6.13 show, the single path configuration (Experiments 1 and 2) experienced almost negligible average jitter of 0.12 ms and 0.11 ms respectively. Experiment 5 (which does not consider latency in path selection) recorded the highest average jitter values, averaging 0.88 ms. This was expected, considering that Experiment 5 did not consider latency in selecting the multipaths, such that there was higher probability for a flow's contiguous packets to traverse paths that have very dissimilar delays and thus experience higher delay variances (jitter).

Although the multipath configurations (Experiments 3,4,5) show insignificant mean jitter of only 0.69 ms, 0.43 ms, and 0.88 ms respectively (Table 6.4), the distribution in Figure 6.13 indicates that at least 25 % of the multipath flows had jitter averaging above 1 ms.

6.5.2 Performance between Nairobi and Cape Town

This section focuses on traffic exchanged between Cape Town and Nairobi. These two PoPs have multiple paths between them, and are used here to highlight the potential impact of multipath traffic engineering for networks that are part of the UbuntuNet Alliance topology. Just like in the general case, the network performance metrics considered here are throughput, latency, jitter and packet loss.

Throughput

Figure 6.14 shows the range of throughput achieved in the experiments. In general, the three multipath traffic engineering experiments (Experiments 3,4,5) achieved throughput

TABLE 6.5: Throughput between Cape Town and Nairobi

(Mbps)	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Mean	26.90	40.80	53.39	49.17	**61.79**
1st Quantile	20.39	36.60	42.45	39.10	52.12
3rd Quantile	32.99	45.80	62.65	58.36	77.60
Max.	48.67	49.68	87.05	79.01	88.18

TABLE 6.6: Latency between Cape Town and Nairobi

Latency (ms)	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Mean	**38.91**	81.70	67.37	49.89	74.13
1st Quantile	37.82	74.21	55.46	29.27	29.14
3rd Quantile	39.55	89.18	82.16	68.11	119.12

levels that were considerably higher than single path packet forwarding (Experiments 1 and 2).

Among the multipath experiments, it was expected that Experiment 5 would have much higher throughput given that it favoured higher capacity links. As expected, Experiment 5 had the highest throughput among the three, achieving average throughput of 61.79 Mbps, whereas Experiment 3 and Experiment 4 obtained average throughput of 53.39 Mbps and 49.17 Mbps, respectively (See Table 6.5).

Experiment 5, in which rewards and path ranking were calculated based on available bandwidth only, achieved the highest mean throughput of 61 Mbps, and an Inter-Quartile Range (IQR) of 52 Mbps to 77 Mbps. The lowest mean throughput among the multipath configurations was from Experiment 4, which achieved a mean throughput of 49 Mbps and an IQR of 39 Mbps to 58 Mbps. Experiment 4 calculated rewards and ranked paths based on latency only. Experiment 3 calculated rewards and ranked paths using both latency and available capacity through each path, and achieved a mean throughput of 53 Mbps and IQR of 42 Mbps to 62 Mbps.

Between the two single path configurations, the highest throughput was achieved in Experiment 2, where a single highest capacity end-to-end path was used for all packets belonging to the same flow. Overall, Experiment 1 achieved the lowest throughput. This was expected as it used only a single path for forwarding, and calculated rewards and Q-values based on link delays without any regard to link capacity.

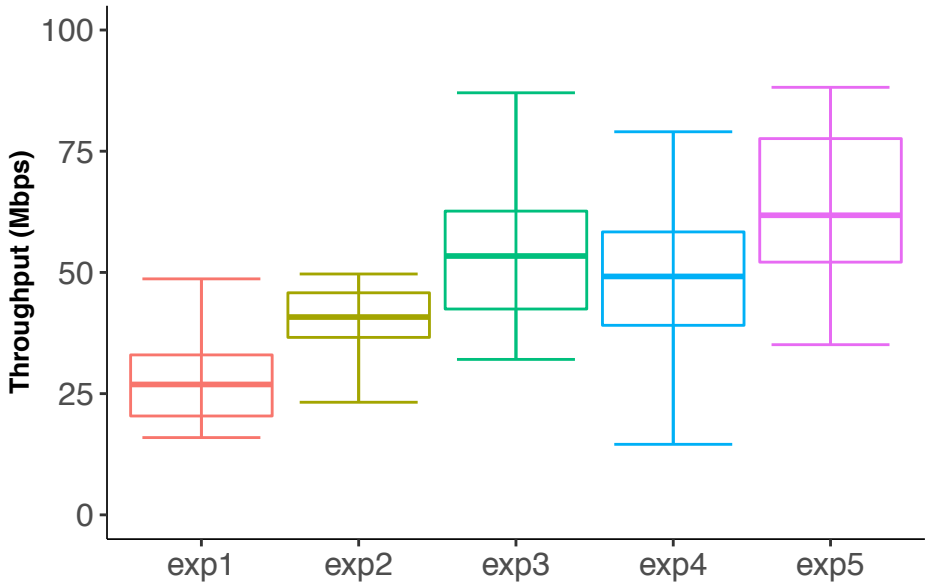


FIGURE 6.14: Throughput measurements between Cape Town and Nairobi, highest throughput of 61 Mbps per flow attained in Experiment 5 (multipath ranking based only on available bandwidth). Experiment 1 (single path selected based only on latency, without regard to available bandwidth) achieved the lowest per flow throughput of 27 Mbps

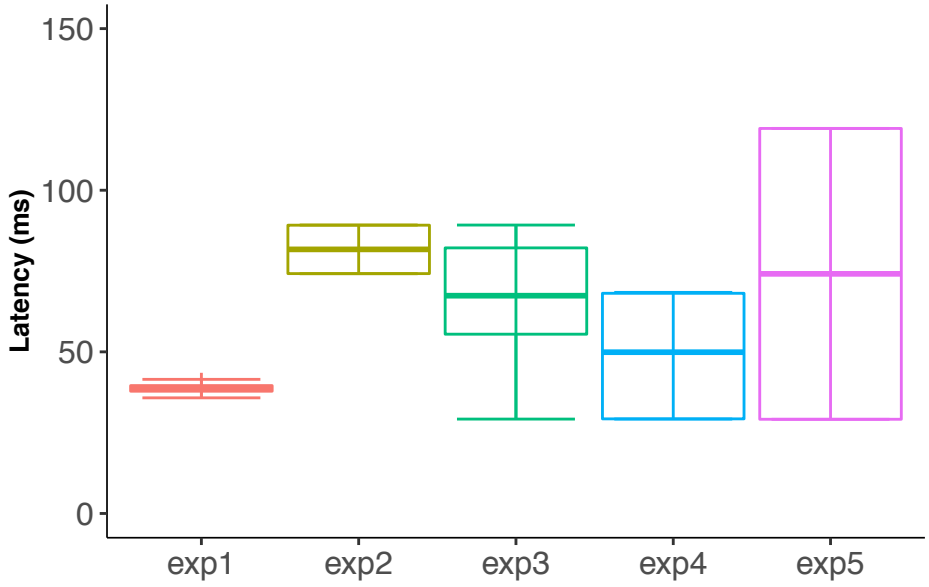


FIGURE 6.15: Latency between Cape Town and Nairobi, Experiment 1 (lowest latency single path) achieved the lowest mean latency of 38 ms. Experiment 4 (multipath selected lower latencies) also achieved a relatively low average latency of 49 ms but with wider dispersion of latencies with IQR 29 ms to 68 ms.

TABLE 6.7: Jitter between Cape Town and Nairobi

Jitter (ms)	**Exp 1**	**Exp 2**	**Exp 3**	**Exp 4**	**Exp 5**
Mean	0.11	0.10	0.63	0.37	1.66
1st Quantile	0.02	0.03	0.07	0.07	0.08
3rd Quantile	0.11	0.09	0.65	0.42	1.61
Max.	1.13	1.19	11.57	4.47	28.58

Latency

Figure 6.15 and Table 6.6 present latencies between the Cape Town and Nairobi in the emulated topology. Experiment 1 achieved the lowest mean latency of 38 ms. This should be expected, considering that this configuration uses the lowest latency path. Experiment 1 also had the least dispersed latencies, with an IQR between 37 ms and 39 ms and this was because all packets travelled on the same lowest latency path, thereby having very small deviations in the packets' end-to-end latencies. Experiment 4 also achieved a relatively low average latency of 49 ms but with wider dispersion of latencies than Experiment 1, ie IQR 29 ms to 68 ms. This is the case because packets were forwarded through multiple paths of different latencies.

The highest mean latencies were recorded in experiments that only used available capacity for path selection (Experiment 2 and 5). While Experiment 2 used single path forwarding and Experiment 5 used multipath forwarding, they both did not give any consideration to path latencies. Instead, the two approaches calculated rewards based on path capacities, and thus high latency paths had just about the same chance of being selected depending on their bandwidth capacity.

Jitter

As can be observed from Figure 6.16, the single path configuration in Experiments 1 and 2 experienced the least amount of jitter. On the other hand, all the three multipath approaches had significant levels of jitter. Experiment 5 had the highest jitter, as expected due to its non-consideration for delay when calculating forwarding rewards and Q-values.

Packet Loss

Figure 6.18 shows that single path experiments (Experiments 1 and 2) did not experience any packet loss, whereas the multipath experiments (Experiment 3,4,5) experienced a considerable levels of packet loss. At least 10 % of flows in Experiment 3 and 4 had packet loss of at least 2.5 %, whereas at least 20 % of Experiment 5 flows experienced packet loss of 2.5 % (See Table 6.8). Experiment 5 experienced substantially higher packet loss than the rest of

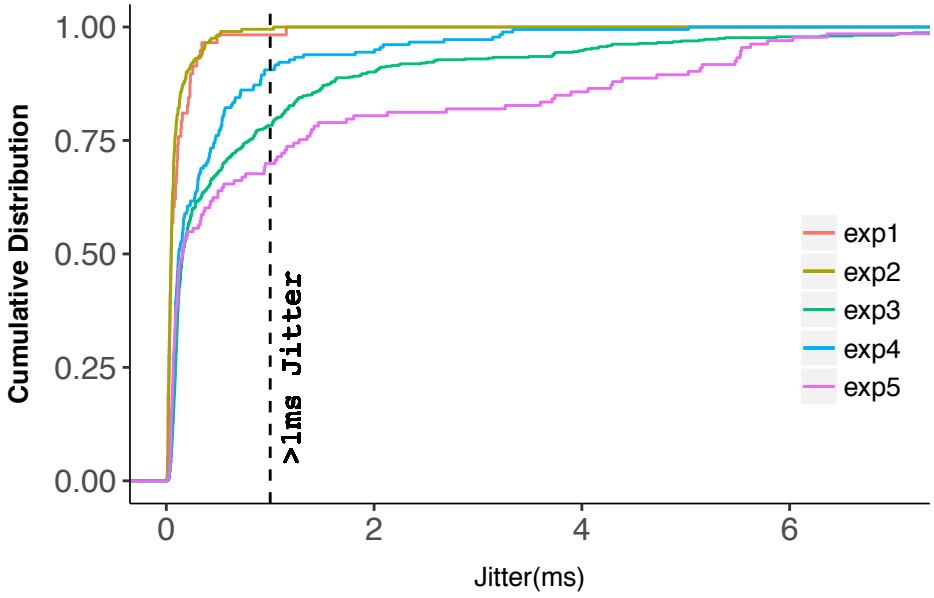


FIGURE 6.16: Jitter between Cape Town and Nairobi

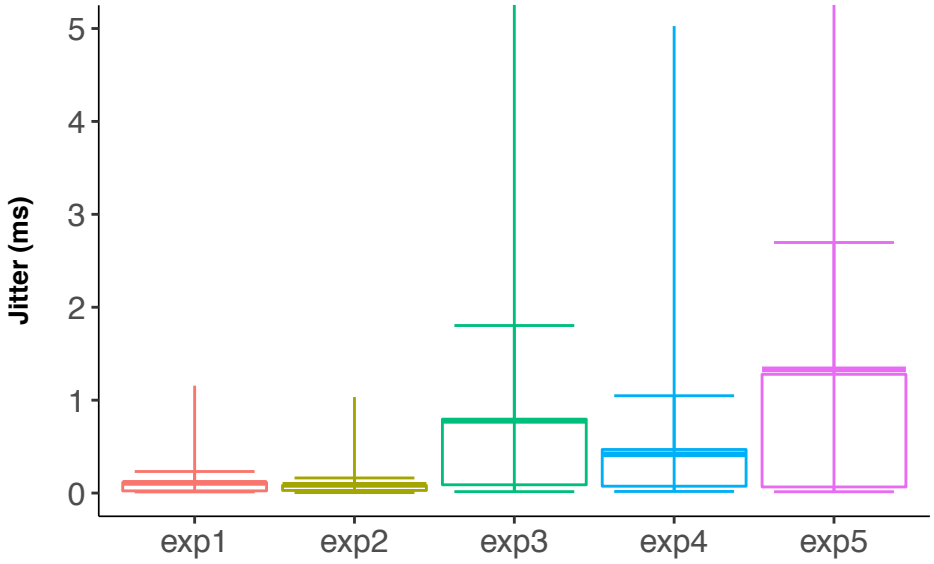


FIGURE 6.17: Jitter between Cape Town and Nairobi

TABLE 6.8: Packet loss between Cape Town and Nairobi

Loss (%)	Exp 3	Exp 4	Exp 5
Mean	1.724704	0.658173	2.787966
1st Quantile	0.005881	0.005464	0.024725
3rd Quantile	2.414113	0.840950	6.159645
Max.	9.904055	3.800906	8.502045

the experiments. Figure 6.19 shows the levels of loss for the subset of flows that had some packet loss.

In Experiment 5, a substantial portion of about 25 % of the lossy flows experienced packet loss of over 6 %. This higher packet loss can be attributed to a higher rate of packets arriving out of order and being discarded. This is particularly the case in Experiment 5 because the packet multiplexing employed therein did not consider path latencies, thereby increasing the likelihood of contiguous packets being forwarded through paths that have significant differences in delays. The path latency difference results in jitter, higher rate of out of order arrivals, and subsequent packet losses.

6.6 Discussion

The primary motivation for the proposed multipath solution was to deal with the problem of latencies that are caused by circuitous routing in the UbuntuNet topology. In this regard, the experiments conducted showed that the best (lowest) latencies were achieved through a single path packet forwarding mechanism, where the lowest latency path is selected for each flow. It could be argued therefore, that for applications that are not bandwidth intensive, but for which delay is crucial, it might be prudent to employ single path forwarding.

If the key traffic engineering motivation is to maximize pair-wise throughput between a pair of PoPs, then the multipath mechanisms evaluated would achieve better results than the single path mechanisms. In the experiments, the multipath packet forwarding achieved higher throughput but with additional performance costs, such as using higher latency paths. The multipath approaches achieved substantially better throughput, but at the same time, also registered higher latencies. This was particularly the case for multipath mechanisms that totally disregard path delays in favour of higher bandwidth paths, as was the case in Experiment 5. For instance, between Cape Town and Nairobi in the topology, Figures 6.14 and 6.15 show how the higher throughput paths also resulted in higher latencies. This would be expected in the UbuntuNet topology given that many higher capacity links are inter-continental. For applications that require high throughput and for which latency is not

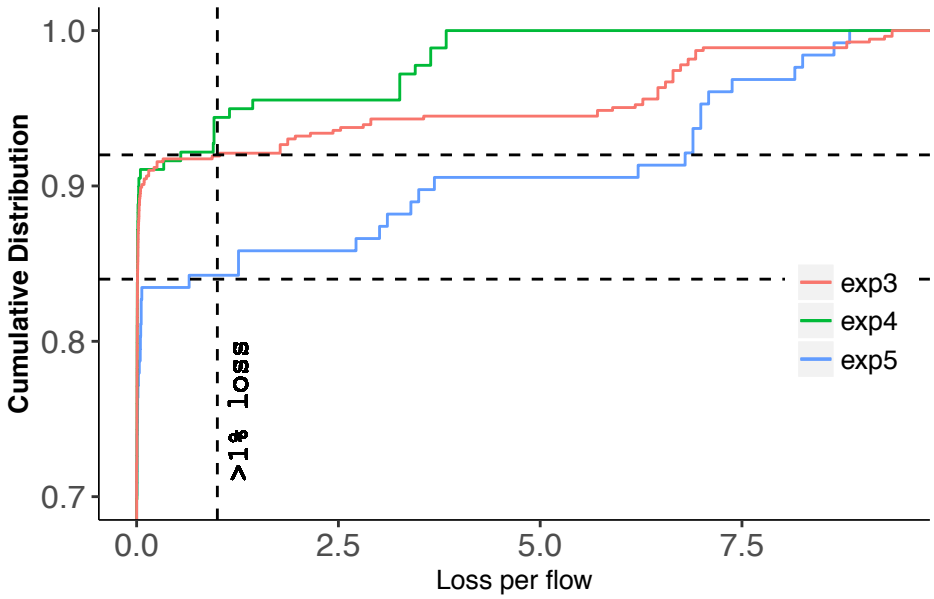


FIGURE 6.18: Loss between Cape Town and Nairobi

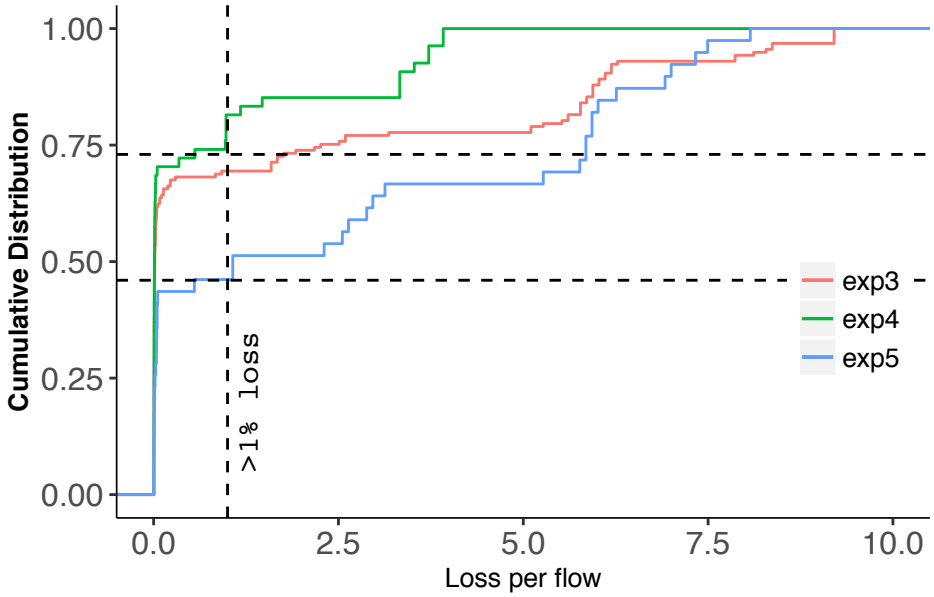


FIGURE 6.19: Lossy flows Cape Town and Nairobi

critical, it is more prudent therefore to employ a multipath forwarding mechanism in order to maximize bandwidth available in a number of paths through multiplexing.

Furthermore, while packet level multipath forwarding was able to increase throughput, it also introduced significant levels of jitter and packet loss. In the experiments, single path packet forwarding achieved the smoothest flows, registering almost zero jitter and packet loss. On the other hand, each of the multipath experiments experienced significant jitter and packet loss. For example, Experiment 5, which did not consider latency in reinforcement assignment, had both the highest jitter and packet loss. The best throughput in multipath setting was achieved when the primary factor for reinforcement rewards was the link's available bandwidth (Experiment 5). However, this configuration gave the worst performance in terms of latency, jitter and packet loss. Of the multipath configurations, the best performance in terms of latency and jitter was obtained when the rewards were given on the basis of both the available link capacity and latency (Experiment 3). On the other hand, single path forwarding is seen to provide the lowest jitter and packet loss. In terms of latency, the best performance is obtained with single path forwarding, where the rewards are based on the link delays.

In the multipath scheme, both jitter and packet loss appear to be minimized when latency is the primary key for ranking and selecting paths. When latency is the main attribute for multipath ranking, there is a higher chance that paths with similar delay paths are chosen for multiplexing, which ensures that the delay imbalances are reduced. In terms of throughput, this has the negative consequence of potentially selecting paths that do not have the best capacity, resulting in reduced overall throughput. In Experiment 3, a solution that reinforces paths based on both latency and capacity appears to reduce the prevalence of jitter and packet loss, while achieving throughput levels that are almost comparable to the other multipath experiments.

Overall, the results show how a scaled combination of the different network metrics in path rankings can help to achieve different multipath optimisation objectives. The evaluation shows how Reinforcement Learning can be used for multipath packet forwarding with the objective to achieve better throughput than standard single path forwarding. Also, dynamic ranking of paths based on latency (or capacity) is helpful in achieving lower latencies (or higher throughput). However, the multipath solution requires careful configuration of the packet block scheduling so as to increase the probability of contiguous packets following paths that are not too different in terms of delays. An important aspect of this configuration is the size of the packet blocks that get forwarded onto each of the multipath links at each hop: a very big block size diminishes the multiplexing and gravitates the solution towards single path forwarding; whereas a very small block size results in a very high rate of out-of-order arrivals and packet loss.

6.7 Summary

This chapter has presented a network architecture for performing multipath packet forwarding using a centralized controller, LISP gateways, and reinforcement learning agents in SDN switches. The strategy used reinforcement learning in an SDN core topology to enable performance-based adaptive routing. Reinforcement learning was used for hop-by-hop ranking of alternate links based on latency and available bandwidth. This was accomplished through continuous active and passive network measurements. A network controller was used to adaptively set up multiple routes through the switching nodes and used performance data to reinforce usage of better paths. Reinforcement learning capabilities within the SDN core topology enabled automatic updating of forwarding policies based on observed path performance, while SDN was used to reconfigure the packet forwarding rules and to enforce packet multiplexing. LISP was used by the NRENs to announce multiple gateways, and for traffic sources to dynamically and explicitly choose the destination's gateway to communicate with. The whole framework was managed by a centralized traffic engineering manager, embedded within an SDN controller. Evaluation of the mechanism was performed in an emulated topology of the UbuntuNet Alliance.

Results indicate that the mechanism was able to provide minimum latencies especially through single path latency-based packet forwarding. Multipath configurations of the mechanism achieved substantial increases in aggregate throughput compared to static single path packet forwarding. However, the multipath packet forwarding also had aggravated levels of jitter and packet loss. Furthermore, this chapter has shown the performance benefits for employing multipath packet forwarding without explicit intervention of or changing the operation of end hosts. In general, the multipath configuration was able to achieve significant throughput improvements without any modifications on the end hosts. It has further been shown how different types of QoS can be achieved by the use of SDN's dynamic path configurations.

Overall, the results in this chapter confirm that adaptive technologies, such as SDN and LISP, as well network-based adaptive learning, can indeed play an important role in improving Africa's inter-NREN data exchange.

Chapter 7

Conclusion

The main aim of this research was to consider the question of *"how African Research and Education Networks' logical topology can be improved to promote the exchange of knowledge and collaboration among research institutions in Africa"*. Driven by the research community's desire to collaborate and share computing resources across institutional boundaries, there has been global trend towards establishment of NRENs, which part from providing Internet access to the research institutions, also run software and systems that to allow scientific collaboration and provide researchers with global access to digital research resources. Many of NRENs have been interlinked into regional and global topologies. In Africa, regional associations of NRENs have formed regional networks: the UbuntuNet Alliance in Eastern and Southern Africa; and WACREN in Western and Central Africa. These inter-NREN topologies have made it possible for African researchers to participate in international collaborative research and to access the otherwise inaccessible digital resources, such as databases, super computers, telescopes, and electron microscopes. The inter-NREN topologies have also made it possible for researchers to experimentation testbeds, such as those supporting experiments in networking and other IT innovations. These collaborative research applications require low latency communication. However, despite the considerable increase in intra-Africa terrestrial fibre optic cables, a large portion of the inter-NREN traffic has been shown to be exchanged through circuitous routes traversing inter-continental links and Internet exchange points in Europe and North America, resulting in high end-to-end latencies. This necessitates traffic engineering techniques that would enable discovery and the use of low latency links across Africa's research networks. It is necessary to further optimize traffic exchange by enabling dynamic selection of routes based on path characteristics.

This research was motivated by circuitous routing and high end-to-end latencies between Africa's NRENs. To gain better understanding of the problem and to propose a potential solution, two main research phases were taken. The first phase involved undertaking topology analysis of the Pan-African NRENs to quantify the circuitous routing problem, as well as to devise and evaluate efficient mechanisms for probing and visualizing the NRENs topology. In the second phase, traffic engineering strategies were devised and evaluated to assess the possibility that NRENs would achieve better inter-NREN connectivity if they employed

flexible and dynamic route selection strategies supported by SDN, LISP and Reinforcement Learning. The thesis has proposed an NRENs traffic engineering framework to leverage the opportunities in SDN, LISP and Reinforcement Learning, and has shown how these technologies could enable NRENs to utilize network performance data to improve throughput and minimize latencies.

This chapter summarises findings from NRENs topology discovery exercises, as well as results of emulation-based traffic engineering experiments that were designed to evaluate the potential utility of implementing SDN and LISP in Africa's NRENs topology. The chapter also highlights the main contributions of this research.

7.1 Summary of Results and Contributions

This thesis makes contributions in two areas. The first contribution relates to the mapping of Africa's NRENs Internet topology and, secondly, pertaining to the design of LISP/SDN based traffic engineering for Africa's NRENs.

7.1.1 Topology Mapping

UbuntuNet Topology Maps

While previous studies have looked at the African Internet topology in general, this research was the first to specifically focus on Africa's inter-NREN topology and the UbuntuNet Alliance. This NRENs' topology is different from the general African Internet, in that the research and education institutions across the region have been working together to improve their interconnectivity through shared physical infrastructure, with a key goal of exchanging traffic more locally within the continent. It was important therefore to study how this cooperation has achieved the goal of keeping traffic local. This question was also aimed at understanding the performance impact of routing intra-Africa traffic through inter-continental. For this purpose, traceroute data was organised into intra-Africa and inter-continental traffic. Intra-Africa traffic is routed from African sources to African destinations and traverses only Africa based routers. On the other hand, inter-continental traffic is exchanged between African end points but traverses other continents.

In Chapter 4, this research has shown that a large portion of traffic is still exchanged through inter-continental routes, and that such inter-continental routing negatively impacts the end-to-end latencies between the NRENs. This research has also highlighted that the physical infrastructure alone is not enough for keeping the traffic local, but that appropriate traffic engineering mechanisms need to be put in place. This scenario negatively impacts the intra-Africa Internet performance, especially with regard to end-to-end latency, as well

as cost of traffic exchange. Presently, inter-NREN traffic engineering in the UbuntuNet Alliance is done through static routing. End-to-end paths are selected based on static link weights configured by network administrators. Given that static rules do not make it possible for the paths to be customized on runtime based on application QoS needs, it is a challenge for the inter-NREN topology to implement performance-based dynamic packet forwarding. Research has shown that there are usually better paths than the default paths (He and Rexford, 2008).

It is worth noting that the topology measurement exercise happened prior to the completion of AfricaConnect2 project in 2016. For this reason, it is likely that the amount of intra-Africa traffic that is actually exchanged through inter-continental links has decreased.

Mechanism for efficient topology discovery

The process of collecting topology data prompted further study on how to efficiently probe the UbuntuNet Alliance network. This was deemed necessary considering that distributed topology discovery platforms, such as Ripe Atlas and CAIDA's Archipelago, assign a cost and limit the number of probe packets one can transmit through the measurement platform. An efficient topology discovery campaign must be able to generate as few packets as possible while being able to discover the topology as completely as possible. On the other hand, to achieve reliable topology discovery, it is necessary to maximize the number of the target IP addresses that would be reached from the vantage points. There was also need to discover alternate, and potentially hidden, paths. To enhance the number of reached destinations and alternate paths, Paris-traceroute, which is capable of discovering load balancing paths was employed. Three probing protocols were used: TCP, ICMP, and UDP. The use of multiple protocols from each vantage point helped to discover more alternate paths, which also increases the completeness of the topology discovered (Section 4.1.4). The use of multiple vantage points ensured that all destinations were reached.

This research implemented and evaluated an efficient topology discovery tool that achieved significant reduction of the number of packets required to measurement the UbuntuNet topology (Chapter 4). More specifically, the mechanism, based on Sequential Topology Inference (Ni et al., 2010; Ni et al., 2008), achieved a 47% reduction in packets required to complete traceroute measurements when path overlaps are considered during measurements.

In terms of topology structure, the study found that at least 90% of the source-destination pairs had more than one end-to-end IP path, with an overall average path diversity of three (Section 4.2.1). This finding was important as it points to the potential for adaptive traffic engineering. The availability of multiple paths between NRENs entails that given the requisite knowledge about the alternate paths and access to the routing controller, better paths can be configured during run time.

Interactive topology visualization

Being multi-stakeholder network, the task of addressing the problem of circuitous routing between the African NRENs would require that all the members are able to perceive the logical topology and the performance implications of routing patterns. This would require topology information is presented to stakeholders in a widely compressible manner. Given the diversity of NREN stakeholders, including technical managers and non-technical policy makers, topology data needs to be presented in a manner that can be effectively perceived by all. One way of presenting topology information to a multi-stakeholder audience would be through visual maps. To test this idea, an interactive geo-spacial topology visualization tool was implemented. Evaluation of the visualization tool showed that users could accurately identify the logical paths.

This research implemented and evaluated an interactive topology visualization tool (Chapter 4) that can help NREN stakeholders to visualize traffic routes between NRENs, as well as to view different elements of the topology, including physical cables and IXPs. A user centred design approach was used, resulting in a visualisation tool that was able to show various dimensions of topology data on a map, including geolocation of IP hops, latencies, as well as location of IXPs and placement of terrestrial fibre. Unlike other traceroute visualisation tools, such as GTrace and Terrapix, which visualise either only a single traceroute measurement or display routes from a single vantage point, this tool allows multiple traceroutes sent from various vantage points to be viewed on the map. The evaluation looked at whether the geospatial visualisation designed could effectively and accurately communicate the network topology to allow users to identify routes. Effectiveness and accuracy were also evaluated by checking the correctness of answers to visual queries relating to physical and logical paths.

7.1.2 Traffic Engineering

An important consequence of the recent fibre optic cable deployments in the UbuntuNet inter-NREN topology is that there is now a substantial amount of path redundancy. Many NRENs are now connected to the Internet and the inter-NREN core topology through more than one gateway. The physical topology of the UbuntuNet has multiple alternate paths between many of the member NRENs. The multi-homing provides a number of advantages, including increased end-to-end path diversity. The results of topology mapping carried out by this research also confirmed the presence of multiple paths between NRENs. Such path diversity increases the potential for traffic engineering, which can lead to performance enhancement through selection of paths based on performance. Multipath routing can result in cost reductions, such as by using cheaper links for ordinary traffic and expensive links for critical or delay sensitive data. Furthermore, recent developments in Internet routing

and switching technology, such as SDN and LISP, offer Pan-African NRENs a chance to implement optimal traffic engineering solutions for NRENs. SDN allows creation of adaptive Internet traffic engineering schemes. These factors provide potential for the UbuntuNet Alliance to improve its logical inter-NRENs topology through traffic engineering based on LISP, SDN and Reinforcement Learning.

After studying the topology of the UbuntuNet NRENs, and having noted the extent and performance impact of circuitous routing for Internet traffic exchanged between the NRENs, as well as the potential provided by the topology's multipath environment, this research sought to test potential improvements through traffic engineering. Traffic engineering frameworks were therefore proposed to study how to leverage the multipath environment of the UbuntuNet Alliance. The research objective of traffic engineering experiments was to study the extent to which performance-based adaptive routing could help reduce latencies and increase throughput between UbuntuNet NRENs.

The second contribution of this thesis is the design of a traffic engineering framework based on SDN, LISP and Reinforcement Learning, highlighting how they can be used together to enable dynamic packet forwarding in a multipath NRENs. Whereas benefits of SDN and LISP have been the subject of research for a while, the extent to which they can be used **together** to reduce latencies and maximize throughput among Africa's NRENs has not been explored. In this study, LISP was used for announcement and discovery of multiple NREN gateways, while SDN was used for dynamic configuration of end-to-end paths between NRENs. Reinforcement Learning (Q-learning Algorithm) was used to maintain a knowledge-base of paths' performance and to adapt forwarding rules towards using better paths. Two traffic engineering mechanisms were evaluated, the first employing only SDN and LISP, while the second used SDN, LISP, and Reinforcement Learning.

Latency-based adaptive LISP/SDN framework

In terms of traffic engineering, the first contribution was the design of a traffic engineering mechanism that can be implemented at the gateways of each NREN. The approach was based on the use of LISP in the NRENs' gateways and would allow the NRENs to have some level of path control by being able to choose the specific source and destination gateways through which to exchange traffic. The mechanism, presented in Chapter 5, incorporates the capabilities of SDN and LISP to allow NRENs to announce multiple gateways, and to dynamically select and configure end-to-end paths. A latency measurement tool was implemented with the LISP gateways to allow dynamic ranking of destination gateways, and an SDN module allowed runtime configuration of end-to-end paths. This enabled interconnected NRENs to exchange traffic through the lowest latency gateways. Existing LISP-based path ranking mechanisms are designed to run in a service provider's network as a client-server application, providing ranked paths to edge networks. In contrast, the proposed LISP

ranking in this thesis is implemented at the edge of each network. Furthermore, the mechanism proposed in this thesis uniquely incorporated an SDN module that interacts with an OpenFlow network controller, making it possible for edge networks to dynamically request configuration of shortest paths between the source and destination gateways.

Using this approach, an NREN may be able to regulate the amount or type of traffic that it transmits through different interconnection points and would be able to determine the gateways through which their traffic is sent or received. However, such a solution is deficient in that it is unable to support dynamic traffic control in the inter-NREN core topology. On the other hand, a purely centralized SDN-based solution, where all the routing decisions are made by the network controller, would mean that the individual NRENs have no control on their traffic's end-to-end paths. Instead, an SDN controller would be responsible for determining the routes between a pair of NRENs. In practice however, NRENs may want to exercise some control on the paths used for the traffic exchange. It is for this reason that LISP was added to the framework to afford the NRENs the opportunity to select the source and destination gateways, while the routing inside the core topology is left to the SDN controller.

Reinforcement Learning in SDN topology

The second traffic engineering contribution was the implementation of reinforcement learning (Q-learning algorithm) for adaptive multipath packet forwarding in SDN, and evaluation of the Q-learning implementation in an emulated SDN/LISP based UbuntuNet topology. The traffic engineering mechanism, described in Chapter 6, allows the topology to track performance (latency) and utilization of its links, and adaptively directs traffic with the objective of achieving either lowest end-to-end latency, or maximum throughput. An emulated topology of the UbuntuNet was implemented in Mininet, and its evaluation provides insight into how performance of the inter-NREN traffic exchange could be improved through SDN and Reinforcement Learning.

While the SDN/LISP traffic engineering solution makes it possible for NRENs to announce multiple gateways, as well as to achieve dynamic route selection based on end-to-end performance, the inclusion of Reinforcement Learning in the framework lets the core network decide on how traffic flows inside the core topology. The integration of LISP, SDN and Reinforcement Learning in the interconnection points would allow NRENs to achieve flexible traffic engineering strategies that utilize paths that provide either the lowest latencies or highest throughput, and be able to dynamically respond to varying network conditions. This could also be useful for helping NRENs to reduce usage of inter-continental Internet paths. The solution would also allow NRENs to employ application specific traffic engineering. For example, delay sensitive traffic between NRENs would be channelled via low latency intra-Africa links, whereas bandwidth intensive flows would be routed through high capacity inter-continental links. Ultimately, the solutions would be useful in helping

to reduce Internet costs and improve performance. The solution could also lead to a flexible Internet peering environment that would support better regional research collaboration.

Together, the SDN/LISP path ranking mechanism presented in Chapter 5, and the SDN/LISP with Reinforcement Learning approach in Chapter 6, do suggest that, at low network congestion levels, the ranking and dynamic configuration of end-to-end paths can help to achieve low latencies. However, the advantage of path ranking appears to diminish as the congestion in the topology increases. This is because as at high congestion levels, latencies are already compromised such that the path ranking mechanism itself is affected and, the impact of adaptive traffic engineering using LISP and SDN does not achieve the intended performance gains. On the other hand, implementation of reinforcement learning and multipath forwarding inside the core topology does help to optimise throughput even when network congestion is high. Even when some of the links/paths between a pair of source and destination hosts are congested, reinforcement learning and multipath packet forwarding ensures that throughput is sustained by re-distributing traffic onto the multiple possible paths.

7.2 Further Research

7.2.1 Topology Discovery

In this research, topology discovery and topology visualization were implemented as separate systems, such that traceroute data had to be manually imported into the visualization tool. Future work should aim to integrate the two systems, and this mean that topology data is automatically available for visualization almost immediately after measurements are run, ensuring that the visualization renders the most up to date information. The integration would also allow for new topology data to be collected, stored, aggregated and displayed in the graphical visualisation, and this would potentially help researchers, network managers and other interested parties in planning new routing policies based on an accurate and up-to-date set of data.

One challenge with NRENs topology discovery in the UbuntuNet Alliance was the limited number of vantage points available inside the topology. An ideal scenario would be to have at least one vantage point inside each and every NREN, or even better, having a vantage point in each and every campus network that is part of the NRENs topology. This would lead to a richer topology dataset that would reveal more detail about the NRENs interconnections and performance. To improve topology discovery and performance monitoring across the UbuntuNet, there has to be a deliberate effort to deploy more Internet measurement vantage points in all the member NRENs. Furthermore, while usability tests

on the topology visualization tool revealed that the system was usable, a longer evaluation would need to be done with managers from more NRENs to determine whether the visualization actually supports decision making, such as with regard to the placement of interconnection points. A longer visualization study would have to work with an integrated topology discovery and visualization system so that users would access continuously updated topology information.

7.2.2 Multipath Traffic Engineering Design

The multipath mechanism employed in this thesis (Chapter 6) was noted to have introduced high levels of jitter and packet loss. Such levels of jitter and packet loss are as a result of a significant levels of out-of-order packet arrivals in multipath packet-level forwarding (Kandula et al., 2007). In packet-level multiplexing, delay variance between the multiple paths results in out-of-order arrivals and packet loss. The out-of-order arrivals are aggravated by significant delay imbalances across the alternate paths. This is particularly problematic for the UbuntuNet, considering that the topology has high variance in path delays between the alternate paths, especially between the intra-Africa and inter-continental paths.

This research employed a packet resequencing mechanism inside the network before delivering the packets to the end nodes. However, further work is required to determine better ways of balancing traffic onto multiple paths, especially when there are significant delay imbalances between the paths. One approach would be to set the appropriate size of the packet blocks based on the imbalances between the paths. Ideally, if the latency difference between the multiple paths is significant, then it is helpful to use bigger blocks of packets (number of packets forwarded to each path based on the Q-value). Splitting traffic in large packet blocks would mean that, for prolonged periods, packets are forwarded towards only one of the paths, leaving the other paths idle. This could lead to a situation similar to that of single path forwarding, where all the packets belonging to the same flow follow a single path. Conversely, smaller packet blocks would mean that there is less idle time in all the alternate paths as packets are forwarded to all the paths almost concurrently. Having smaller packet blocks also means that more contiguous packets would take different paths. However, if the multipaths have very similar end-to-end delays, there is a lower probability that the contiguous packets that go onto the different paths would arrive out of order, and in that case that smaller packet blocks would have a lesser negative impact in terms of out-of-order arrivals, while at the same time providing better bandwidth utilization.

An ideal solution would therefore need to be able to dynamically set the optimal packet block sizes, by considering the delay imbalance between a set of multipaths. In this regard, further research will need to evaluate mechanisms for setting the optimal packet block sizes

so as to maximize usage of the multipaths while minimizing out of order arrivals, jitter and packet loss.

While the Q-learning approach employed in this thesis makes the forwarding decisions on a hop by hop basis, future research could also evaluate how the mechanism for and the utility of applying Reinforcement Learning on and end-to-end basis, i.e on complete paths. Similarly, instead of applying Q-learning decisions on each and every hop, a hybrid mechanism would apply the learning mechanism on sets of hops, forming sub-paths. Such an approach could, for example, be designed to work with pathlet routing mechanisms, in which networks advertise fragments of paths that can be concatenated by source networks to form end-to-end paths (Godfrey et al., 2009). Also, instead of the topology having to employ only one of the possible dynamic routing mechanisms, future work may investigate how a topology may dynamically adapt the traffic engineering mechanisms based on network conditions or QoS requirements. In this regard, future work will have to include further experimentations on how the learning mechanism can adapt to dynamic network environments that may change more rapidly, including on per second basis. The emulation based experiments will also have to be augmented with deployments in real testbeds.

7.3 Concluding Remarks

This thesis proposes that for the UbuntuNet Alliance to achieve optimal inter-NREN routing, in terms of reducing latencies and optimizing bandwidth, the Alliance needs a more flexible and dynamic traffic engineering environment. A flexible environment needs to allow the NRENs to exploit the opportunities of the multipath UbuntuNet Alliance topology by being able to use knowledge of the interconnectivity to dynamically set optimal interdomain paths. A collaborative inter-NREN traffic engineering framework could enable optimal routing among the NRENs through the sharing of multiple gateways, and dynamically choosing end-to-end paths based on network path conditions. The design of the traffic engineering frameworks was guided by the notion that for NRENS to achieve a flexible and dynamic routing environment, they need to be able to do at least the following three things: firstly, NRENs need to be able to discover the alternate inter-NREN paths; once the alternate paths are known, NRENs need to be able to determine, on a continuous basis, the prevailing performance of the paths; and lastly, the NRENs need the capability to reconfigure alternate end-to-end paths based on quality of the paths. To achieve these three requirements, this thesis looked at how SDN, LISP and Reinforcement Learning could be used together in the inter-NREN topology as a mechanism for dynamic discovery and configuration of alternate paths. While the traffic engineering frameworks in this thesis have been described

and evaluated in terms of the UbuntuNet Alliance topology, the approach could also be applicable for other types of federated networks, especially where there is collaborative traffic management and sharing of network resources.

Bibliography

Abras, Chadia, Diane Maloney-Krichmar, and Jenny Preece (2004). "User-centered design". In: *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications* 37.4, pp. 445–456.

Afergan, Mike and John Wroclawski (2004). "On the benefits and feasibility of incentive based routing infrastructure". In: *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*. ACM, pp. 197–204.

Agarwal, Sugam, Murali Kodialam, and TV Lakshman (2013). "Traffic engineering in software defined networks". In: *INFOCOM, 2013 Proceedings IEEE*. IEEE, pp. 2211–2219.

Ager, Bernhard et al. (2012). "Anatomy of a large european IXP". In: *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, pp. 163–174.

Ahmad, M.Z. and R. Guha (2011). "Internet exchange points and Internet routing". In: *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, pp. 292–294. DOI: 10.1109/ICNP.2011.6089065.

— (2012). "A tale of nine Internet exchange points: Studying path latencies through major regional IXPs". In: *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, pp. 618–625. DOI: 10.1109/LCN.2012.6423683.

Akyildiz, Ian F. et al. (2014). "A roadmap for traffic engineering in SDN-OpenFlow networks". In: *Computer Networks* 71.0, pp. 1 –30. ISSN: 1389-1286. DOI: http://dx.doi.org/10.1016/j.comnet.2014.06.002. URL: http://www.sciencedirect.com/science/article/pii/S1389128614002254.

Al-Fares, Mohammad, Alexander Loukissas, and Amin Vahdat (2008). "A Scalable, Commodity Data Center Network Architecture". In: *SIGCOMM Comput. Commun. Rev.* 38.4, pp. 63–74. ISSN: 0146-4833. DOI: 10.1145/1402946.1402967. URL: http://doi.acm.org/10.1145/1402946.1402967.

Al-Fares, Mohammad et al. (2010). "Hedera: Dynamic Flow Scheduling for Data Center Networks." In: *NSDI*. Vol. 10, pp. 19–19.

Allalouf, M., E. Kaplan, and Y. Shavitt (2009). "On the feasibility of a large scale distributed testbed for measuring quality of path characteristics in the Internet". In: *Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, pp. 1–6. DOI: 10.1109/TRIDENTCOM.2009.4976195.

Alliance, UbuntuNet (2014). *UbuntuNet, the regional high-speed Internet network*. URL: http://www.geant.net/MediaCentreEvents/news/Pages/UbuntuNet-commissioned.aspx (visited on 07/14/2015).

Altman, Eitan and Hisao Kameda (2005). "Equilibria for multiclass routing problems in multi-agent networks". In: *Advances in Dynamic Games*. Springer, pp. 343–367.

Altman, Eitan et al. (2006). "A survey on networking games in telecommunications". In: *Computers & Operations Research* 33.2, pp. 286–311.

Andronico, Giuseppe et al. (2011). "e-Infrastructures for e-Science: a global view". In: *Journal of Grid Computing* 9.2, pp. 155–184.

Araújo, Joao Taveira et al. (2014). "Software-defined network support for transport resilience". In: *Performance Evaluation* 67, p. 5.

Atlas, RIPE (2015). "A Global Internet Measurement Network". In: *Internet Protocol Journal*.

Augustin, B., T. Friedman, and R. Teixeira (2007a). "Multipath tracing with Paris traceroute". In: *End-to-End Monitoring Techniques and Services, 2007. E2EMON '07. Workshop on*, pp. 1–8. DOI: 10.1109/E2EMON.2007.375313.

Augustin, Brice, Timur Friedman, and Renata Teixeira (2007b). "Measuring load-balanced paths in the Internet". In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, pp. 149–160.

— (2011). "Measuring multipath routing in the internet". In: *IEEE/ACM Transactions on Networking (TON)* 19.3, pp. 830–840.

Avallone, Stefano et al. (2004). "D-ITG distributed internet traffic generator". In: *Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings. First International Conference on the*. IEEE, pp. 316–317.

Awduche, Daniel O, Johnson Agogbua, and Jim McManus (1998). "An approach to optimal peering between autonomous systems in the Internet". In: *Computer Communications and Networks, 1998. Proceedings. 7th International Conference on*. IEEE, pp. 346–351.

Badger, Merete et al. (2013). "Virtual Campus Hub: A single sign-on system for cross-border collaboration". In: *EDULEARN13 Proceedings*, pp. 5759–5759.

Banda, Tiwonge Msulira, Bjorn Pehrson, and Mike Jensen (2007). "Ubuntunet Alliance: Zomba Strategic Plan for the Period 2007 To 2010". In:

Banfi, Dario et al. (2016). "Endpoint-transparent Multipath Transport with Software-defined Networks". In: *Local Computer Networks (LCN), 2016 IEEE 41st Conference on*. IEEE, pp. 307–315.

Barry, Boubakar (2013). "AAU Research and Education Networking Unit (RENU)–Phase 2". In:

Barry, Boubakar et al. (2010). "eGY-Africa: better Internet connectivity to reduce the digital divide". In: *IST-Africa, 2010*. IEEE, pp. 1–15.

Becker, Richard A., Stephen G. Eick, and Allan R. Wilks (1995). "Visualizing network data". In: *IEEE Transactions on visualization and computer graphics* 1.1, pp. 16–28.

Benson, Theophilus, Aditya Akella, and David A Maltz (2010). "Network traffic characteristics of data centers in the wild". In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, pp. 267–280.

Berman, Mark et al. (2014). "GENI: A federated testbed for innovative network experiments". In: *Computer Networks* 61, pp. 5–23.

Bhatia, Manav (2003). *Advertising Equal Cost Multi-Path (ECMP) routes in BGP*. Tech. rep. Internet draft,< draft-bhatia-ecmp-routes-in-bgp-00. txt>, work in progress (expired).

Bonacich, Phillip (2007). "Some unique properties of eigenvector centrality". In: *Social networks* 29.4, pp. 555–564.

Botta, Alessio, Alberto Dainotti, and Antonio Pescapé (2012). "A tool for the generation of realistic network workload for emerging networking scenarios". In: *Computer Networks* 56.15, pp. 3531 –3547. ISSN: 1389-1286. DOI: http://dx.doi.org/10.1016/j.comnet.2012.02.019. URL: http://www.sciencedirect.com/science/article/pii/S1389128612000928.

Boyan, Justin A and Michael L Littman (1994). "Packet routing in dynamically changing networks: A reinforcement learning approach". In: *Advances in neural information processing systems*, pp. 671–671.

Branigan, Steve et al. (2001). "What can you do with traceroute?" In: *Internet Computing, IEEE* 5.5, p. 96.

Braun, Wolfgang and Michael Menth (2015). "Load-dependent flow splitting for traffic engineering in resilient OpenFlow networks". In: *Networked Systems (NetSys), 2015 International Conference and Workshops on*. IEEE, pp. 1–5.

Brooke, John et al. (1996). "SUS-A quick and dirty usability scale". In: *Usability evaluation in industry* 189.194, pp. 4–7.

Brownlee, Nevil (2005). "Some observations of Internet stream lifetimes". In: *International Workshop on Passive and Active Network Measurement*. Springer, pp. 265–277.

Busoniu, Lucian, Robert Babuska, and Bart De Schutter (2008). "A comprehensive survey of multiagent reinforcement learning". In: *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews* 38.2, p. 156.

Cabellos, Albert et al. (2011). "LISPmob: Mobile Networking through LISP". In: *LISPmob white paper*.

Caesar, Matthew et al. (2005). "Design and implementation of a routing control platform". In: *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, pp. 15–28.

Camacho, Jose M. et al. (2013). "BGP-XM: {BGP} eXtended Multipath for transit Autonomous Systems". In: *Computer Networks* 57.4, pp. 954 –975. ISSN: 1389-1286. DOI: http://dx.

doi.org/10.1016/j.comnet.2012.11.011. URL: http://www.sciencedirect.com/science/article/pii/S1389128612003957.

Campista, Miguel Elias M et al. (2014). "Challenges and research directions for the future internetworking". In: *IEEE Communications Surveys & Tutorials* 16.2, pp. 1050–1079.

Carlos, AB Macapuna, Christian Esteve Rothenberg, and F Magalhães Maurício (2010). "In-packet Bloom filter based data center networking with distributed OpenFlow controllers". In: *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*. IEEE, pp. 584–588.

Carr, David A (1999). "Guidelines for designing information visualization applications". In: *Proceedings of ECUE 99*, pp. 1–3.

Chatzis, Nikolaos et al. (2013). "There is more to IXPs than meets the eye". In: *ACM SIGCOMM Computer Communication Review* 43.5, pp. 19–28.

Chetty, Marshini et al. (2013). "Measuring broadband performance in South Africa". In: *Proceedings of the 4th Annual Symposium on Computing for Development*. ACM, p. 1.

Chiesa, Marco, Guy Kindler, and Michael Schapira (2014). "Traffic engineering with equal-cost-multipath: An algorithmic perspective". In: *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, pp. 1590–1598.

Choi, Samuel PM and Dit-Yan Yeung (1996). "Predictive Q-routing: A memory-based reinforcement learning approach to adaptive traffic control". In: *Advances in Neural Information Processing Systems*, pp. 945–951.

Chun, Wu et al. (2014). "Multiagent Reinforcement Learning Dynamic Spectrum Access in Cognitive Radios". In: *Sensors & Transducers* 164.2, p. 170.

Csardi, Gabor and Tamas Nepusz (2006). "The igraph software package for complex network research". In: *InterJournal, Complex Systems* 1695.5, pp. 1–9.

De Vito, Luca, Sergio Rapuano, and Laura Tomaciello (2008). "One-way delay measurement: State of the art". In: *IEEE Transactions on Instrumentation and Measurement* 57.12, pp. 2742–2750.

DeNardis, Dr et al. (2012). "Governance at the Internet's Core: The Geopolitics of Interconnection and Internet Exchange Points (IXPs) in Emerging Markets". In: *Governance at the Internet's Core: The Geopolitics of Interconnection and Internet Exchange Points (IXPs) in Emerging Markets (March 27, 2012)*.

Desai, Rahul and BP Patil (2015). "Cooperative reinforcement learning approach for routing in ad hoc networks". In: *Pervasive Computing (ICPC), 2015 International Conference on*. IEEE, pp. 1–5.

Donnet, B. and T. Friedman (2007). "Internet topology discovery: a survey". In: *Communications Surveys Tutorials, IEEE* 9.4, pp. 56–69. ISSN: 1553-877X. DOI: 10.1109/COMST.2007.4444750.

Edmundson, Anne et al. (2016). "A First Look into Transnational Routing Detours". In: *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*. ACM, pp. 567–568.

Egilmez, Hilmi E et al. (2012). "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks". In: *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*. IEEE, pp. 1–8.

Eriksson, Brian et al. (2010). "Toward the practical use of network tomography for internet topology discovery". In: *INFOCOM, 2010 Proceedings IEEE*. IEEE, pp. 1–9.

— (2012). "Efficient network tomography for internet topology discovery". In: *IEEE/ACM Transactions on Networking (TON)* 20.3, pp. 931–943.

Escalona, E. et al. (2013). "Using SDN for Cloud Services Provisioning: The XIFI Use-Case". In: *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, pp. 1–7. DOI: 10.1109/SDN4FNS.2013.6702561.

Evans, E, ROGER Burritt, and JAMES Guthrie (2013). "The Virtual University: Impact on Australian Accounting and Business Education". In:

Fanou, Rodérick, Pierre Francois, and Emile Aben (2015). "On the diversity of interdomain routing in africa". In: *International Conference on Passive and Active Network Measurement*. Springer, pp. 41–54.

Fanou, Rodérick et al. (2016). "Pushing the frontier: Exploring the African web ecosystem". In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 435–445.

Feamster, Nick, Jay Borkenhagen, and Jennifer Rexford (2003). "Guidelines for interdomain traffic engineering". In: *ACM SIGCOMM Computer Communication Review* 33.5, pp. 19–30.

Feigenbaum, Joan et al. (2005). "A BGP-based mechanism for lowest-cost routing". In: *Distributed Computing* 18.1, pp. 61–72.

Foley, Michael (2016). "The Role and Status of National Research and Education Networks in Africa". In:

Fortz, Bernard and Mikkel Thorup (2000). "Internet traffic engineering by optimizing OSPF weights". In: *INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*. Vol. 2. IEEE, pp. 519–528.

Freitas, Carla MDS et al. (2002). "On evaluating information visualization techniques". In: *Proceedings of the working conference on Advanced Visual Interfaces*. ACM, pp. 373–374.

Fryer, Thomas et al. (2014). "Research and Education Networks around the World and their Use". In:

Galperin, Hernán (2013). "Connectivity in Latin America and the Caribbean: The role of Internet Exchange Points". In:

Gilmore, JS, NF Huysamen, and AE Krzesinski (2007). "Mapping the African Internet". In: *Proceedings Southern African Telecommunication Networks and Applications Conference (SATNAC),(Sept 2007), Mauritius*.

Godfrey, P et al. (2009). "Pathlet routing". In: *ACM SIGCOMM Computer Communication Review* 39.4, pp. 111–122.

Goldenberg, David K et al. (2004). "Optimizing cost and performance for multihoming". In: *ACM SIGCOMM Computer Communication Review*. Vol. 34. 4. ACM, pp. 79–92.

Greene, Jonathan et al. (1991). "Segmented channel routing". In: *Proceedings of the 27th ACM/IEEE Design Automation Conference*. ACM, pp. 567–572.

Guestrin, Carlos, Michail Lagoudakis, and Ronald Parr (2002). "Coordinated reinforcement learning". In: *ICML*. Vol. 2, pp. 227–234.

Gupta, Arpit et al. (2013). "SDX: A Software Defined Internet Exchange". In:

Gupta, Arpit et al. (2014). "Peering at the Internet's Frontier". In: *Passive and Active Measurement Conference 2014*.

Haeri, Soroush, Majid Arianezhad, and Ljiljana Trajkovic (2013). "A Predictive Q-Learning Algorithm for Deflection Routing in Buffer-Less Networks". In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, pp. 764–769.

Haeri, Soroush et al. (2013). "A reinforcement learning-based algorithm for deflection routing in optical burst-switched networks". In: *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*. IEEE, pp. 474–481.

Handigol, Nikhil et al. (2009). "Plug-n-Serve: Load-balancing web traffic using OpenFlow". In: *ACM Sigcomm Demo* 4.5, p. 6.

Handigol, Nikhil et al. (2011). *Aster* x: Load-balancing web traffic over wide-area networks*.

He, Jiayue and Jennifer Rexford (2008). "Toward internet-wide multipath routing". In: *Network, IEEE* 22.2, pp. 16–21.

Heller, Brandon et al. (2010). "ElasticTree: Saving Energy in Data Center Networks." In: *Nsdi*. Vol. 10, pp. 249–264.

Heller, Brandon et al. (2012). "Reproducible Network Experiments using Container Based Emulation". In: *Proc. ACM CoNEXT*.

Hong, Chi-Yao et al. (2013). "Achieving High Utilization with Software-driven WAN". In: *SIGCOMM Comput. Commun. Rev.* 43.4, pp. 15–26. ISSN: 0146-4833. DOI: 10.1145/2534169.2486012. URL: http://doi.acm.org/10.1145/2534169.2486012.

House, Packet Clearing (2014). *Internet exchange directory*.

Hyun, Young (2006). "Archipelago measurement infrastructure". In: *Proceedings of the 7th CAIDA-WIDE Workshop*.

Izumi, S. et al. (2015). "An Adaptive Multipath Routing Scheme Based on SDN for Disaster-Resistant Storage Systems". In: *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, pp. 478–483. DOI: 10.1109/BWCCA.2015.73.

Jacob, Philip and Bruce Davie (2005). "Technical challenges in the delivery of interprovider QoS". In: *IEEE Communications Magazine* 43.6, pp. 112–118.

Jain, Ankur and Joseph Pasquale (2012). "Internet Distance Prediction Using Node-Pair Geography". In: *Network Computing and Applications (NCA), 2012 11th IEEE International Symposium on*. IEEE, pp. 71–78.

Jain, Sushant et al. (2013). "B4: Experience with a Globally-deployed Software Defined Wan". In: *SIGCOMM Comput. Commun. Rev.* 43.4, pp. 3–14. ISSN: 0146-4833. DOI: 10.1145/2534169.2486019. URL: http://doi.acm.org/10.1145/2534169.2486019.

Janz, Robert et al. (2016). "Building the Digital Silk Road: Charting the Development of Academic Collaborations between Europe and Central Asia". In: *Bildung und Erziehung* 69.1, pp. 11–40.

Jemec, Miha (2012). *packETH–Ethernet packet generator*.

Jeong, Kwangtae, Jinwook Kim, and Young-Tak Kim (2012). "QoS-aware network operating system for software defined networking with generalized OpenFlows". In: *Network Operations and Management Symposium (NOMS), 2012 IEEE*. IEEE, pp. 1167–1174.

Jo, Ju-Yeon et al. (2002). "Internet traffic load balancing using dynamic hashing with flow volume". In: *ITCom 2002: The Convergence of Information Technologies and Communications*. International Society for Optics and Photonics, pp. 154–165.

Johari, Ramesh and John N Tsitsiklis (2004). "Routing and peering in a competitive Internet". In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 2. IEEE, pp. 1556–1561.

Kandula, Srikanth et al. (2007). "Dynamic load balancing without packet reordering". In: *ACM SIGCOMM Computer Communication Review* 37.2, pp. 51–62.

Keim, Daniel A (2002). "Information visualization and visual data mining". In: *IEEE transactions on Visualization and Computer Graphics* 8.1, pp. 1–8.

Kende, M and C Hurpy (2012). "Assessment of the impact of Internet Exchange Points (IXPs)-empirical study of Kenya and Nigeria". In: *Internet Society (ISOC)*.

Kolahi, Samad S. et al. (2011). "Performance Monitoring of Various Network Traffic Generators". In: *International Conference on Computer Modeling and Simulation*. DOI: 10.1109/UKSIM.2011.102.

Koua, Etien L and M-J Kraak (2004). "A usability framework for the design and evaluation of an exploratory geovisualization environment". In: *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*. IEEE, pp. 153–158.

Lam, Heidi et al. (2011). "Seven guiding scenarios for information visualization evaluation". In:

Landa, Raúl et al. (2013). "The large-scale geography of Internet round trip times." In: *Networking*, pp. 1–9.

Lara, A., A. Kolasani, and B. Ramamurthy (2014). "Network Innovation using OpenFlow: A Survey". In: *Communications Surveys Tutorials, IEEE* 16.1, pp. 493–512. ISSN: 1553-877X. DOI: 10.1109/SURV.2013.081313.00105.

Lee, Eun Kyung, Hariharasudhan Viswanathan, and Dario Pompili (2016). "RescueNet: Reinforcement-learning-based communication framework for emergency networking". In: *Computer Networks* 98, pp. 14–28.

Lee, Sanghwan, Zhi-Li Zhang, and Srihari Nelakuditi (2004). "Exploiting as hierarchy for scalable route selection in multi-homed stub networks". In: *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, pp. 294–299.

Leung, Ka-Cheong, Victor OK Li, and Daiqin Yang (2007). "An overview of packet reordering in transmission control protocol (TCP): problems, solutions, and challenges". In: *IEEE transactions on parallel and distributed systems* 18.4.

Li, Ke, Sheng Wang, and Xiong Wang (2011). "Edge router selection and traffic engineering in LISP-capable networks". In: *Communications and Networks, Journal of* 13.6, pp. 612–620. ISSN: 1229-2370. DOI: 10.1109/JCN.2011.6157477.

Li, Yu and Deng Pan (2013). "OpenFlow based load balancing for Fat-Tree networks with multipath support". In: *Proc. 12th IEEE International Conference on Communications (ICC'13), Budapest, Hungary*, pp. 1–5.

Li, Zhonghui et al. (2010). "Study on partial-transit Inter-domain routing over the TEIN2 backbone". In: *The 19th Annual Wireless and Optical Communications Conference (WOCC 2010)*. IEEE, pp. 1–6.

Liu, Kunpeng, Bijan Jabbari, and Stefano Secci (2013). "Generalized multipath load sharing using vectorized routing model". In: *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, pp. 1507–1512.

Luckie, Matthew (2010). "Scamper: a scalable and extensible packet prober for active measurement of the internet". In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, pp. 239–245.

Luckie, Matthew, Young Hyun, and Bradley Huffaker (2008). "Traceroute probe method and forward IP path inference". In: *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. ACM, pp. 311–324.

Machiraju, Sridhar and Randy H Katz (2004). "Verifying global invariants in multi-provider distributed systems". In: *Proc. SIGCOMM Workshop on Hot Topics in Networking (HotNets)*, pp. 149–154.

Madhyastha, Harsha V et al. (2006). "iPlane: An information plane for distributed services". In: *Proceedings of the 7th symposium on Operating systems design and implementation*. USENIX Association, pp. 367–380.

Mahajan, Ratul, David Wetherall, and Thomas Anderson (2004). "Towards coordinated interdomain traffic engineering". In: *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*.

Mahajan, Ratul, David Wetherall, and Thomas Anderson (2005). "Negotiation-based routing between neighboring ISPs". In: *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, pp. 29–42.

Mahajan, Ratul, David Wetherall, and Thomas E Anderson (2007). "Mutually Controlled Routing with Independent ISPs". In: *NSDI*.

Mannie, Eric (2004). "Generalized multi-protocol label switching (GMPLS) architecture". In: *Interface* 501, p. 19.

Mao, Zhuoqing Morley et al. (2003). "Towards an accurate AS-level traceroute tool". In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, pp. 365–378.

Matthews, Warren and Les Cottrell (2000). "The PingER project: active Internet performance monitoring for the HENP community". In: *IEEE Communications Magazine* 38.5, pp. 130–136.

McCreary, Sean and KC Claffy (2000). *Trends in wide area IP traffic patterns*.

Mendiola, A. et al. (2016a). "Multi-domain bandwidth on demand service provisioning using SDN". In: *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 353–354. DOI: 10.1109/NETSOFT.2016.7502407.

Mendiola, Alaitz et al. (2016b). "A survey on the contributions of Software-Defined Networking to Traffic Engineering". In: *IEEE Communications Surveys & Tutorials*.

Mérindol, Pascal et al. (2009). "Quantifying ASes multiconnectivity using multicast information". In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, pp. 370–376.

Motamedi, Reza, Reza Rejaie, and Walter Willinger (2015). "A Survey of Techniques for Internet Topology Discovery". In: *IEEE Communications Surveys & Tutorials* 17.2, pp. 1044–1065.

Ni, J. et al. (2008). "Network Routing Topology Inference from End-to-End Measurements". In: *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*. DOI: 10.1109/INFOCOM.2008.16.

Ni, Jian et al. (2010). "Efficient and dynamic routing topology inference from end-to-end measurements". In: *IEEE/ACM Transactions on Networking (TON)* 18.1, pp. 123–135.

Nielsen, Jakob (2001). "Success rate: The simplest usability metric". In: *Jakob Nielsen's Alertbox* 18.

Nong, Thanh-Hieu et al. (2014). "Aggregating Internet access in a mesh-backhauled network through MPTCP proxying". In: *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, pp. 736–742.

Obar, Jonathan A and Andrew Clement (2013). "Internet surveillance and boomerang routing: A call for Canadian network sovereignty". In:

Orda, Ariel, Raphael Rom, and Nahum Shimkin (1993). "Competitive routing in multiuser communication networks". In: *IEEE/ACM Transactions on Networking (ToN)* 1.5, pp. 510–521.

Owens II, Harold and Arjan Durresi (2015). "Video over software-defined networking (vsdn)". In: *Computer Networks* 92, pp. 341–356.

Perez-Gonzalez, Daniel, Pedro Soto-Acosta, and Simona Popa (2014). "A Virtual Campus for E-learning Inclusion: The Case of SVC-G9." In: *J. UCS* 20.2, pp. 240–253.

Periakaruppan, Ram, Evi Nemeth, et al. (1999). "GTrace: A Graphical Traceroute Tool." In: *LISA*. Vol. 99, pp. 69–78.

Peshkin, L. and V. Savova (2002a). "Reinforcement learning for adaptive routing". In: *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*. Vol. 2, pp. 1825–1830. DOI: 10.1109/IJCNN.2002.1007796.

Peshkin, Leonid and Virginia Savova (2002b). "Reinforcement learning for adaptive routing". In: *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*. Vol. 2. IEEE, pp. 1825–1830.

Phung, Dung Chi et al. (2014). "The OpenLISP control plane architecture". In: *IEEE Network* 28.2, pp. 34–40.

Qazi, Zafar Ayyub et al. (2013). "SIMPLE-fying middlebox policy enforcement using SDN". In: *ACM SIGCOMM computer communication review* 43.4, pp. 27–38.

Quan, Lin and John Heidemann (2010). "On the characteristics and reasons of long-lived internet flows". In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, pp. 444–450.

Quereilhac, A. et al. (2011). "NEPI: An integration framework for Network Experimentation". In: *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pp. 1–5.

Quoitin, Bruno and Olivier Bonaventure (2005). "A cooperative approach to interdomain traffic engineering". In: *Next Generation Internet Networks, 2005*. IEEE, pp. 450–457.

Raghavan, Barath et al. (2012). "Software-defined Internet architecture: Decoupling architecture from infrastructure". In: pp. 43–48.

Raiciu, Costin et al. (2010). "Data center networking with multipath TCP". In: *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, p. 10.

Rajahalme, Jarno et al. (2011). "On name-based inter-domain routing". In: *Computer Networks* 55.4, pp. 975–986.

RIPE, NCC (2015). *OpenIPMap database*.

Rodriguez-Natal, Alberto et al. (2015). "LISP: a southbound SDN protocol?" In: *IEEE Communications Magazine* 53.7, pp. 201–207.

Rothenberg, Christian Esteve et al. (2012). "Revisiting routing control platforms with the eyes and muscles of software-defined networking". In: ACM, pp. 13–18.

Rouskas, George N et al. (2013). "ChoiceNet: Network innovation through choice". In: *Optical Network Design and Modeling (ONDM), 2013 17th International Conference on*. IEEE, pp. 1–6.

Russell, Stuart and Andrew Zimdars (2003). "Q-decomposition for reinforcement learning agents". In: *ICML*. Vol. 3, p. 656.

Ryu, SDN (2015). *Framework Community,"Ryu SDN Framework,"*

Saucez, D. et al. (2008). "Interdomain traffic engineering in a locator/identifier separation context". In: *Internet Network Management Workshop, 2008. INM 2008. IEEE*, pp. 1–6. DOI: 10.1109/INETMW.2008.4660330.

Saucez, D. et al. (2012). "Designing a Deployable Internet: The Locator/Identifier Separation Protocol". In: *Internet Computing, IEEE* 16.6, pp. 14–21. ISSN: 1089-7801. DOI: 10.1109/MIC.2012.98.

Saucez, Damien, Benoit Donnet, and Olivier Bonaventure (2008). "Idips: Isp-driven informed path selection". In:

Saucez, Damien, Luigi Iannone, Olivier Bonaventure, et al. (2009). "OpenLISP: An open source implementation of the locator/ID separation protocol". In: *ACM SIGCOMM Demos Session*, pp. 1–2.

Saucez, Damien et al. (2011). "Mechanisms for interdomain Traffic Engineering with LISP". PhD thesis. UCL.

Sauro, Jeff (2011a). *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC.

— (2011b). "Measuring usability with the system usability scale (SUS)". In:

Secci, S. et al. (2011a). "Resilient Traffic Engineering in a Transit-Edge Separated Internet Routing". In: *Communications (ICC), 2011 IEEE International Conference on*, pp. 1–6. DOI: 10.1109/icc.2011.5963439.

Secci, Stefano, Kunpen Liu, and Bijan Jabbari (2013). "Efficient inter-domain traffic engineering with transit-edge hierarchical routing". In: *Computer Networks* 57.4, pp. 976–989.

Secci, Stefano et al. (2011b). "Peering equilibrium multipath routing: a game theory framework for internet peering settlements". In: *Networking, IEEE/ACM Transactions on* 19.2, pp. 419–432.

Secci, Stefano et al. (2011c). "Resilient inter-carrier traffic engineering for Internet peering interconnections". In: *Network and Service Management, IEEE Transactions on* 8.4, pp. 274–284.

Shavitt, Y. and U. Weinsberg (2011). "Quantifying the Importance of Vantage Point Distribution in Internet Topology Mapping (Extended Version)". In: *Selected Areas in Communications, IEEE Journal on* 29.9, pp. 1837–1847. ISSN: 0733-8716. DOI: 10.1109/JSAC.2011.111008.

Shavitt, Y. and N. Zilberman (2011). "A Geolocation Databases Study". In: *Selected Areas in Communications, IEEE Journal on* 29.10, pp. 2044–2056. ISSN: 0733-8716. DOI: 10.1109/JSAC.2011.111214.

Shavitt, Yuval and Eran Shir (2005). "DIMES: Let the Internet measure itself". In: *ACM SIGCOMM Computer Communication Review* 35.5, pp. 71–74.

Shneiderman, Ben (1996). "The eyes have it: A task by data type taxonomy for information visualizations". In: *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, pp. 336–343.

Shrimali, Gireesh, Aditya Akella, and Almir Mutapcic (2010). "Cooperative interdomain traffic engineering using Nash bargaining and decomposition". In: *IEEE/ACM Transactions on Networking (TON)* 18.2, pp. 341–352.

Shu, Z. et al. (2016). "Traffic engineering in software-defined networking: Measurement and management". In: *IEEE Access* 4, pp. 3246–3256. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2582748.

Singh, Sandeep Kumar, Tamal Das, and Admela Jukan (2015). "A survey on internet multipath routing and provisioning". In: *IEEE Communications Surveys & Tutorials* 17.4, pp. 2157–2175.

Song, S (2011). "AfTerFibre (Archived) Mapping Terrestrial Fibre Optic Cable Projects in Africa-Archived". In: *Many Possibilities*.

Steiner, Roy et al. (2005). "Promoting African Research and Education Networking". In: *International Development Research Center, January*.

Stringer, Jonathan et al. (2014). "Cardigan: SDN distributed routing fabric going live at an Internet exchange". In: *2014 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, pp. 1–7.

Subedi, Tara Nath, Kim Khoa Nguyen, and Mohamed Cheriet (2015). "OpenFlow-based in-network Layer-2 adaptive multipath aggregation in data centers". In: *Computer Communications* 61, pp. 58–69.

Suksomboon, Kalika et al. (2010). "PC-nash: QoS provisioning framework with path-classification scheme under nash equilibrium". In: *The Computer Journal*, bxq084.

Tirumala, Ajay et al. (2005). "Iperf: The TCP/UDP bandwidth measurement tool". In: *htt p://dast. nlanr. net/Projects*.

Tomovic, Slavica, Neeli Prasad, and Igor Radusinovic (2014). "SDN control framework for QoS provisioning". In: *Telecommunications Forum Telfor (TELFOR), 2014 22nd*. IEEE, pp. 111–114.

Tusubira, FF (2009). "UbuntuNet Alliance updates: implementing CORENA; phase 1 output and phase 2 plans, FEAST Meeting, Kampala". In:

UbuntuNetAlliance (2016). "State of the art of the Research Networking Infrastructure - Eastern and Southern Africa". In: *IST-Africa Conference Proceedings, 2016*. IEEE.

Valera, Francisco et al. (2011). "12 Multi-path BGP: motivations and solutions". In: *Next-Generation Internet*, p. 238.

Valiati, Eliane RA, Marcelo S Pimenta, and Carla MDS Freitas (2006). "A taxonomy of tasks for guiding the evaluation of multidimensional visualizations". In: *Proceedings of the 2006 AVI workshop on Beyond time and errors: novel evaluation methods for information visualization*. ACM, pp. 1–6.

Van Dusen, Gerald C (2014). "The Virtual Campus: Technology and Reform in Higher Education. ASHE-ERIC Higher Education Report, Volume 25, No. 5." In:

Ventre, P. L. et al. (2017). "SDN-Based IP and Layer 2 Services with an Open Networking Operating System in the GEANT Service Provider Network". In: *IEEE Communications Magazine* 55.4, pp. 71–79. ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600194.

Walton, Daniel et al. (2016). *Advertisement of multiple paths in BGP*. Tech. rep.

Wang, Ning et al. (2008). "An overview of routing optimization for internet traffic engineering". In: *Communications Surveys & Tutorials, IEEE* 10.1, pp. 36–56.

Wang, Ping and Ting Wang (2006). "Adaptive routing for sensor networks using reinforcement learning". In: *Computer and Information Technology, 2006. CIT'06. The Sixth IEEE International Conference on*. IEEE, pp. 219–219.

Wang, Richard, Dana Butnariu, Jennifer Rexford, et al. (2011). "OpenFlow-Based Server Load Balancing Gone Wild." In: *Hot-ICE* 11, pp. 12–12.

Wang, Wen, Wenbo He, and Jinshu Su (2015). "M2SDN: Achieving multipath and multihoming in data centers with software defined networking". In: *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)*. IEEE, pp. 11–20.

Wang, Yangyang, Jun Bi, and Jianping Wu (2010). "Empirical evaluation for the impact of core-edge separation on Internet routing scalability". In: *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*. IEEE, pp. 1–2.

Wang, Yufei, Zheng Wang, and Leah Zhang (2001). "Internet traffic engineering without full mesh overlaying". In: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 1. IEEE, pp. 565–571.

Wassink, Ingo et al. (2009). "Applying a user-centered approach to interactive visualisation design". In: *Trends in Interactive Visualization*. Springer, pp. 175–199.

Watkins, Christopher JCH and Peter Dayan (1992). "Q-learning". In: *Machine learning* 8.3-4, pp. 279–292.

Welzl, Michael (2005). *Network congestion control: managing internet traffic*. John Wiley & Sons.

Whyte, Scott (2012). "Project CARDIGAN An SDN Controlled Exchange Fabric". In: *NANOG (The North America Network Operators Group)* 57.

Wójcik, Robert et al. (2016). "A survey on methods to provide interdomain multipath transmissions". In: *Computer Networks* 108, pp. 233–259.

Wolf, Tilman et al. (2012). "Choice as a principle in network architecture". In: *ACM SIGCOMM Computer Communication Review* 42.4, pp. 105–106.

Xiao, Xipeng and Lionel M Ni (1999). "Internet QoS: a big picture". In: *IEEE network* 13.2, pp. 8–18.

Xu, Wen and Jennifer Rexford (2006). "MIRO: Multi-path Interdomain Routing". In: *SIGCOMM Comput. Commun. Rev.* 36.4, pp. 171–182. ISSN: 0146-4833. DOI: 10.1145/1151659.1159934. URL: http://doi.acm.org/10.1145/1151659.1159934.

Xu, Xin, Lei Zuo, and Zhenhua Huang (2014). "Reinforcement learning algorithms with function approximation: Recent advances and applications". In: *Information Sciences* 261, pp. 1–31.

Yan, Jinyao et al. (2015). "HiQoS: An SDN-based multipath QoS solution". In: *China Communications* 12.5, pp. 123–133.

Yin, Hao et al. (2014). "Big data: transforming the design philosophy of future internet". In: *IEEE network* 28.4, pp. 14–19.

Zennaro, M et al. (2006). "Scientific Measure of Africa's Connectivity". In: *Information Technologies & International Development* 3.1, pp–55.

Zhang, Chongjie and Victor Lesser (2013). "Coordinating multi-agent reinforcement learning with limited communication". In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1101–1108.

Zhang, Min et al. (2009). "Analysis of udp traffic usage on internet backbone links". In: *Applications and the Internet, 2009. SAINT'09. Ninth Annual International Symposium on*. IEEE, pp. 280–281.

Zhang, Wei et al. (2014). "Multipath transport based on application-level relay service and traffic optimization". In: