INSTITUTE of TECHNOLOGY CARLOW
Institiúid Teicneolaíochta Cheatharlach

Enabling models of Internet eXchange Points to support spatial planning: the case for East Africa

Submitted by Diarmuid Ó Briain

For the award of Doctor of Philosophy
from Quality and Qualifications Ireland

Institute of Technology, Carlow

Supervised by

Dr Yvonne Kavanagh
Eng. Dr Dorothy Okello
Mr David Denieffe

Submitted to the Institute of Technology, Carlow, September 2019

Declaration

I, the undersigned, certify that the information I have provided in this thesis is correct and that I have read and am aware of my responsibilities as detailed in Institute of Technology Carlow's Policy and Procedures for Postgraduate Research Students (Admissions, Registration, Supervision and Examination). I further confirm that I am unaware of any potential conflicts of interest that would compromise the Institute and / or the applicant in pursuit of the level 10 award sought. I hereby declare that I am the author of the thesis and that no part has been submitted to any other institution for examination. I certify that, to my knowledge, this thesis does not violate any plagiarism standards or infringe on copyright or proprietary rights. Any ideas, knowledge, techniques, statements and any other included materials are rightfully acknowledged in accordance with standard referencing practices. This document is a true copy of my thesis to date and includes all final revisions by my supervisors. This is original work and all conclusions made are drawn from original work and data gathered by me.

_____ 30 September 2019

Diarmuid Ó Briain

Acknowledgements

I would like to thank the Uganda Communications Commission (UCC) for the generous grant that enabled me to access the equipment necessary to build the proof of concept and models in this degree. I would also like to thank Kyle Spencer, Director of the Uganda Internet eXchange Point (UIXP) for allowing me to volunteer at the eXchange and put my mark on it. I must also acknowledge Barry O'Donovan and Nick Hilliard from Internet Neutral Exchange Association (INEX) who were a technical soundboard whenever I needed it and a special thank you to Barry Rhodes the former Chief Executive at INEX who went to his eternal reward in September 2018.

I would also like to acknowledge the emotional and logistical support of the staff at the Irish Embassy, Kampala where my wife Áine has been posted for the last four years.

I extend a very special thank you to the technical staff, students and researchers at netLabs! UG in the Department of Electrical and Computer Engineering at Makerere University and in particular my colleague and good friend Dr Jonathan Serugunda who was always positive, enthusiastic and supportive.

Sincere thanks are also due to my supervisors, Dr Yvonne Kavanagh and David Denieffe at the Institute of Technology, Carlow and to Eng. Dr Dorothy Okello from Makerere University. Their guidance on both continents as well as their friendship helped me navigate through this research work and to strengthen it as it progressed.

My family back home in Ireland, my father James, my mother Sarah and my sister Edel in Tipperary were always a positive source of energy. My sons Cian and Conor always monitored my progress and offering their support from the treaty city.

My wife Áine who gave me the space, support and love to do this work. She encouraged, pushed and made sure I had all I needed to pursue the research to the end, for all this and much more I dedicate this thesis to her.

Diarmuid Ó Briain, Kampala, September 2019

List of publications

Ó Briain, D., Denieffe, D., Kavanagh, Y., Okello, D. (2018). A Proposed Architecture for Distributed Internet eXchange Points in Developing Countries. IST-Africa 2018, 09-11 May 2018, Gaberone, Botswana. IEEE. ISBN: 978-1-5386-7165-8.

Ó Briain, D., Denieffe, D., Kavanagh, Y., Okello, D. (2017). Rebuilding the Internet eXchange Point in Uganda. 28th Irish Signals and Systems Conference, 20/21 June 2017, Killarney, Ireland. IEEE. ISBN: 978-1-5386-2221-6, DOI: 10.1109/ISSC.2017.7983593.

Ó Briain, D., Denieffe, D., Kavanagh, Y., Okello, D. (2017). Elastic Everything - What of the Developing World?. IST-Africa 2017 Conference 31 May - 02 June 2017, Windhoek, Namibia. IEEE. ISBN: 978-1-5386-3837-8, DOI: 10.23919/ISTAFRICA.2017.8102289.

Ó Briain, D., Denieffe, D., Kavanagh, Y., Okello, D. (2016). The move to a software defined future and the implications for Uganda. 15 September 2016. National Conference on Communications (NCC), Mbarara. Uganda Communications Commission (UCC). ISBN: 978-9970-615-01-8.

Awards

The Uganda Communication Commission (UCC) award for best paper at the 4th National Conference on Communications (NCC) on the 16 September 2016 in Mbarara, Uganda entitled "*The move to a software defined future and the implications for Uganda*".

Abstract

In 2009 fibre-optic cables landed on the East coast of Africa, the last major area of the world to be connected to the Internet triggering a decade of Internet development. During the same period there has been a general transformation of the Internet from static content to video streaming. Technologies such as Software Defined Networking (SDN) and Network Functions Virtualisation are about to reshape the Internet once again. Globally Internet eXchange Points (IXP) have been a key node on the Internet and a central location for Content Delivery Networks, though in East Africa they have generally been confined to large cities. There is an understanding that if technology hubs are to develop in other cities, the Internet ecosystem, including IXPs, must extend outwards.

This research uses a Proof of Concept (PoC) system design methodology to investigate solutions that containerise IXP functions and develops affordable models for IXPs of various sizes and configurations based on both traditional and software-defined switching paradigms as well as automate the IXP build function. The PoC also includes the centralised management of remote IXPs. The research argues that it is necessary to develop a national IXP ecosystem by supplementing the national IXP with local IXPs to support economic development outside of the major economic cities of the region. The technology solutions must be used in conjunction with research on the political economy landscape plus optimum deployment to ensure success. This research demonstrates that systems can be designed which are achievable and affordable by exploiting the most suitable model and switching technology for each site. It also determines that software-defined models offer the potential for application development across the IXP .

This research concludes that with a combination of function containerisation and astute model selection it is possible to build an affordable set of IXPs to support multiple technology hubs across a national Internet ecosystem. Proposed systems are discussed in the context of East Africa and testbed results discussed in relation to the optimum system design which can be deployed in any IXP setting.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

4G	Fourth Generation
A4AI	Alliance for Affordable Internet
AFIX	African Internet eXchange Point Association
AfPIF	African Peering and Internet Forum
AIXP	Arusha City Internet eXchange Point
AMS-IX	Amsterdam Internet eXchange
AR	Action Research
AS	Autonomous System
ASN	Autonomous System Number
AWS	Amazon Web Services
BCS	Bandwidth & Cloud Services
BGP4	Border Gateway Protocol version 4
BIND9	Berkeley Internet Name Domain version 9
BIRD	BIRD Internet Routing Daemon
BS	Black-hole Server [AS112 anycast service]
BTS	Base Transceiver Stations
CAPEX	CAPital EXpenditure
CDIP	Control/Data Interface Pair
CEO	Chief Executive Officer
CINX	Cape Town Internet Exchange
CLI	Command Line Interface
COTS	Common Off The Shelf
CS	Route Collector Server
DAL	Device and Resource Abstraction Layer
DE-CIX	Deutscher Commercial Internet eXchange
DINX	Durban Internet Exchange Point
DIXP	Dodoma Internet eXchange Point
DOCSIS	Data Over Cable Service Interface Specification
DRC	Democratic Republic of the Congo
DSRM	Design Science Research Methodology
EACO	East African Communications Organisation
EAIXP	East Africa Internet eXchange Point
EASTECO	East African Science and Technology Commission
EUPL	European Union Public Licence
FE	Fast Ethernet
FNA	Facebook Network Appliance
FTTH	Fibre To The Home
GDP	Gross Domestic Product
GEPON	Gigabit Ethernet Passive Optical Network
GNU	GNU is Not Unix
GPON	Gigabit Passive Optical Network
GSM	Global System for Mobile communications
HDTV	High Definition Television
HVAC	Heating, Venting, and Air Conditioning
iCAIR	International Center for Advanced Internet Research
ICTES	ICT Enabled Services
IDC	Internet Data Centre
iGENI	International GENI
IoE	Internet of Everything
IP	Internet Protocol
ISG	Industry Specification Group
ISTA	Information Society and Technology, IST Africa
IT	Information Technology
IXP	Internet eXchange Point
JSON	JavaScript Object Notation
KIEMS	Kenya Integrated Election Management System
KPI	Key Performance Indicator
LADR	Laboratory based ADR
LGPL	Lesser General Public License
LSP	Local Service Provider
LTS	Long Term Support
LXD	LinuX Container hypervisor Daemon
MAC	Medium Access Control
MBA1	Mombasa One
MIXP	Mwanza Internet eXchange Point
MoMo	Mobile Money
5G	Fifth Generation
ADR	Action Design Research
AfNOG	African Network Operators Group
AfriNIC	African Network Information Centre
ALTO	Application-Layer Traffic Optimisation
API	Application Programming Interface
ARPU	Average Revenue Per User
ASIC	Application Specific Integrated Circuit
ASP	Application Service Provider
BBS	Burundi Backbone System
BDIXP	Burundi Internet eXchange Point
BGP	Border Gateway Protocol
BIND	Berkeley Internet Name Domain
BPO	Business Process Outsourcing
BSC	Base Station Controllers
CAK	Communications Authority of Kenya
ccTLD	country code Top Level Domain
CDN	Content Delivery Network
CIN	Cloud Integrated Network
cIXP	core Internet eXchange Point
CNAME	Canonical NAME
C-RAN	Centralised - Radio Access Network
CTO	Chief Technical Officer
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
dIXP	distributed set of Internet eXchange Points
DNS	Domain Name System
DPID	Datapath Identifier
DR	Design Research
EAC	East African Community
EADC	East African Data Centre
EASSy	Eastern Africa Submarine Cable System
ETSI	European Telecommunications Standards Institute
Euro-IX	European Internet Exchange Association
FINTECH	FINancial TECHnology
FPCF	Finite Population Correction Factor
FTTK	Fibre To The Kurb
GENI	Global Environment for Network Innovations
GGC	Google Global Cache
GPL	GNU General Public License
GSMA	GSM Association
GSP	Global Service Provider
HTTPS	Hyper Text Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICP	Internet Content Provider
ICT	Information and Communications Technology
IDI	ICT Development Index
INEX	Internet Neutral Exchange Association [Irish IXP]
IoT	Internet of Things
ISC	Internet Systems Consortium
ISP	Internet Service Provider
IST	Information Society and Technology
ITU	International Telecommunication Union
JINX	Johannesburg Internet Exchange Point
KENET	Kenya Education Network
KIXP	Kenya Internet eXchange Point
KVM	Kernel Virtual Machine
LAN	Local Access Network
LINX	London INternet eXchange
LTE	Long Term Evolution
LXC	LinuX Container
M2M	Machine to Machine
MAN	Metropolitan Area Networks
mIXP	mini Internet eXchange Point
MNO	Mobile Network Operator
MOP	Maintenance Operation Protocol

MPL	Mozilla Public License
MTU	Maximum Transfer Unit
NBI	National Backbone Infrastructure
NCIP	Northern Corridor Integration Projects
NFV	Network Functions Virtualisation
NICI	National Information Communication Infrastructure
NICTBB	National ICT Broadband Backbone
NOFBI	National Optic Fibre Backbone
NR	New Radio
NSFNET	National Science Foundation Network
NXDOMAIN	Non-eXistent DOMAIN
OF	OpenFlow
ONF	Open Networking Foundation
OOP	Object Oriented Programming
OSI	Open Systems Interconnection
OTT	Over the Top tax implemented in Uganda
P4	Protocol-independent Packet Processors
PII	Personally Identifiable Information
PISA	Protocol Independent Switch Architecture
PoP	Points of Presence
REN	Research and Education Network
REST	REpresentational State Transfer [sometimes RESTful]
RINEX	Rwanda INternet EXchange
RITA	Rwanda Information Technology Authority
RP	Rendezvous Point
RSA	Rivest, Shamir and Adleman
RTT	Round Trip Times
SAT-2	South Atlantic Telecommunications 2
SBI	South Bound Interface
SDG	Sustainable Development Goals
SDN	Software Defined Network(ing)
SEAS	Seychelles to East Africa System
SME	Small Medium Enterprise
SMS	Short Message Service
SRD	Short Range Device [2.4 and 5 Ghz bands]
SUNET	Swedish University Computer Network
TCP	Transmission Control Protocol
TEAMS	The East African Marine System
THDC	Telephone House Data Centre
TISPA	Tanzania Internet Service provider Association
TLS	Transport Layer Security
TouSIX	Toulouse Software Internet eXchange
TTCL	Tanzania Telecommunications Company Limited
UAE	United Arab Emirates
UCC	Uganda Communications Commission
UIXP	Uganda Internet eXchange Point
URL	Uniform Resource Locator
vCE	virtual Customer Edge
VDI	Virtual Disk Image
VLAN	Virtual Local Area Network
VoD	Video on Demand
VSAT	Very Small Aperture Terminal
WASC	West Africa Submarine Cable
WEF	World Economic Forum
YAML	Yet Another Markup Language
MSIXP	Mombasa Internet eXchange Point
NAP	Network Access Point
NBI	North Bound Interface
NDP	National Development Plan
NGO	Non-Governmental Organisation
NIC	Network Information Centre
NOC	Network Operations Centre
NoVA	LINX North Virginia
NS	DNS server
NS	Name Server
ODL	OpenDaylight
OIX	Open Internet eXchange Point Association
OOB	Out of Band Management
OPEX	OPerational EXpenditure
OS	Operating System
OvS	Open virtual Switch
PCP	Primary Cross-connection Point
PISA	Protocol Independent Switch Architecture
PoC	Proof of Concept
PTR	Pointer [Reverse DNS lookup]
RENU	Research and Education Network for Uganda
RICTA	Rwanda Internet Community and Technology Alliance
RIR	Regional Internet Registry
RoI	Return on Investment
RRH	Remote Radio Head
RS	Route Server
RV	Rendezvous Point
SAT-3	South Atlantic Telecommunications 3
SC	Software Defined Network Controller
SDNFV	Software Defined NFV
SDX	Software Defined eXchange
SLAAC	Stateless Address Autoconfiguration
SME	Subject Matter Expert
SQL	Structured Query Language
STM	Synchronous Digital Hierarchy
systemd	System daemon
TCRA	Tanzania Communication Regulatory Authority
TESPOK	Technology Service Providers of Kenya
TIA	Telecommunications Industry Association
TIX	Tanzania Internet eXchange
TouIX	Toulouse Internet eXchange
TSI	Total Societal Impact
TV	Television
UCC	Uganda Communications Commission
UETCL	Uganda Electricity Transmission Company Limited
URA	Uganda Revenue Authority
VAT	Value Added Tax
vCPE	virtual Customer Premise Equipment
VDSL2	Very high bit rate Digital Subscriber Line v2
VM	Virtual Machine
VoIP	Voice over Internet Protocol
WAN	Wide Area Networks
WDM	Wavelength Division Multiplexing
WTO	World Trade Organisation
ZIXP	Zanzibar Internet eXchange Point

Terms in use

This document uses standard industry terminology and a list of abbreviations is included to aid the reader. Each abbreviation is given in full at first use in each chapter, subsequent usage of the term within the chapter is displayed in abbreviated form.

Square brackets *[]* used in a quote represents additional text placed there to clarify the meaning, for example: "*It is not happening only in ICT because even .....*". It is not clear what the context is for the quote. To be explicit the word *investment* is added in square brackets to clarify the context "*[Investment] is not happening only in ICT because even .....*".

As the word ***region*** can be interpreted differently depending on the context within which it is used this document attempts to simplify the language for the purposes of clarity. Where possible throughout this text the term ***continent*** is used to refer to the continent of Africa, ***region*** refers to the East African Community (EAC) and Uganda, Kenya and Rwanda are collectively referred to as the ***sub-region***. The term ***local*** is used as a general term to refer to an economic hub city or town and its hinterland.

1. Introduction

Very little changed in computer networking and telecommunications during the migration of Information Technology (IT) to cloud computing during the last decade with the exception of speed improvements. However, disruptive technologies in the form of Software Defined Networking (SDN) and Network Functions Virtualisation (NFV) are beginning to make their presence felt. These technologies have established the concept of elastic network and elastic functions which complement elastic compute and elastic storage technologies underpinning the earlier cloud computing transformation.

Another change that has occurred quietly is a flattening of the Internet. Traffic patterns today have switched from static text and picture based content to video which now accounts for over 75% of all traffic today. Of that, over half is strategically placed on Content Delivery Networks (CDN) located on local Internet Service Provider (ISP) and Internet eXchange Point (IXP) networks, bypassing the tiered approach to Internet design. It is expected that by 2022 the share of video hosted on CDNs will rise to 70%. In many developed countries the original IXP at national level has been augmented to include regional or local IXPs located in alternative hub towns and cities. These developments have responded to the ever increasing demands of applications for lower latency, lower packet loss and higher bandwidth.

In developing countries urban/rural migration is leading to ever increasing city populations and the growth of unplanned slum areas. The U.N. through Sustainable Development Goal (SDG) 9: *Industry, innovation and infrastructure* has indicated that spatial planning is necessary to address this ever growing problem. In the context of this larger problem, IXPs form a key component for the delivery of the Internet outside of capital cities. The development of regional or local IXPs, as has occurred in developed countries, has not always been commercially and/or technically viable in developing countries and can remain hidden among the many other challenges to infrastructure provision associated with spatial planning.

1.1 Focus and Scope

Living in East Africa, particularly when travelling outside capital cities, it is all too easy to consider *how Internet penetration can be improved?* and *if it is improved could it have a genuine impact on the marginalised in society?*. There are so many who migrate to the capitals in search of opportunities and find themselves living in slums at the edge of the city with little to no services provided by central government or city authorities.

Having over 30 years experience in the networking industry in Europe, East Africa this presented new and interesting challenge. To really understand the underlying Internet and networking industry it was important to first understand the environment. These challenges lead to the Uganda Internet eXchange Point (UIXP) which in turn lead to the question; *is the IXP in Kampala enough?* and *do not other towns and cities also deserve the facility of an IXP?* It was these questions that led to this research.

The research is divided into two parts, a mixed-methods social-political study across the region which analyses Internet development in East Africa since the landing of submarine fibre-optic cables in 2009. The research also posits what the future might bring in the context of this changing infrastructure and technologies.

The second part focuses on a single aspect of this future development; a Proof of Concept (PoC) that can demonstrate the feasibility of a distributed set of IXPs in the context of a developing nation. The PoC addresses simplicity through automation, affordability through containerisation and the selection of appropriate hardware for various types of site requirements. The PoC has developed models that support both adaptability and exchange scalability, from small remote mini IXPs (mIXP) to larger centralised core IXPs (cIXP) in either IXP or Software Defined eXchange (SDX) modes. Skills gaps as well as the scalability of a set of IXPs are catered for through a centralised management model applied to a distributed set of IXPs. Future proofing is considered through the incorporation of SDN such that the IXP can also be deployed in a SDX mode.

Network connectivity between IXP sites and the security of Internet connectivity serving an

IXP site can take many forms. Apart from a general recommendation that information and network security considerations of IXPs sites as well as privacy concerns must be considered in a production environment, they are outside the scope of this research.

1.2 Relevance of the research

East Africa was the last area of the world to be connected to the Internet via submarine fibre-optic cables. This has had a particular impact on the Internet's development in the region and continues to influence both the political economy of the Internet and network design considerations by service providers today. In parallel, governments are grappling with the issue of migration and improving spatial planning as a key tool to address this issue.

IXPs are a key element of the Internet ecosystem and it is therefore necessary to consider them within spatial development plans. Developing countries have achieved this through the development of independent IXPs in hub towns and cities. SDN has also been incorporated into such IXPs to form SDXs. However, such developments and the associated research has largely focused on large scale solutions as well endeavouring to address future challenges such as enabling more secure and flexible traffic engineering techniques to inter-domain and multi-domain problems and the interconnection of SDN islands. By contrast, this research focuses on the delivery of models that can deliver the correct scale, configuration, technology and independence for each individual site while also allowing each to participate within a centrally managed distributed set of IXPs.

1.3 Vision

"*Spatial cities in developing countries deserve a fully developed Internet ecosystem that includes an active IXP*". This research enables this to become a reality.

1.4 Problem Statement

Today, due to ineffective spatial planning, migration within developing countries is into capital cities. People migrate in search of employment opportunities and better services. One key element of the solution is the need for IXPs within the Internet ecosystem of hub towns and cities.

The problem is exacerbated by a lack of local skills in these hub towns and cities in order to manage and maintain independent IXPs. Any solution that supports technology hubs to thrive must therefore incorporate an IXP that is affordable, scalable and easily managed.

1.5 Research Questions and Objectives

In order to develop an effective system which can be deployed in the East Africa context the following questions form the basis for this research. It is important to first understand the landscape of the area to determine what is required and therefore the research question that informs this is:

> *What are the political, social, economic and technical drivers that have influenced Internet development in East Africa over the last decade and into the future?*

This answer to this question leads to critical review of the the Internet from 2009 to today and into the future. It forms the basis for a specific objective to *investigate the political, social, economic and technical drivers that have influenced Internet development in East Africa to date and what is the expected direction into the future*.

The next three questions are considered via a *build-evaluate* PoC cycle that answers each question in turn and builds upon each answer with the next iteration of the cycle.

> *How can models of IXP be developed to cater for local IXPs in the context of developing countries?*

This question drives the objective to *develop a PoC to identify potential system models, which will deliver a key set of easily deployable IXP nodes in the future Internet ecosystem, within the constraints of a developing nation*. Once the PoC is operational consideration as to the potential of software-defined technologies offering the potential for new services leads to the next question,

> *How and what are the potential benefits of the incorporation of SDN into IXP models to create an SDX?*

which establishes an objective to *evaluate if SDN can be incorporated into the IXP models to create a SDX, to list the possible benefits and suggest optimum solutions based on the East*

African context.

> *How can remote IXPs be centrally managed without interfering with their independence from a peering perspective?*

The objective raised from this final question considers how *a method within the PoC for centralised management of the set of IXPs within a dIXP can be identified*. This resolves potential skills issues at remote IXPs as the day-to-day management can be carried out centrally. Answering these questions has led to the development of a new system of solutions, which are affordable, scalable that can be deployed in remote areas and managed centrally.

1.6 Conceptual Framework

This research focuses on the delivery of IXPs through the development of a PoC informed by a mixed methods study and survey. A number of inputs were identified and formed the nucleus of the literature review topics which in turn informed the technological landscape.

The processes begin with a research design that provides a roadmap for the remaining processes within the research. The next step involved conducting and analysing a mixed methods study to gain a deep understanding of the Internet in East Africa that was both rich in quality and thick in quantity (Dibley, 2011). A set of *build-evaluate* iterative processes looped such that the output of one iteration fed the input of the next loop iteration with each passing through system architecture, design, development and build process stages.

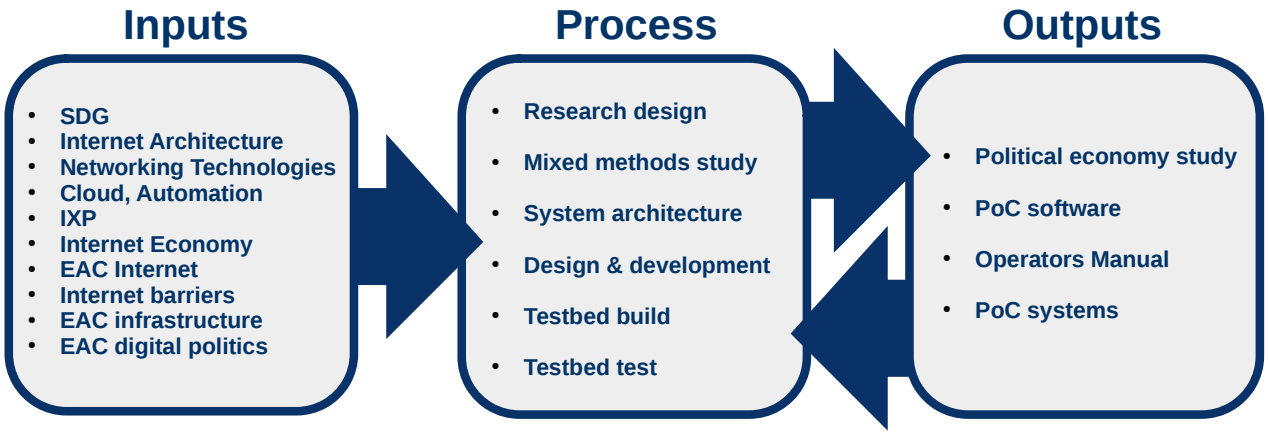


Figure 1: Conceptual Framework

The outputs of the political economy study are documented in chapter 4 of this thesis. The output of each iteration of the *build-evaluate* iterative processes produced a working build of code as well as a PoC operators manual. These were not terminal outputs as they fed back into the next build-evaluate iteration. These relationships are illustrated in Figure 1.

1.7 Overview map of the structure

This research was conceived in East Africa, the last major area of the world to gain access to the Internet via submarine fibre-optic cable just a decade ago. A literature review as well as a mixed methods study was carried out in order to inform the research since the events are still fresh in the memories of the engineers and bureaucrats who conceived, designed, built and regulated these networks. The IXP, as a key pillar of the the Internet ecosystem was explored with a view to expediting their incorporation into local Internet ecosystems to improve network latencies, reduce hop count and packet loss as well as jitter.

Chapter 2 – *Literature Review* is a narrative review of current knowledge relating to the contribution of the Internet to regional and national Gross Domestic Product (GDP). It explores the elements of the Internet and how these elements deliver a regional Internet ecosystem. New technologies in the form of SDN and NFV are considered as these have the potential for disruption. A close examination was undertaken of the IXP including how they are constructed and operate as well as a view to their future in the form of SDX. There is a short review of the economy of the Internet before a detailed examination of the Internet in East Africa is presented.

Chapter 3 – *Methodology* sets out the overall design of this research as well as the architecture of the PoC system. It describes the *build/evaluate* looped nature of the development with each iteration built upon its antecedent.

Chapter 4 – *The Internet in East Africa, a mixed methods study* leverages the recent arrival of the Internet in East Africa by capturing the experiences and opinions of the people involved in the business since the first cable landed in 2009. The chapter details a targetted subset of the mixed methods study that explored the history, operations and future of the Internet in East Africa starting

in 2009 when the submarine fibre-optic cables landed. It has a particular focus on IXP and software-defined disruptive technologies. Acquiring the information in this way not only obtained the facts in relation to the evolution of the Internet in East Africa but also sets out the key opinions of SMEs who were and continue to drive this evolution.

Chapter 5 – *A Proof of Concept for cost effective models of IXPs* develops new IXP and SDX models as part of the PoC. The PoC started with a high level functional specification derived from both the literature review and the mixed methods study, as well as drawing on the researcher's 30 years of industry experience. This functional specification guides the design of the PoC testbed and software. The chapter details the structure of the PoC for both *traditional* and *software-defined* models, the importance of the IXP schema, the steps and underlying automation the PoC takes to build each IXP from the schema information. It also traverses the internal structures created during the automated build phase as well as during the day-to-day management of IXP peers. The potential for IXP application development is considered via the exposure of the functional RESTful APIs, particularly with the SDN Controller in SDX mode. An example build of the testbed brings the chapter to a close with a demonstration of connectivity tests and a discussion of usability tests that establish conformity to the design specifications.

Chapter 6 – *Results and Discussion* considers the findings from the overall research as well as the link between the literature and the outputs of the mixed methods study with the PoC. The chapter outlines the results of the PoC as well as outlining some limitations associated with the research.

Chapter 7 – *Conclusions, Recommendations and Future work* outlines the answer to each of the research questions and reflects on the functionality demonstrated by the PoC. Some recommendations, including in relation to any future productisation of the PoC, are presented as well as a discussion on future research that can be considered arising from this thesis completes the thesis.

2. Literature Review

2.1 Introduction

This literature review provides an overview of the development of the Internet ecosystem with a particular focus on Africa, specifically East Africa. It evaluates current and expected international systems which have been developed and considers their applicability to this region and context.

An overview of the current direction networking is taking is also provided endeavouring to grapple with the continuing disruptive influence of virtualisation, containerisation and cloud computing which are manifesting themselves through the *Software-Defined* paradigms of Software Defined Networking (SDN) and Network Functions Virtualisation (NFV). The potential for flexibility and automation that these technologies offer, suggest major upcoming shifts in the industry. This can been seen in the design plans for 5G New Radio (NR) as well as research into Software Defined eXchanges (SDX). The changes to networks and functions through software-defined technologies, alongside the transformational effect of cloud computing, could lead to a further paradigm shift from a centralised model to a more distributed model to a Cloud Integrated Network (CIN) (Weldon, 2015). It is also important to note that statistical evidence demonstrates that Internet traffic patterns have changed to the point where the vast majority of traffic today is video in nature and this growth trend is expected to increase even further over the next decade (Cisco, 2019).

The review considers the current make-up of the Internet focusing on the function of the Internet eXchange Point (IXP) node in national and regional Internet ecosystems. It also explores the changing nature of the IXP as a key location for Internet Content Providers (ICP) to locate Content Delivery Networks (CDN) thereby flattening the traditional tiered model of the Internet. Business cases for IXP local hubs in these countries is also examined.

Africa has been the forgotten continent for many decades and relied on satellite connectivity to access the global Internet. Apart from South Africa, which was connected to Europe in 1993 via

submarine fibre-optic cable, West Africa wasn't connected until 2002 and East Africa until 2009 (TeleGeography, 2019). These fibre-optic connections have has a transformational impact on the Internet across the continent.

Until the arrival of the submarine fibre-optic cables, regions that did not need to have cross-border connections as satellite connections were beamed directly to each Internet Service Provider (ISP) and business via earth landing sites typically located in Europe. There was then a sudden imperative for neighbouring countries to co-operate in order that the inland countries could benefit from the new submarine fibre-optic cables. Over the last decade an ecosystem of terrestrial fibre-optic connections has linked the internal regions within African countries to varying extents. Section 2.7 of this literature review focuses on East Africa, in particular on Kenya, Uganda and Rwanda. This sub-region includes Kenya, on the coast, with multiple submarine fibre-optic landing sites at Mombasa; an inland country, Uganda which borders Kenya and is dependent on connectivity from its neighbour; and finally Rwanda which depends on multiple relationships (that with its neighbours in Uganda and Tanzania but also Kenya with which it doesn't share a border). This sub-region is a microcosm of Africa. These relationships are not simply technical, they are also highly influenced by political and economic considerations.

2.2 Sustainable Development Goals: ICT can contribute

The U.N. General Assembly adopted the 2030 *Agenda for Sustainable Development* which includes 17 Sustainable Development Goals (SDG) emphasising a holistic approach to achieving sustainable development for all (U.N., 2015). Goal 9: *Industry, innovation and infrastructure* identifies that Internet access has a direct impact on Gross Domestic Product (GDP). It is estimated that on average a 10% increase in mobile broadband adoption leads to an 0.8% increase in GDP (Edquist *et al.*, 2018). This penetration both increases access to services and presents a new avenue for businesses to access customers. The Internet also provides a new platform for business to develop products and local content creation drives more usage which has a general upswing for economic growth.

Goal 9 further identifies that the Internet can also be a tool to enable governments to encourage the redistribution of jobs from capital cities that in many cases cannot absorb the level of rural migration happening today. This requires spatial planning and the development of innovation hubs in other cities and towns. These hubs require adequate Internet access and mobile penetration as well as a local Internet infrastructure to facilitate ecosystems away from the capital.

Currently, across all the key SDG indicators, Africa lags behind other regions of the world. For example, at 25%, Africa has the lowest Internet usage in the world, followed by Asia/Pacific at 48% and Europe leading the way at ~80% (ITU-T, 2018). This is borne out in the International Telecommunication Union's (ITU) Global Information Communications Technology (ICT) Development Index (IDI) (ITU, 2017) which places most African countries in the lower quartile. Kenya is the highest placed East African country in 138th place of the 176 countries measured. The sub-region index average is 2.43 which is below the African continental index of 2.64 and well below the global average of 5.11. The Alliance for Affordable Internet (A4AI) has further analysed this data in their 2018 Affordability Report (A4AI, 2018) and note that despite a recognition of the critical relationship between Internet access and GDP, the pace of the policy change required to drive Internet prices down had slowed penetration growth to 1% in the 12 months since the previous report.

2.3 The Internet

The Internet is a large ecosystem of connected computers and networks from around the world that allows for the sharing of information and communication using the Internet Protocol (IP). It is structured in a hierarchy of ISPs that are interconnected to provide services.

2.3.1 Transit

> **Internet Transit** is the business relationship whereby an Internet Service Provider provides (usually sells) access to the global Internet (Norton, 2014).

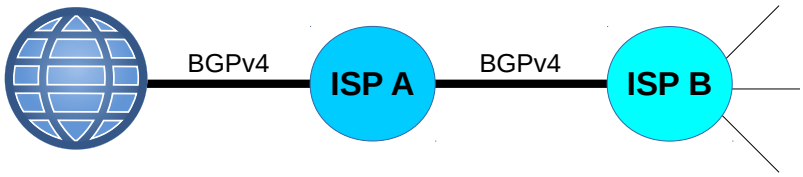


Figure 2: Internet transit

As can be visualised in Figure 2, Internet transit is essentially a link between two ISPs where one ISP purchases access to the rest of the Internet. ISP A announces the Internet Routes to ISP B as well as announcing ISP B's routes upstream to the rest of the Internet. If ISP A is a tier two ISP then it in turn must purchase Internet transit from either another tier two or preferably a tier one ISP. The routing protocol for this sharing of routes is called Border Gateway Protocol (BGP) (Hares, Rekhter and Li, 2006).

Over the last three decades there has been a global trend of Internet transit prices falling by between 15 – 50% year on year, averaging approximately 30%. In contrast Internet demand has grown at over 50% year on year. Therefore, such continuous fall in transit pricing is facilitated by the greater growth demand for Internet traffic (Norton, 2014).

2.3.2 Peering

> ***Internet peering*** *is the business relationship whereby two companies reciprocally provide access to each other's customers. Internet Peering is typically settlement-free, meaning that neither party pays the other for access to each other's customers, reflective of the underlying notion that peering is a relationship of approximately equal value to each party. Since both parties benefit about the same from the relationship, there is no need to bother with the overhead of measurement and settlement (Norton, 2014).*

Internet peering is an alternate method of acquiring routes to Internet transit. Internet companies, typically of similar size, will choose to connect to each other to share routes using BGP version 4 (BGPv4) (Hares, Rekhter and Li, 2006) and thereby remove these routes from Internet transit. Figure 3 demonstrates that peering between two ISPs typically occurs at an Internet Data Centre (IDC) where the ISPs can share a physical interconnection between each others border routers, or they can peer directly with each other using the switching infrastructure of an IXP. This

form of connection is called private peering. An additional form of peering popular at IXPs is public peering where the exchange members peer with a Route Server (RS). The RS accepts routes from each peering member and passes the received routes transparently to each other peering member (Jasinska *et al.*, 2017).

There are four main motivations for ISPs to peer (Norton, 2014):

- Reduction of Internet transit costs,
- Improved end-user experience, particularly with streaming services,
- Traffic control strategy,
- Increase traffic consumption, means more revenue.

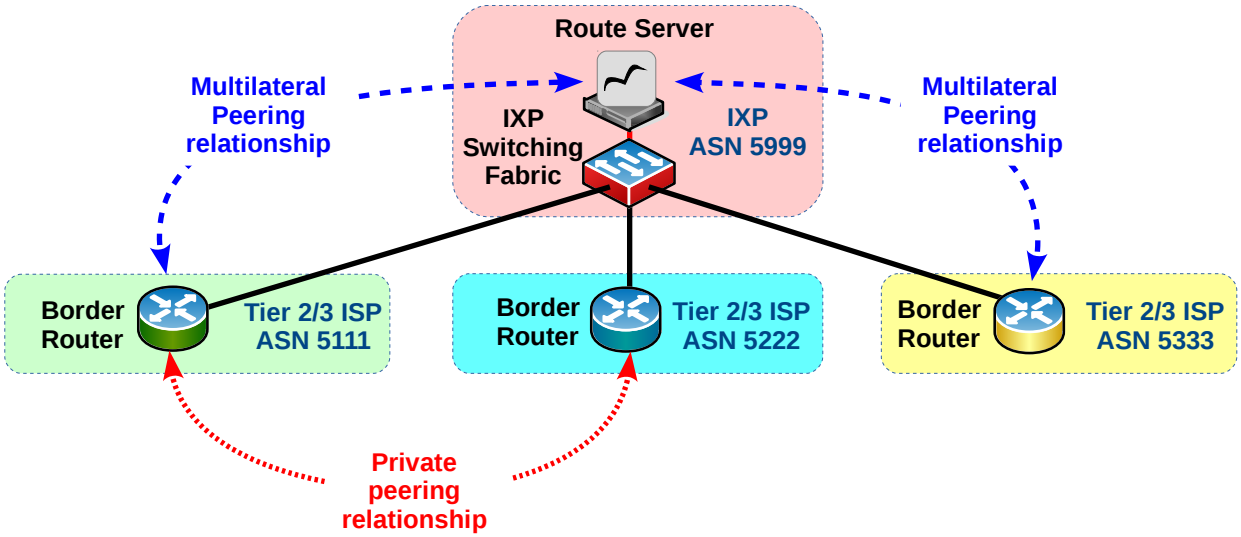


Figure 3: IXP ecosystem architecture

2.3.3 The regional Internet peering ecosystem

A set of Internet companies and organisations who collectively peer with each other within an Internet region is termed a regional Internet peering ecosystem. The ISPs in the ecosystem are represented as three tiers and also include ICP, IDC and IXPs (Norton, 2014). Each ISP is a single technical administration entity that represents an Autonomous System (AS) and therefore is assigned a unique Autonomous System Number (ASN) by the Regional Internet Registry (RIR). In the case of Africa that is the African Network Information Centre (AfriNIC). Using the inter-AS

routing protocol BGP4 the ISPs exchange network reachability information with each other in relationships called BGP peering sessions. The regional Internet peering ecosystem consists of the following entities:

- ***Tier one ISP***: An ISP that has access to the entire Internet region routing table only through settlement-free peering relationships. If the ISP requires anything more than its free and reciprocal peering arrangements, it is not a tier one ISP in that Internet region.
- ***Tier two ISP***: An ISP that purchases transit to reach some destination(s) within an Internet region but reaches some destinations through settlement free peering with other tier two ISPs either through direct bilateral connections or via multilateral connections at an IXP. Tier two ISPs sell transit to tier three ISPs, businesses as well as end-users.
- ***Tier three ISP***: An ISP that purchases Internet transit to reach all destinations (such providers are often termed Access Providers as they provide last mile connectivity to business residential customers). Considering this definition most modern ISPs, particularly those that are members of an IXP, are considered tier two ISPs.
- ***Content Providers****:* Companies that operate an Internet-based service but do not sell transit within the Internet Peering Ecosystem.
- ***IDC:*** A specialist building that houses computer servers and associated components like storage, networking and telecommunications infrastructure.
- ***IXP***: A switching or routing infrastructure that permits ISPs and ICPs to peer with each other for the purpose of sharing traffic between their networks. ICPs often do this by the implementation of CDN at the IXP.

For example, a tier three ISP in Uganda solely connected to a regional tier two ISP, all of its traffic, not destined for customers on its own network, will transit via the tier two ISP onto the Internet via a tier one ISP. Should the destination of that traffic be local and the end customer supported by a different tier two or three ISP who has purchased transit via a different tier one ISP,

then this traffic will hairpin to at least the nearest point of intersection of these networks, in the case of East Africa that could be in Kenya or even more likely via one of the European big three IXPs, Deutscher Commercial Internet eXchange (DE-CIX), Amsterdam Internet eXchange (AMS-IX) or the London INternet eXchange (LINX). In fact considering all paths through the worlds nine largest IXPs, these three account for 76% of the paths (Ahmad and Guha, 2012). The process is referred to as *tromboning* and the point of intersection is called a Rendezvous Point (RV) (Gupta *et al.*, 2013).

According to (Chavula *et al.*, 2014), on average, 75% of traceroutes from vantage points established in Africa to African NRENs traversed inter-continental links in Europe, such as Amsterdam, London, Lisbon, and Marseille.

This phenomenon has the effect of imposing three constraints of i) *latency*, ii) *cost* and iii) *reliability* on the regional Internet ecosystem and the ISPs that provide it (Kini *et al.*, 2014). In many cases, from African networks, it is faster to reach European or North American networks that those in other regions of Africa (Formoso *et al.*, 2018).

2.3.4 Data Centres

An IDC is a specialist building that houses computer servers and associated components such as storage, networking and telecommunications infrastructure. An IDC consists of the following components:

- ***Security***: A secure building of typically 2000 – 30,000m^2,
- ***Power***: Reliable, typically redundant power infrastructure,
- ***Cooling***: Heating, Venting, and Air Conditioning (HVAC) system,
- ***Connectivity***: Internet access,
- ***Peering***: Private and/or public peering,
- ***Operations***: Network Operations Centre (NOC) to manage and monitor the IDC.

As IDCs became critical infrastructure of the Internet, it became necessary to set standards

and the generally accepted standard is the Telecommunications Industry Association (TIA), Telecommunications Infrastructure Standard for Data Centres, (TIA-942-B) (TIA, 2017). TIA-942-B grades IDCs in four levels:

- ***Level-I***: Basic site infrastructure (99.671% availability),
- ***Level-II***: Redundant capacity component site infrastructure (99.741% availability),
- ***Level-III***: Concurrently maintainable site infrastructure (99.982% availability),
- ***Level-IV***: Fault tolerant site infrastructure (99.995% availability).

2.3.5 Internet maturity levels

To measure the quality of Internet in any area it is necessary to assess it based on the services it is capable of supporting. To this end it can be viewed as a list of *maturity levels* which progressively offer more services (WEF, 2018). Table 1 illustrates this as an increasing set of maturity levels from one to five, with each subsequent level offering more services than its predecessor. The table also lists the speed ranges and acceptable latency bands associated with each level to deliver the services as well as listing the fixed and mobile technologies that can deliver these maturity levels.

Table 1: Internet maturity levels

Maturity level	Acceptable latency (ms)	Speeds (Mb/s)	Fixed Broadband	Mobile Broadband	Characteristics
1	500 - 1000	0.1 – 0.5	ISDN, ADSL, DOCSIS 1.0	2.5G GPRS CDMAone	Basic services, email, browsing
2	200 - 400	0.5 – 3	ADSL, DOCSIS 1.0 – 2.0	3G UMTS, CDMA2000	Streaming video, VoIP, online business
3	100 - 50	24 – 50	VDSL, PON, DOCSIS 1.0 – 3.0	4G LTE, 4G LTE-A	Cloud storage, Online applications
4	10 - 20	100 – 1000	GPON, GEPON, DOCSIS 3.0,3.1	4G LTE-A, 5G NR	Triple play, HDTV, IoT, smart applications
5	1 - 10	1000 +	GPON, GEPON, DOCSIS 3.0,3.1	5G NR	Industry 4.0, IoE, Evolving technologies

2.3.6 The rise of video and the flattening of the Internet

Internet traffic today has switched from static text and picture based content, with video traffic expected to rise from 70% in 2015 to 82% of all consumer Internet traffic by 2020. It is also

expected that nearly 70% of all Internet video traffic by 2020 will be carried on CDNs (Cisco, 2019).

As far back as 2008 it was becoming evident that the large ICPs were bypassing the tier one ISPs by deploying their own Wide Area Networks (WAN). This brought their content nearer to their customers by peering with a larger number of lower tier ISPs via IXPs and in doing so, flattening the Internet hierarchy (Gill *et al.*, 2008). By bringing their content closer to the customer ICPs are acting on the problems associated with video delivery. As a result, Internet video content quality can today compete with traditional TV services and this is evidenced by the popularity of pay-TV services such as pay-TV stand-alone streaming devices (LRG, 2019).

2.4 Cloud Computing, Management and Automation

The move by the Information Technology (IT) sector traditional owner-owned infrastructure, software and systems to cloud computing (The Open Internet, no date) has forced industry to consider management of servers. As a result a number of cloud orchestration tools have been developed for cloud automation. Such tools offer end-to-end process management by controlling and coordinating lower level processes to deliver a completed resource or service. One of the most well known and used tools is the OpenStack Cloud Operating System (OS) which has been developed to control pools of compute, storage and network resources on standardised hardware (OpenStack Foundation, 2017). The automation of services via cloud computing speeds up the delivery of services by handing control to the end-user. An end user who requires a service, accesses the service providers portal, fills out the necessary information, pays for the service using an online credit facility, (typically a credit card or Mobile Money (MoMo)) and the automated orchestration process is invoked which in turn reserves the necessary compute, storage and network resources and instantiates the ordered service. This process usually takes no more than a few minutes. To deliver cloud based services a network maturity level of a least three is required.

2.4.1 Software-defined networks and functions

Networking as a key component of infrastructure which has not undergone the revolutionary

change witnessed in the compute and storage sectors, due to the fact that powerful vendors have controlled the industry with proprietary, specialist hardware. There have been increasing speeds and added features, but in general vendors have maintained both control and data forwarding planes within their networking platforms. These devices have relied on protocols such as BGP in use as far back as 1989 (Rekhter and Lougheed, 1989) to communicate.

Traditional networks with abstraction layering have levels of header encapsulation that impact negatively on the Maximum Transfer Unit (MTU) size, thereby reducing the effective capacity of the frames. SDN and NFV are two disruptive technologies that are having a transformative effect on networking (Stallings, 2015) with SDN having already taken hold in IDCs. These applications are supplanting traditional switching and routing technologies. The flexibility offered by SDN and NFV are key ingredients in the delivery of automated cloud based applications and are central concepts to the design of 5G NR (Lien *et al.*, 2017).

2.4.1.1 Software Defined Networking

SDN is the extraction of the control functions from networking equipment hardware (Haleplidis *et al.*, 2015). This leaves the hardware with only data plane functionality. Therefore SDN is a separation of the control and data forwarding functions within the network. The control plane functions are migrated as software functions to be ran on standard industry hardware, or more often than not on server instances located on virtualised cloud platforms.

SDN triggered a requirement for a major change in switching architecture right down to the hardware as outlined in Figure 4. The era of cloud computing also generated a need for software based switches to switch between compute devices such as Virtual Machines (VM) and Containers. Switching devices in the forwarding plane retained the data forwarding function to ensure high speed switching is maintained. The control function however is passed to a centralised SDN Controller which has visibility across the set of switches under its control.

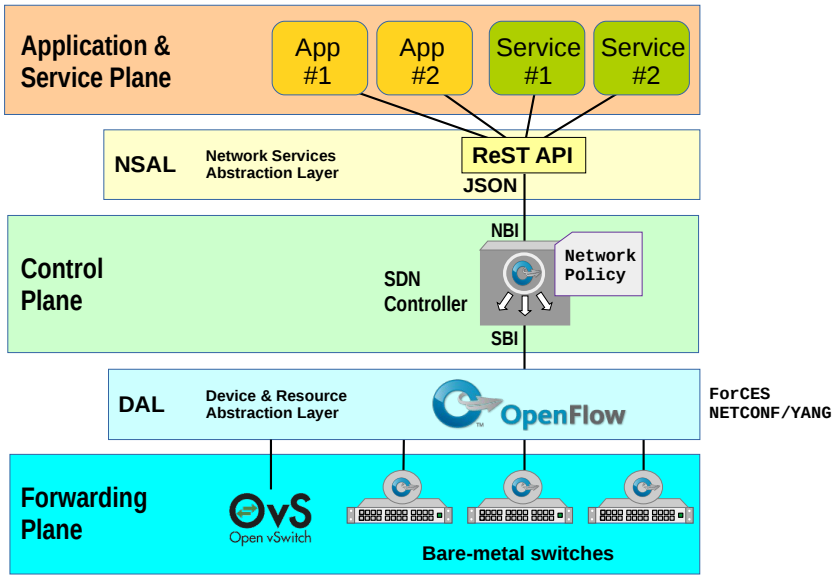


Figure 4: SDN Architecture

There are a number of protocols that can be used for this control function with OpenFlow (OF) being the most popular (McKeown *et al.*, 2008) at the Device and Resource Abstraction Layer (DAL). The SDN Controller has a South Bound Interface (SBI) that accesses each OF switch via a control channel (Transport Layer Security (TLS) over Transmission Control Protocol (TCP)/6633) to enable it to modify the OF switch flow tables.

In 2008 the virtual Switch daemon (vswitchd) was produced for the GNU/Linux kernel to demonstrate OF functionality. This project has an open source project under the Apache 2 license known as Open virtual Switch (OvS) (*Open vSwitch*, no date).

The SDN Controller installs a low priority *table-miss* rule in each switch at the time of initial registration. When a frame arrives at the OF switch and it has no specific flow rule to deal with it then the OF switch buffers the frame and forwards it to the SDN Controller in an OF *PacketIn* message for processing. The SDN Controller consults its network policy and makes a decision as to what happens the frame. It then returns an OF *PacketOut* message to the OF switch informing it what to do with the buffered message. It then follows up with an add flow rule to deal with future events of frames that have the same pattern match. In this way subsequent similar frames are handled directly by the OF switch without the need to consult the SDN Controller (RYU,

2018).

Applications and services access the SDN Controller via the North Bound Interface (NBI). This interface is typically a REpresentational State Transfer (RESTful) Application Programming Interface (API) (Fielding, 2000) at the Network Services and Abstraction Layer (NSAL) and data is represented in JavaScript Object Notation (JSON) format (Brady, 2017). With such access these services and applications can manipulate the network policy of the SDN Controller in order to control the network behaviour.

2.4.2 SDN data plane control

Over the last three years there has been a push to extend SDN beyond the control plane and look at ways to program the switch hardware itself. Switches, even OF ones, use rigid switching Application Specific Integrated Circuit (ASIC) based chip hardware and research is ongoing to develop new Protocol Independent Switch Architecture (PISA) chips, such that, it becomes possible to program the data plane processing directly and remove the reliance on vendor ASICs (Cascone, 2018). OF permits limited flexibility over the control plane as the switch vendors define which headers they support on their ASIC. OF actually gives the *SC* a means to populate the ASIC's fixed tables with flows based on these fixed header types. Future PISA based hardware will permit the direct programming of the switch using languages such as Programming Protocol-independent Packet Processors (P4) (P4, 2018). Networking languages such as P4, are the future direction for SDN as they offer a new level of flexibility through data plane programmability.

Another interesting development is work being carried out under the Stratum project at the Open Networking Foundation (ONF). This project is developing a new Protocol Independent Switch Architecture (PISA) for switch hardware design which will allow software, such as the P4 programming language, to manipulate the data plane of switches through a P4Runtime program independent Application Programming Interface (API) (Cascone, 2018). With P4 it is expected that OF switches will be (1) reconfigurable in the field, (2) independent of specific network protocols and offer (3) target packet-processing independent the design of the underlying hardware (Bosshart

et al., 2014).

This is a welcome development as it can help to avoid the vendor lock-in that exists today on switching hardware via proprietary ASIC interfaces and closed software APIs and instead deliver on the promise of PISA based white-box switch solutions (O'Connor, 2018).

2.4.2.1 Network Functions Virtualisation

NFV virtualises services on a hypervisor running on Commercial Off The Shelf (COTS) hardware called a virtual Customer Premise Equipment (vCPE). An ISP offering services via hardware devices can replace the equipment at the customer site with a single vCPE in a one time visit. All services that are required by the customer can then be managed and orchestrated using cloud orchestration tools. New services can be established and torn down on the vCPE remotely. High quality broadband links from the ISPs IDC to an vCPE at the customer premises, allows the ISP to maximise the benefit of investment by migrating services from physical devices to virtual functions.

NFV standardisation is managed under the NFV Industry Specification Group (ISG) within European Telecommunications Standards Institute (ETSI). The initial phase of work for the group involved the development of applicable frameworks and standards (ETSI, no date).

Since the inception of NFV, both SDN and NFV have been seen to complement each other without necessarily being inter-dependent. Having said that it is becoming increasingly evident that there is an inter-dependency and recent industry discussion of the term Software Defined NFV (SDNFV) (Pitt, 2016) underlines it. New NFV functional services which link multiple functions that may or may not be in the same location are being developed that require the flexibility and control of SDN in order to chain them.

2.5 The Internet eXchange Point

IXPs provide ISPs, Mobile Network Operators (MNO) and ICPs the facility to interconnect locally which has a positive effect on Key Performance Indicators (KPI) such as latency, hop count,

packet loss and jitter. They also reduce the number of external transit ISPs traversed by the traffic between the subscribers located on member networks (Di Lallo, 2015). The origins of IXPs can be found in Europe when Telia and Tele2 connected via the Swedish University Computer Network (SUNET) switch in 1992 (netnod, no date). North American IXPs evolved as successors to the four Network Access Points (NAP) that were mandated as part of the decommissioning of the National Science Foundation Network (NSFNET) in 1994/95 (Chatzis, Smaradgakis, *et al.*, 2013). The origins of each has had a long term effect on their development with North American IXPs becoming largely for profit with a focus on revenue generation while European IXPs retain a focus on mutual value through cost sharing between a community of members (rather that paying customers) (Norton, 2014). There have been a number of attempt to revive the relatively stagnant North American IXP marketplace by importing the European Open-IX Association (OIX) model to solve the lack of diversity in available peering opportunities created by the North American commercial model (Chatzis, Smaradgakis, *et al.*, 2013). Examples such at the LINX North Virginia (NoVA) in Washington D.C. demonstrate this (LINX, 2014). The open and co-operative nature of European IXPs means that there are hundreds of connected members at each and their peering switching infrastructures are handling aggregate traffic that is peaking at multiples of Tb/s with individual IXPs supporting peering fabrics with 50K-100K actively used public peering links. In stark contrast, the largest IXPs in North America are typically owned and operated by leading commercial IDCs and these companies can have less than 100K actively used public peering links across all their IXPs in North America. Cross connect pricing as a result in North America is typically six times the European prices (Chatzis *et al.*, 2015).

Considering the African context, the scale of IXP is much smaller than North America or Europe, with the notable exception of South Africa. For example the Internet Neutral Exchange Association (INEX) in Ireland with 127 members and traffic levels exceeding 200 Gb/s (INEX, 17 Sept 2019) (an IXP serving one of Europe's smallest states) is 40 times larger in traffic terms than the Uganda Internet eXchange Point (UIXP) which is currently one of East Africa's largest IXPs with 26 members and with traffic in the region of 5 Gb/s (UIXP, 17 Sept 2019). Yet the open co-

operative nature of the European model with a focus on mutual value, the sharing of costs with community oversight offering a convenient place for ICPs to locate their CDNs, as well as a location for local content, is obviously more attractive than the commercial profit oriented North American model (Chatzis, Smaradgakis, *et al.*, 2013). The European Internet Exchange Association (Euro-IX) has also proved very supportive to both African IXPs and the African IXP Association (AFIX). This is evident by their continued support of the African Peering and Internet Forum (AfPIF).

2.5.1 The structure of an IXP

The architecture of a typical IXP is illustrated in Figure 5. Connectivity between members is typically provided by a shared Open Systems Interconnection (OSI) Data Layer switched Local Access Network (LAN). To connect to the IXP each member must have a public ASN. Each IXP member either locates a router at the IXP facility or cross-connects to one over the IDC infrastructure hosting the IXP. These routers have a BGPv4 instance that can peer directly with the other member BGPv4 instances bi-laterally over the peering LAN (Chatzis, Smaradgakis, *et al.*, 2013). In some cases that is exactly what happens if the IXP is very small, or the level of traffic between two members warrants it. More typically the IXP establishes an RS in order to reduce the number of direct peering relationships on the switching fabric. The RS acts to reflect routes received from each member to the other members. This is not a traditional router function as it does not record routes learnt in the hosts routes table and it does not carry any traffic. It uses BGPv4 to exchange reachability information with IXP members and then passes the *NEXT HOP* attribute unmodified to the other IXP members. The RS therefore does not prepend its own AS number to the *AS PATH*. For all intents and purposes it appears as if IXP members are facilitated by the RS to peer with each other (Bakker *et al.*, 2016).

A typical implementation of an RS at IXPs is the use of the BIRD Internet Routing Daemon (BIRD). BIRD was developed as a college project at Faculty of Math and Physics, Charles University Prague. Currently it is developed and supported by CZ.NIC Labs (BIRD, no date).

A second specialist RS called a Route Collector Server (CS) is also often established at an IXP. This operates identically to the RS except it only imports routes, it does not export routes. This allows the IXP to insist that all members peer with the CS for statistical purposes while catering for members with a restrictive peering inclination who do not want to peer with the RS.

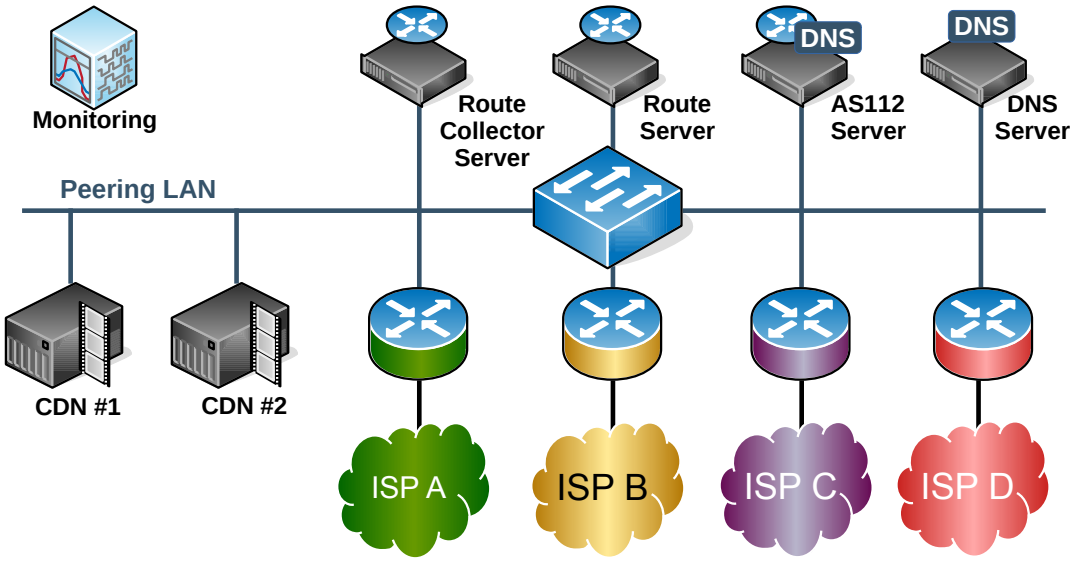


Figure 5: Structure of an IXP

Many IXPs also offer an AS112 Blackhole Service (BS). This is a service named after the ASN assigned to it. The *BS* is an attempt to deal with Domain Name System (DNS) *reverse lookup* queries for IP addresses that are part of the private address space. As these IP addresses have only local significance it is not possible for public DNS servers to respond to such queries with useful answers. The AS112 servers are deployed, mainly in IXPs, as an anycast distributed sink for such queries in order to reduce the load on the public authoritative servers (Abley and Sotomayor, 2015).

Some IXPs implement an open source IXP management system called *IXP Manager* which includes an administration and customer portal as well as end-to-end provisioning and graphing. IXP Manager is implemented by integrating the software onto an existing set of IXP functions such as the RS, CS and BS (*IXP Manager, no date)*. At UIXP, IXP Manager was deployed as a management system and while it does ease administration on a large IXP, it also complicates the troubleshooting of the underlying IXP.

Each IXP member agrees to general terms and conditions which will set out rules such as mandatory peering with the CS as well as rules associated with peering to the RS and the BS (Chatzis, Smaradgakis, *et al.*, 2013).

2.5.2 Peering Policies

Peering policies are an articulation of an operators peering inclination and the policy allows a prospective peering ISP a means of predicting the likely peering inclination before it attempts to peer (Norton, 2014).

Operators, whether they are ISPs, MNOs or ICPs, will adopt a strategy based on factors such as their traffic volume relative to the prospective peer, traffic ratios between the peers, geographic locations where the peers overlap as well as the competitive advantage that the new peer may gain or loose. Smaller ISPs for example will be inclined to make interconnection decisions not based on good networking practice but based on cost minimisation while larger providers are incentivised to sell interconnection to as many smaller networks as possible (DeNardis, 2012).

There are typically four levels of peering inclination:

- ***Open***: the ISP is open to peer with anyone and will typically have a peering relationship with the RS at the local IXP.
- ***Selective***: The ISP sets a baseline criteria. Other ISPs that exceed the baseline will be allowed peer. This is common with larger ICPs and large tier two ISPs who do not want to manage multiple peering relationships with many smaller ISPs. They consider that the smaller ISP are getting more from the relationship than they are and should be buying transit from them.
- ***Restrictive***: The ISP is not inclined towards peering and apart from their existing peers will not peer with new members except in exceptional circumstances. This is the typical inclination of tier one ISPs. They peer with other tier ones and all other peering relationships are Internet transit with lower tier ISPs from whom they collect revenue. Depending on the

size of the IXP and/or IDC they may have a presence there in order to peer with customers.

- ***No peering***: These ISPs tend not to be part of the IXP at all. This is an unusual inclination and includes companies who prefer to interact with the Internet via a transit provider ISP who removes the headache of networking from them, preferring instead to focus on generating content. Such ISPs are considered tier three ISPs.

There is a relationship between the peering inclinations of ISPs and size. Smaller ISPs with lower traffic volumes will be inclined towards an open policy, whereas ISPs with a large customer base and large traffic volumes are more likely to have an inclination towards a selective policy due to the fact that they have more to give away in terms of routes. To be fair to this group of ISPs, studies have shown that almost half of them still maintain open policies at ISPs and less than 8% adopt very restrictive policies (Lodhi *et al.*, 2014).

2.5.3 The changing nature of the IXP

The IXP market has adapted to address the challenges of changing traffic patterns by offering IXPs as carrier neutral node locations to host CDNs. A number of different models have emerged that are worth considering.

- ***Independent local IXPs***: In this model each IXP within the IXP group is independent in that the IXPs are not directly connected by the IXP organisation, (particularly at the OSI Data Link layer for the purpose of cross IXP peering or Private Virtual LANs (PVLAN)). It may well be however, as in the case of LINX and INEX, that the group of IXPs come under the same management (LINX, no date), (Gorey, 2016). The reason for the reluctance of LINX and INEX to link their IXPs is the potential hazard it would create if the very ISPs that made these IXPs successful would now consider the IXP as competition in the transit and back-haul space and potentially withdraw support from the eXchange project, the *IXP interconnection hazard*. By remaining neutral, with a focus on the provision of mutual value through the sharing of costs, while remaining out of the IP transit and back-haul markets, the IXPs continue to grow while staying focused

on the providing core IXP services.

- ***Interlinked IXPs using commercial links***: In this model the network of IXPs are interlinked by a number of third party ISPs. IXP customers can connect at one exchange and peer directly with customers of another IXP in the same group. This typically results in reduced OPerational EXpenditure (OPEX) to eXchange customers. However, the model does not exclude ISPs from participating in the business. DE-CIX in Germany is an example of this model (DE-CIX, no date).

- ***Interlinked IXPs using owned links***: This model creates a fully integrated IXP over an area with multiple Points of Presence (PoP) interconnected by the IXP owner. The investment required to build such a model is typically only feasible for national governments. The National ICT Broadband Backbone (NICTBB) in Tanzania is one such example (Pazi and Chatwin, 2013).

2.5.4 distributed IXPs

It has been seen in other jurisdictions that an increase in the number of IXPs helps to increase network speed and reduce costs. Brazil invested heavily between 2006 and 2014 and consequently the Brazilian IXP ecosystem (called IX.br) has grown from 4 IXPs to the current 25 in operation with 16 new locations under evaluation. IX.br follows a non-profit business model that facilitates multilateral agreements. (Brito *et al.*, 2016). The additional IXPs helped to decrease network latency and bypass international traffic fees (WEF, 2018).

An African example exists in South Africa, the NAPAfrica Internet eXchanges, located in Teraco IDCs in Cape Town, Durban and Johannesburg come under single management (McCann, 2018). However, mindful of the *IXP interconnection hazard* these IXPs maintain independence in terms of peering (*NAPAfrica*, no date).

In addition to NAPAfrica, South Africa has the Cape Town Internet Exchange (CINX), Durban Internet Exchange Point (DINX) and Johannesburg Internet Exchange Point (JINX) and between them they keep South African local traffic within South Africa. In contrast, local IXPs are

much less prevalent along paths between South Africa and other African countries (Gupta *et al.*, 2013), (Fanou, Francois and Aben, 2015).

The South African Internet ecosystem bears little resemblance to the rest of the continent, it resembles the European ecosystem. For example NAPAfrica IXPs alone have over 500 connected networks and CINX, DINX and JINX add over 200 additional networks. When compared to East Africa where the combined total of peers at active IXPs is just over 100 (Lodhi *et al.*, 2014), (*PeeringDB*, no date).

Another interesting concept is the RemIX distributed IXP for remote and rural networks. While this model appears similar to a traditional IXP it turns the model on its head and provides a community of small IXPs with a shared Internet transit via a larger ISP. This concept uses the IXP concept to support a community of small ISPs under a single ASN where each node is separated by the vast distances, across rural Scotland in this case. The members under RemIX form a confederation with a transit provider that presents the RemIX members collectively to upstream providers and other IXPs. The linking of member nodes to RemIX is agnostic but in the main consists of wireless links, many of which use the Short Range Device (SRD) 2.4 and 5 Ghz bands (Waites *et al.*, 2016). This form of distributed IXP could have a use in rural parts of East Africa though for the moment it appears that access is being left to the larger MNOs.

Like many developing countries, East African countries (with the exception of Tanzania) either have no IXP or a single IXP, generally linked to the capital city.

2.5.5 Software-defined IXPs

As far back as 2013 there was hope that SDN could form the basis for new inter-domain routing mechanisms by allowing the control plane of BGP to evolve independently from the underlying switch and router hardware and bringing software control and logic to inter-domain routing. SDN was seen as the solution to increasing scalability issues for the largest, mainly European, IXPs as well as a mechanism for increased functionalities at the RS such as advanced route filtering and auto-provisioning (Chatzis, Smaradgakis, *et al.*, 2013). There are underlying

problems with the Internet's routing system which are sourced in the operation of BGPv4. A SDX can offer the flexibility to address them as well as offering new innovations in inter-domain routing such as security and scalability. Google have demonstrated the viability of an SDX through Project Cardigan (Stringer *et al.*, 2014). An SDX prototype that allows participants to override default BGP routing behaviour with more fine-grained SDN policies has been developed (Gupta *et al.*, 2014) and from this project an Industrial-Scale SDX (iSDX) prototype that provides each participant AS with the abstraction of a dedicated switch that it can program using *match-action* policies to control traffic flows (Gupta *et al.*, 2016). In parallel to this work by Gupta et al, the Toulouse IXP (TouIX) implemented an SDX based on the Ryu SDN framework and Pica8 OpenFlow (OF) switches in a live environment and (apart from the Google prototype) it is considered the first live SDX example (Lapeyrade, Bruyère and Owezarski, 2016). This exchange was renamed to the Toulouse SDN Internet eXchange (TouSIX) to reflect the change. Further work to separate the OF control-channel from the data plane has been implemented at TouSIX to ensure that data plane issues cannot affect control plane messages which could lead to a slow or unresponsive control plane (Bruyere *et al.*, 2018). The issue of iSDXs has been further explored through the E.U. Horizon 2020 funded, ENDEAVOUR project. This project funds work on distributed SDN control planes and SDN programming abstractions for large scale IXPs. By considering the current design at largest IXPs such as the European big three AMS-IX, LINX and DE-CIX the project has aimed to addressed problems rooted in multi switch, tiered, core-edge network design. Problems such as advanced black-holing to detecting and prevent Distributed Denial of Service (DDoS) attacks (Chiesa *et al.*, 2016) as well as the reduction of the number of policies at edge switches to outbound policies only forwarding state duplication is avoided as well as the problems associated with failover recovery in a multi-hop IXP environments to which solutions have been proposed through the duplication of outbound policies, bouncing packets back to the source ingress switch and the injection of recovery information in the packets (Antichi *et al.*, 2017).

In the future it may be possible to see an overhaul of Internet routing and the operation of IXPs using SDN. As SDX deployments evolve and begin to take hold, new approaches to inter-

domain policies have the potential to offer more secure and flexible traffic engineering techniques, including the delivery of Application-Layer Traffic Optimisation (ALTO) information services (Alimi *et al.*, 2014) (Brito *et al.*, 2016).

As well as considering the incorporation of SDN into existing IXP architectures with BGPv4 at the core, there has also been consideration given to the design, implementation, and operation of SDXs that enable the interconnection of SDN islands in future scenarios where SDN has replaced the traditional AS by organisations such as the International Center for Advanced Internet Research (iCAIR). Such solutions will require the development of new techniques to extend the single domain orientation of the SDN approach to incorporate multi-domain control, signalling, and dynamic provisioning. Testbed examples such as the StarLight SDX prototype of iCAIR and the large scale distributed Advanced Networking Research Facility OF based testbed from the international Global Environment for Network Innovations (iGENI) has undertaken multiple SDN experiments to investigate these concepts (Mambretti, Chen and Yeh, 2014b). SDX can enable many dynamic provisioning options, the faster implementation of new and enhanced services, enabling new applications, functions to control core network resources and improved options for customisation of networks. All these enhancements can improve operational effectiveness and efficiency (Mambretti, Chen and Yeh, 2014a).

SDX presents the potential for a much enhanced Internet core; however, for this evolution to happen there is a requirement for the overhaul of BGPv4 as the core routing protocol upon which the complete Internet routing infrastructure relies on today. It may well be some time and involve a long transition before such fundamental change is realised in inter-domain routing.

2.6 The economy of the Internet

> *The effect of a telecommunications network is proportional to the square of the number of connected users of the system. Metcalfe's n^2 Law (Mendelsohn, 2016)*

The speed of Internet development has been phenomenal, the exponential rate of this development, initially fuelled by cheap modems and personal computers which has been followed

by the addition of smartphones and now *things* has seen the Internet pass rapidly through different stages of development. To-date this development has been funded by the carriers and ISPs through the profits from their increasing subscriber base. This Internet development in the main has not attracted public government funding globally, as has been the case with other infrastructure such as transport, water, sanitation and electrical. However, as mobile saturation levels are reached and Average Revenue Per User (ARPU) falls the question of whether this can continue is debatable.

The GSM Association (GSMA) considers that the ongoing business case to deliver mobile services for MNOs has become weak because they have only a small share in the value of the projected growth. They also project that demand will require MNOs to at least double or even triple their capital expenditure (GSMA, 2018a). This problem is exacerbated in dense urban cities where 5G requires smaller femtocell and picocell sizes to deliver the expected level of service as macro cells reach their geographic saturation point earlier. Such deployments are restricted owing to difficulty in obtaining access to buildings and rights of way for fibre and power. Such a level of expenditure is not feasible in the face of saturated networks and falling ARPU.

Even in fixed Internet, carriers, cable operators and ISPs have all but exhausted the capabilities of copper and co-axial with Very high bit rate Digital Subscriber Line version 2 (VDSL2) reaching 100 Mb/s. To deliver Internet maturity level five it has become necessary to replace the existing copper and co-axial networks with fibre. The cost of installing FTTH is prohibitive particularly the last piece from the kerb to the premises. For example, take fibre from a city node to a housing estate Primary Cross-connection Point (PCP) as being approximately 5 Km long and if there are 100 houses in the estate that are on average 300 m from the PCP, in this case 30 Km of fibre-optic cable is required in the estate alone, a ratio of 1:6. A town of 100,000 people with an average of four people per house would require 7,500 Km of fibre-optic cable for the last piece. Traditional carriers have an advantage as they tend to have the duct network in place from the copper era, however, these are not always fit for purpose and require significant civil works. The costs associated with delivery means that companies cherry-pick areas of profitability, typically dense urban areas where a Return on Investment (RoI) can be realised soonest. It has also led to

operators considering models such as co-operation at the physical layer while they compete at the data-link and higher layers as well as fully open access networks (Verbrugge *et al.*, 2011).

When the problem is considered from a wider economic perspective, it can be shown that the return to society in the form of GDP exceeds the infrastructure capital expense in as little as 7 to 18 months (WEF, 2018). Taking Ireland as an example, using E.U. data it can be estimated that a €2.4 billion investment to enable national maturity level three would lead to an uplift in GDP of €3.4 - €9 billion. This would result in a payback in less than one year. Unfortunately within the E.U. such investment by member countries has been considered state aid and therefore prohibited least it distortion competition in the market. However, the E.U. has recognised the problem and realise that the sector has significantly evolved. There is a realisation that both consumers and businesses increasingly rely on data and Internet access services and as a result the E.U. established an Electronic Communication Code (EU, 2018) at the end of 2018 to address the challenge.

In general, governments and banks have avoided investment in Internet infrastructure related projects mainly due to concerns of competition in the market and uncertainty with regard to technological development. Internet development is considered by many private investors as overly complex and as such Internet development is seen as private sector traditional carrier, ISP and MNO activity instead of necessary public infrastructure. The rapid change in the technological landscape also marked it as a different a class for investment from the longer term transport, water, sanitation and electrical projects (WEF, 2018).

Given that the Internet is a significant driver of economic growth, it should certainly be considered a candidate for public investment. The Total Societal Impact (TSI) of such investment is an important measure as it demonstrates the economic, social, and environmental impact (Eccles, 2017).

2.7 Connecting East Africa to the Global Internet

Apart from South Africa which was connected to the Internet in 1993 (Huurdeman, 2003) (Jagun, 2008) the rest of the continent was connected to the Internet via low capacity satellite links.

Connection to the global Internet via fibre-optic submarine cable started with North and West Africa connected via the South Atlantic 3 (SAT-3)/West Africa Submarine Cable (WASC) in 2002 (Jagun, 2008) and East Africa which was connected in 2009 via SEACOM (SEACOM, 2010), Eastern Africa Submarine cable System (EASSy) (Muller, 2018) and The East African Marine System (TEAMS) cables (TEAMS, no date). This landing of submarine fibre-optic cables at the coast triggered a flurry of activity within the region to develop terrestrial fibre-optic links as well as inter-governmental initiatives such as the Northern Corridor Integration Projects (NCIP) between Kenya, Uganda, Rwanda and South Sudan which aims to develop a regional strategy for greater Internet penetration and co-operation (NCIP, 2016).

2.7.1 The root of transformation in the East African Internet

Mobile back-haul and Internet access in East Africa up to the landing of the submarine cables on the coast in 2009 consisted of Very Small Aperture Terminal (VSAT) C-band or K_U-band satellite connections. In the case of Internet access, ISPs in East Africa accessed service via VSATs whereas in the case of 2G Global System for Mobile communications (GSM), Base Station Controllers (BSC) at centralised sites connected via a centralised satellite hub site to remote Base Transceiver Stations (BTS) via VSATs.

The region was not only the last in Africa but the last in the world to be linked via submarine fibre-optic. South Africa had been connected as early as 1993 for data via the South Atlantic Telephone 2 (SAT-2) linking Melkbosstrand to Madeira which was connected to Portugal (ANACOM, no date). It was followed in 2002 by a 340 Gbit/s sister cable called SAT-3/WASC which linked Sesimbra to Melkbosstrand following a route along the West African coast that included landing sites in the West African countries of Senegal, Côte d'Ivoire, Ghana, Benin, Nigeria, Cameroon, Gabon and Angola (Jagun, 2008). This cable was supplemented in 2012 with an additional 5.12 Tb/s cable that facilitated the decommissioning of the SAT-2 cable as well as developing new West African landing sites at the Canary Islands, Cape Verde, Togo, the Republic of Congo (Brazzaville), the Democratic Republic of Congo (DRC) and Namibia.

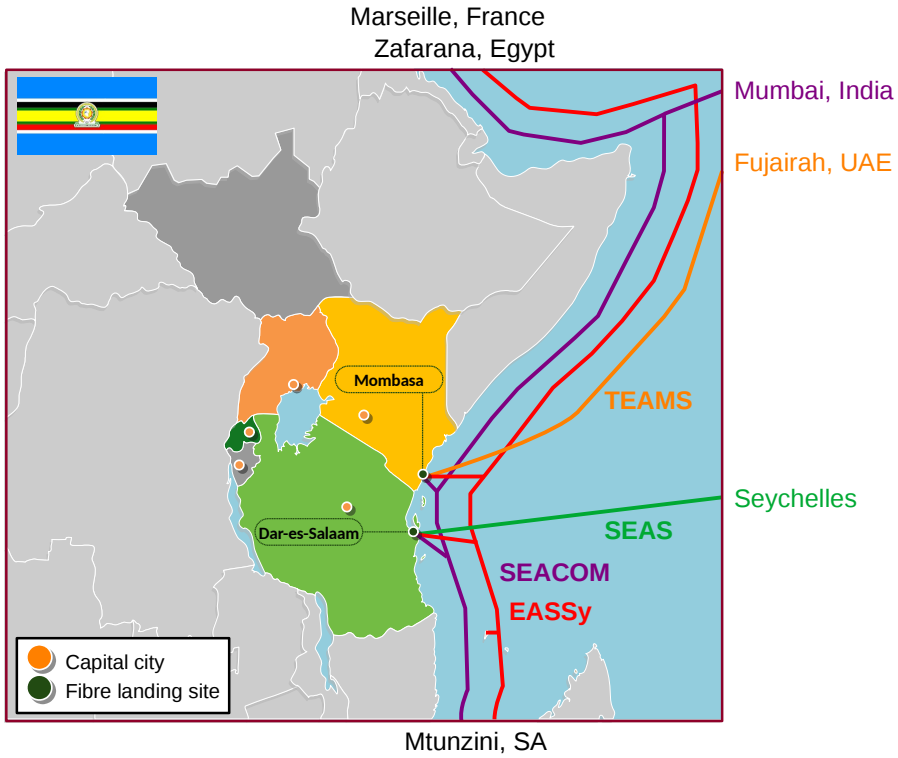


Figure 6: East African submarine fibre-optic Internet connectivity

As illustrated in Figure 6, East Africa was finally connected when the SEACOM launched their 1.28 Tb/s submarine fibre-optic cable in 2009 which has since been increased to 12 Tb/s. It connects Mtunzini in South Africa to Marseille in France with landing points in Mozambique, Dar-es-Salaam in Tanzania, Mombasa in Kenya, Djibouti and includes a spur to Mumbai in India (SEACOM, 2010).

The Kenyan Government also supported a project called The East African Marine System (TEAMS) to link Mombasa to the United Arab Emirates (UAE) with a 1.2 Tb/s cable which was also launched in 2009. This has provided the East African region with more resilience and competition in order to drive down costs (TEAMS, no date).

The Eastern Africa Submarine cable System (EASSy) quickly followed in 2010 also connecting Mtunzini with Marseille and included landing points in Mozambique, Madagascar, Dar-es-Salaam in Tanzania, Comoros Islands, Mombasa in Kenya, Somalia and Djibouti providing redundancy and competition. As a result the region's access prices began to fall (Muller, 2018).

2.7.2 The barriers to Internet Growth

As countries grapple with the challenges of increasing Internet penetration and push for full access, it is necessary to identify the barriers to Internet penetration growth. There are four identified barriers that must be overcome by countries to push towards full Internet penetration (WEF, 2016):

- Infrastructure,
- Affordability,
- Digital Skills and awareness,
- Relevant content.

The issue of cost to a country is an important one. As Internet access and penetration has a direct impact on GDP, as well as, offering new avenues to education, business opportunities and communication it is a cost effective investment for countries to push for increased Internet penetration.

The World Economic Forum (WEF) carried out a costing exercise of the NCIP countries (WEF, 2017). The exercise considered interventions that could overcome the barriers and significantly increase Internet penetration across the northern corridor as outlined in Table 2.

Table 2: Identified interventions for NCIP "Internet for All" model

Barrier	Measure	Result	Interventions
Infrastructure	Mobile maturity level 2 or higher	58%	Increase 3G & 4G coverage
Affordability	Smartphone adoption	10% - 29%	Increase access to smartphones
Skills and Awareness	Youth in education	< 50%	Train two people per family in digital skills, train 10% with advanced skills
Content	Internet domains per 1000 people	0.1	Develop tech park to develop local content

The next phase of the exercise as demonstrated in Table 3 costed those interventions at €1.6 billion, a wholly unsustainable figure given the economic context of the countries involved.

However, by considering some further interventions the cost can be reduced by as much as

23% through infrastructure sharing by operators and the development of sensible spectrum policies by regulators freeing up lower frequency bands which increase coverage, particularly in rural underserved communities as well as through the reduction or removal of taxes (WEF, 2017).

Table 3: Costed interventions for NCIP "Internet for All" model

Barrier	Intervention	Cost (millions)	Further interventions	Costs (millions)
Infrastructure	Increase 3G & 4G coverage	€375	Infrastructure sharing, spectrum management	€100
Affordability	Increase access to smartphones	€535	Remove VAT	€450
Skills and Awareness	Train 2 people per family in digital skills, train 10% with advanced skills	€645		€645
Content	Develop tech park to develop local content	€45		€45
		€1,600		**€1,240**

An interesting finding from the costing model was the fact that infrastructure was not the biggest cost, in fact skills awareness training makes up the largest cost in developing countries. The second largest cost is affordability of access devices such as smartphones, making up 33% of the overall cost. One way that government can intervene to reduce cost is by removing Value Added Tax (VAT) and other charges. The government of Kenya did that and removed cumulatively 21% of handset costs, they found that handset sales doubled during the period of the experiment and overall national penetration jumped from 50 to 70% (WEF, 2018). Interventions to increase smartphone penetration need not all be from government. Enterprises and Non-Governmental Organisations (NGO) whose businesses benefit from higher penetration can also assist in novel ways. For example Farm Africa U.K. based charity that works with farmers, pastoralists and forest communities in East Africa. In Uganda, Farm Africa realised that communications via a smartphone application to their team of trainers was the optimal way to manage training delivery. The challenge was that many of the trainers were not smartphone owners so they provided each trainer with a smartphone for an up front payment of UGx 30,000 (€7) and held back a small percentage of their fee over each training until the full cost was realised. In this way each trainer retained the smartphone even after the

completion of their relationship with Farm Africa.

2.7.3 Connecting the region

Immediately after the landing of the submarine fibre-optic cables in 2009 there was a growing push to connect inland to the non-coastal East African cities. Kampala, Uganda connected via Mombasa, Kenya, Kigali, Rwanda connected via both Kampala as well as Dar-es-Salaam in Tanzania, Kampala, Uganda was connected to the SEACOM cable via a spur, installed on the 23 July 2009 (Mugabe, 2009).

Liquid Telecom, a subsidiary of Econet Global, was originally a Satellite operator called Econet Satellite Services, realised the impact that the impending SEACOM and EASSy cables landing in East Africa would have on their satellite business, so in 2004 they rebranded as Liquid Telecom and entered the cross-border fibre-optic market. They were perfectly placed to build their business when the cables landed in Mombasa and Dar-es-Salaam. This was transformational in terms of the regional broadband, providing cross-border competition services to inland countries such as Uganda, Rwanda and Burundi as demonstrated in Figure 7.

As well as private sector investment, national governments began to realise the importance of fibre-optic development on their economies now that the region was connected globally via the submarine fibre-optic cables.

Kenya built a National Optic Fibre Backbone (NOFBI) linking all 47 counties and included Metropolitan Area Networks (MAN) in 35 of the counties. Resilient links have been added, as well as, cross border connectivity to South Sudan (NOFBI, no date).

The Uganda, the government has also built a National Backbone Infrastructure (NBI) (UBIST, 2009) (Ssempebwa and Lubuulwa, 2011) to link the rest of the country to the international connectivity now in place in Kampala (Government of Uganda, 2015). As well as linking the country the NBI has connections to Kenya, Tanzania, and the DRC. This however has had some technical (Bulega *et al.*, 2011) and political (techjaja, 2017) challenges along the way (Walubiri, 2018) and today utilisation of the infrastructure is only at 30% (Government of Uganda, 2018).

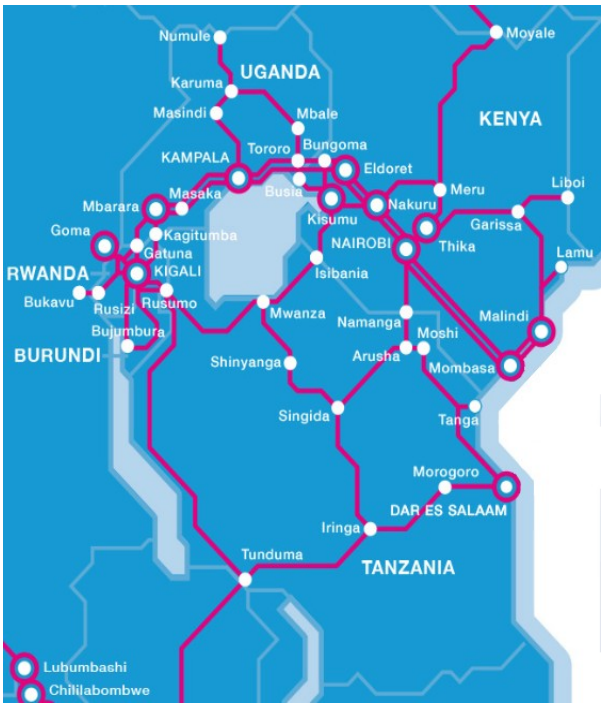


Figure 7: Liquid East African fibre-optic network (Liquid Telecom, 2019)

In Rwanda, the National Information Communication Infrastructure (NICI) has developed a national fibre-optic backbone spanning the districts and border posts and the Kigali MAN was developed to interconnect government institutions around a national IDCe. Rwanda developed 2.5 Gb/s of capacity to the SEACOM and TEAMS submarine cables through both Uganda and Tanzania (Government of Rwanda, 2013).

In Tanzania the need for improved ICT infrastructure was identified in 2003 (Government of Tanzania, 2003) and was actioned via the building of the NICTBB with funding from China, a terrestrial fibre-optic backbone of 7,500 Km that interconnected the urban cities and towns of Tanzania with the submarine fibre-optic cable landing stations of the SEACOM, EASSy, Seychelles to East Africa System (SEAS) submarine fibre-optic cables in Dar-es-Salaam. The project includes cross-border connections to Kenya, Uganda, Rwanda, Burundi, DRC, Zambia, Malawi, and Mozambique (Sedoyeka and Sicilima, 2016). Having built the NICTBB there is now an imperative to recover the investment which will have an impact on the wholesale rates offered to operators (Pazi and Chatwin, 2013) (Esselaar and Adam, 2014). Through regulation the government has directed that all IXPs must be directly connected to one another and that all service providers must

connect to their nearest located IXP (Government of Tanzania, 2011). The downside of this of course is that there is little incentive for commercial providers to build parallel competing networks that could drive down prices.

In 2016 Burundi, with funding from the World Bank, developed the Burundi Backbone System (BBS), a fibre-optic communication network that is connected to the submarine cable landing stations in Mombasa. This network will lease connections to operators, companies and the government (WEF, 2018).

However, given all this work and according to analysis in the recent Uganda National Broadband Plan 2018 (Government of Uganda, 2018) the cost of bandwidth from the east coast of Africa to Uganda is still more expensive than from the east coast of Africa to Europe or the U.S.

2.7.4 Internet penetration in East Africa

Despite East Africa being the last region of the world to gain access to the Internet via submarine fibre-optic cables in 2009, it played catchup in the last decade as illustrated in Figure 8. In fact Kenya now has the highest rate of Internet penetration on the continent at 85%, surpassing even South Africa at 54% despite the fact that it has had a submarine fibre-optic cable since 1993.

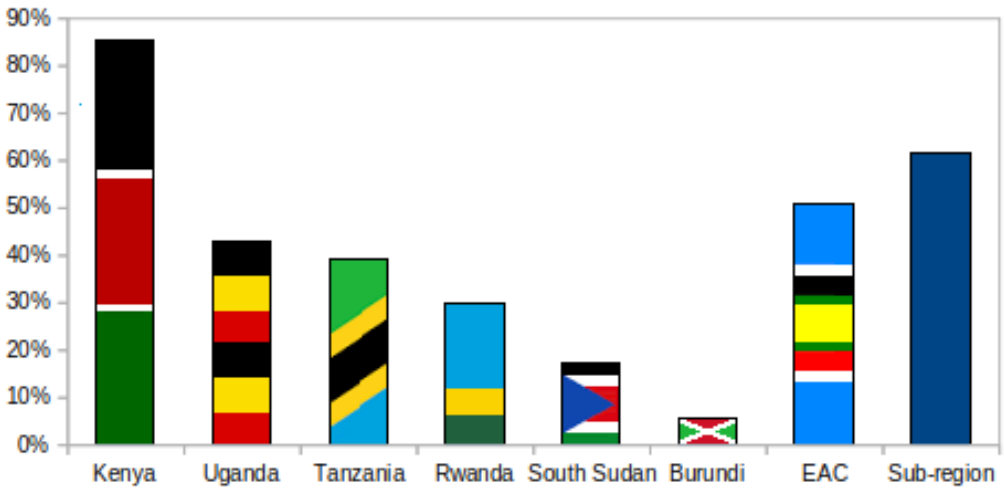


Figure 8: Internet penetration in East Africa

Uganda at 44% has been preforming well, in particular considering it is an inland country, though this has dipped since the introduction of the Over The Top (OTT) tax in 2018 (Kasemire,

2019). Tanzania is slightly behind with 39% despite the fact that there are submarine fibre-optic cables landed directly in Dar-es-Salaam (Walubiri, 2018). South Sudan and Burundi have the lowest penetration rates in East Africa. Considering that South Sudan is the world's youngest country having received independence in 2011, it has been war-torn for most of the time since and therefore Internet penetration has not been a priority. According to the World Bank, Burundi has the lowest GDP per capita of any country in the world at €280 (WBG, 2017) and not surprisingly there is a clear link between countries with low development indicators, such as life expectancy, education and income equality and low Internet penetration (Walubiri, 2018).

Figure 9 compares Internet penetration among the countries in the region with the rest of the world and while it demonstrates that the region is trailing significantly, it is well ahead of the overall African continent Internet penetration figures. Also when the sub-region is extracted it demonstrates an Internet penetration level of 53% that is very slightly lower than the world average Internet penetration figure of 54%. This is somewhat positively skewed for Uganda and Rwanda by the very high penetration figure enjoyed by Kenya.

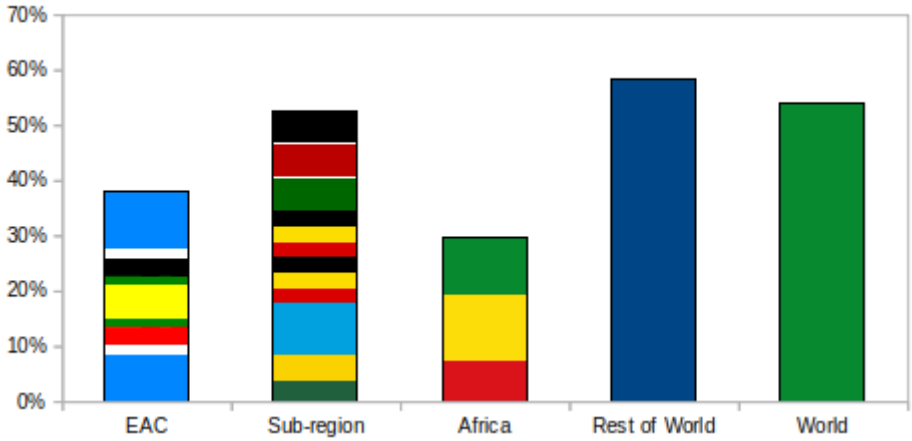


Figure 9: East Africa penetration by comparison

2.7.5 Data Centres

As well as the delay in getting access to the Internet via submarine fibre-optic cables, East Africa has been very slow to build IDC infrastructure. Europe and North America have lead the way in IDC development. An IDC with a floor space capacity of 2,000m^2 is considered small and larger IDCs typically have floor space between 20,000 and 30,000m^2. Within the sub-region there are currently two IDCs, the East Africa Data Centre (EADC) (EADC, no date) in Nairobi is the largest at 2,000m^2 and is certified at Level-III but is owned by Liquid Telecom so it cannot be considered carrier neutral, also in Kenya is the the icolo owned Mombasa One (MBA1) with a floor space of 450m² (icolo.io, no date). icolo are currently building a second IDC in Nairobi as a sister to the Mombasa centre and it is planned to have approximately the same floor space of 450m². To-date both Uganda and Rwanda have no carrier neutral IDCs; however, Raxio, a Ugandan company, owned by the U.S. based Roha Group, has committed to building a 2,500m^2 carrier neutral IDC on the outskirts of Kampala, scheduled to open in 2019 (Raxio, 2018) (but more likely to open in 2020 at this stage). In Tanzania, as part of the NICTBB a tier III government owned IDC was built in the Kinondoni District of Dar-es-Salaam. Like the rest of the NICTBB it is operated by TTCL. Disaster recovery centres are also being added in Dodoma and Zanzibar (Smolaks, 2017).

2.7.6 Transit

Since the submarine cables landed in East Africa there has been a sharp rise in bandwidth demand with a corresponding fall in cost per Mb/s. This is fuelled by the global trends discussed in section 2.3.1 and rising demand as illustrated in Figure 10 for Uganda.

Figure 11 demonstrates the relationship between the price per Mb/s and the bandwidth volume acquired by a medium sized ISP in Kampala with a focus on Business to Business (B2B). Prices have fallen from €1,900 per Mb/s over satellite in 2008 to €7 per Mb/s today. Immediately after the installation of the submarine fibre-optic cables they were able to acquire a pair of *dark* fibre-optic cores from Mombasa via UETCL for €17,500 per month. On top of this they paid €520 per Mb/s for a Fast Ethernet (FE) 100 Mb/s link in 2009.

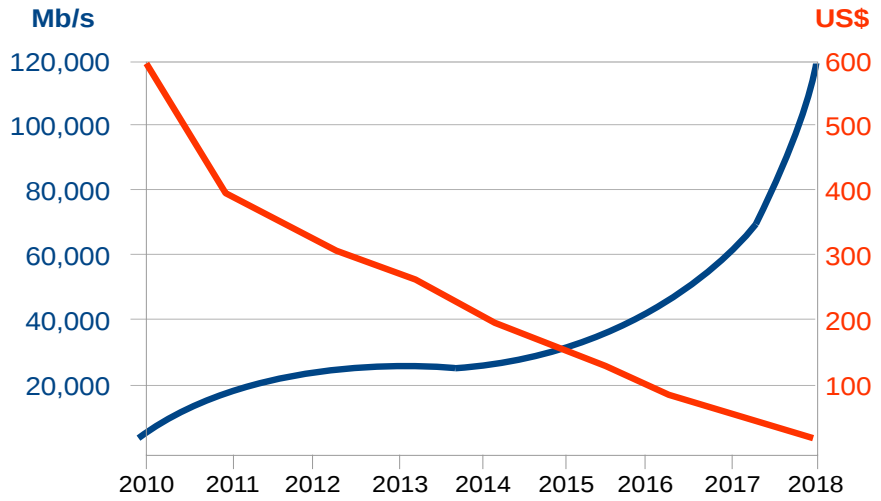


Figure 10: Uganda total bandwidth vs Price per Mb/s (UCC, no date)

(extracted from UCC market reports and ISP B/W pricing information)

Relatively quickly the Bandwidth & Cloud Services (BCS) Group started offering bandwidth directly in Kampala. This removed the need for fibre-optic core rental as well as negotiating with bandwidth providers in Mombasa. By 2010 they were able to acquire bandwidth at €350 per Mb/s. Over the remainder of the decade, more competition, as well as the general principle of transit prices falling year on year, has seen the price fall steadily.

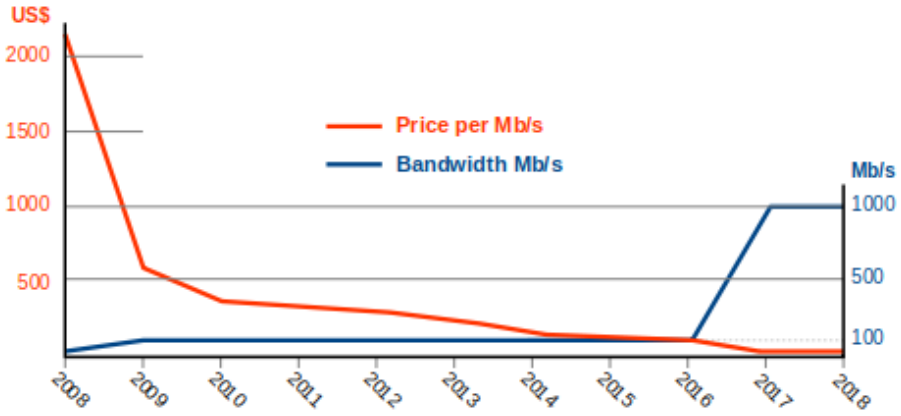


Figure 11: Internet transit vs Growth trend for Ugandan ISP
(Figures obtained from Ugandan ISP, 2018)

2.7.7 Peering

According to CloudFlare transit prices in Africa (Rao, 2016) are amongst the highest in the world at 14 times the global benchmark, with notable variance across the continent, from Cairo to Mombasa to Johannesburg. Fortunately, of the traffic that CloudFlare currently serve locally in Africa they peer about 90% (with a mix of carriers and ISPs).

Despite the rapid installation of terrestrial connectivity to meet the submarine fibre-optic cables, the paths taken by traffic between different African countries demonstrates that extreme tromboning is still a problem. Table 4 demonstrates this trend. Estimate Round Trip Times (RTT) were calculated by selecting the most direct road route (as given by Google Maps) as fibre paths generally follow road development. In the case of Mozambique to Madagascar an estimate was made using a direct sea path to Toliara on the west coast of Madagascar, plus the road distance to the capital Antananarivo.

Having the distance, the RTT was calculated as follows:

Given: *Speed of light in a vacuum (c) = 299,792,458 m/s*

Effective Group Index of Refraction (N eff) = 1.4682

Speed of light in a fibre (c_f) = c / N eff = 299,792,458 / 1.4682 = 204,190,476 m/s.

Round-Trip Time (RTT) (ms) = $\frac{Distance\,(km)\,x\,1000\,x\,2}{c_f}$ (Corning, 2014)

Traceroute tests were taken between a number of capital cities in Africa and compared to the estimates (Roberts, 2018). The traceroute tests give the path taken, in all these cases the paths took a circuitous route via Europe which accounts for the difference figure.

It would be expected that those countries with a direct physical connections would enjoy lower RTTs, however, this is not necessarily true in Africa (Formoso *et al.*, 2018). A 2017 investigation into the possibility of interconnecting the African IXPs by creating a distributed IXP layout spanning the continent (Fanou *et al.*, 2017).

Table 4: Tracepaths in Africa 2018

From/to	Direct distance (Km)	Estimated RTT (ms)	Tested RTT (ms)	Difference (ms)	Path taken
Cape town Casablanca	11,266	110	192	82	South Africa, Portugal, UK, Germany, France, Spain, Morocco
Cape town Cairo	10,059	99	209	110	South Africa, UK, Italy, Egypt
Djibouti Douala	5,665	55	202	147	Djibouti, Italy, France, UK, Nigeria, Cameroon
Lusaka Lagos	5,650	55	299	244	Zambia, South Africa, France, Netherlands, France, Spain, Portugal, Nigeria
Mombasa Kinshasa	3,948	39	296	257	Kenya, France, UK, Congo
Maputo Antananarivo	1,710	17	589	589	Mozambique, South Africa, Portugal, UK, France, Madagascar

The aim of the investigation was to estimate the best case benefits that could be realised in terms of traffic localisation and subsequent improved performance within Africa. The investigation hypothesised that by connecting the IXPs in the various African countries, traffic would remain on the continent through an hierarchical IXP substrate of ISPs connected to local IXPs interconnected via regional IXPs. Simulations demonstrated that continental intra-African paths would double which would reduce length and drastically decrease RTTs. They also estimated that to build this continental IXP would cost between between €65 million and €1.5 billion. They did acknowledge that this proposal is not necessarily realisable due to prevailing external factors such as political instability.

Considering the member states of the E.U., where a set of national IXPs are under the control of the same organisation (with the exception of PoPs in the same city) they are generally not linked in peering terms. Members connected at one IXP in the set cannot peer directly with another member connected to a different IXP in the set through the IXP infrastructure.

In Ireland for example, INEX has IXPs in Dublin and Cork but they are not connected. In the U.K., LINX have IXPs in Cardiff, Edinburgh, Manchester as well as London and they are not connected. The reason is the *IXP interconnection hazard*. Connecting IXPs places the IXP in

competition with the ISPs who they rely on to make up their membership.

The function of IXPs has not changed fundamentally since Telia and Tele2 connected via the Swedish University Computer Network (SUNET) switch in 1992. They have evolved somewhat, in that CDNs have located (PWC, 2018) themselves at IXPs in many cases driving up traffic levels but that fundamentally supports the notion of *Local is Local*, low latency and lower costs for members and better services to end-users.

IXPs are not the solution to African continental routing issues which are a problem that can best be resolved by ISPs managing their peering relationships and national regulators making it easier for ISPs to connect across borders. For example in East Africa a Kenyan based ISP which is licensed in Kenya could have a simplified process to get an operations license in other East African countries.

2.7.8 Internet eXchange Points

There are a number of IXPs in East Africa. The Technology Service Providers of Kenya (TESPOK) launched the first first carrier-neutral, non-profit IXP in the region in November 2000 called Kenya IXP (KIXP) but it was shut down by the government regulator after two weeks acting on a complaint received from the then incumbent operator Telecom Kenya. However, during the time the IXP was running it was shown to reduce latency by a factor of 20 and bandwidth by a factor of 16. This was due to traffic tromboning via Europe without the IXP. The Communications Commission of Kenya eventually issued a license on the basis that it was a peering facility rather than an international gateway and KIXP went live on the 14 February 2001 (Souter and Kerretts-Makau, 2012). TESPOK also entered a partnership with the Amsterdam Internet eXchange Point (AMS-IX) in 2013 (AMS-IX, 2013) to develop an East African regional Internet IXP at the cable landing site at Mombasa (MSIXP). The partnership was short lived however, the following year AMS-IX withdrew from the partnership (Mang'unyi, 2015) citing the lack of traffic through the exchange as the reason. TESPOK continue to maintain MSIXP in Telephone House Data Centre (THDC) (Jumbe, 2016).

UIXP was established in Kampala, Uganda in 2001 with the support of the UCC. Today, the eXchange has 27 members and traffic peaks at 8 Gb/s. In 2015/16 UIXP went through a major upgrade, where it went from one to four cabinets, a core cabinet housing the core switching, virtualised server infrastructure and Akamai, the first CDN to come on board, a Google Global Cache (GGC) and two access cabinets to house member equipment (O'Briain *et al.*, 2017). Traffic peaked beyond 2 Gb/s for the first time in January 2017.

The Tanzanian IXP (TIX) was started by the Tanzania Internet Service provider Association (TISPA) in September 2003 and housed in the Posta Building in Dar-es-Salaam. It quickly established itself and by 2009 had 22 peers by the time the NICTBB was in place. The NICTBB included IXPs at PoPs along the terrestrial fibre-optic infrastructure and the government, through the Electronic and Postal Communications Act, CAP 206 (Government of Tanzania, 2011) directed that all IXPs shall be directly connected to one another. Any ICP, ISP or MNO is required to connect to the nearest located IXP. Today, there are four other IXPs active in addition to the original TIX, in Mwanza, Arusha, Dodoma and the semi-autonomous island of Zanzibar. However, the traffic levels are so small at these IXPs that it is obvious operators are connecting to comply with the act and associated regulations but are not exchanging traffic.

The need for an IXP in Rwanda was identified in 2002 and the Rwanda Information Technology Authority (RITA) with support from the Swedish government launched the Rwanda INternet EXchange (RINEX) in 2004. In 2009 its management came under the control of Rwanda Internet Community and Technology Alliance (RICTA) the organisation managing the Rwanda .RW country code Top Level Domain (ccTLD) (RURA, 2009) (EACO, 2014). Today it supports 15 members (RICTA, 2017).

The final and newest IXP in the region is the Burundi IXP (BDIXP) in Bujumbura. It was establish in 2017 according to the IXP website and has 10 connected members with 450 Mb/s of traffic.

Table 5: East Africa Internet eXchange Points

eXchange Point	Location	Peers	Annual Traffic Average		Annual Growth
			2017 *	2018 ┤	
Kenya (KIXP)	Nairobi, Kenya	27	4.9 Gb/s	6.76 Gb/s ┴	1.38 ┴
Tanzania (TIX)	Dar-es-Salaam, Tanzania	33	3.52 Gb/s	4.58 Gb/s	1.3
Uganda (UIXP)	Kampala, Uganda	24	1.4 Gb/s	2.23 Gb/s	1.6
Rwanda (RINEX)	Kigali, Rwanda	15	1.65 Gb/s	2.07 Gb/s	1.25
Burundi (BDIXP)	Bujumbura, Burundi	10	450 Mb/s ┼	1.5 Gb/s ┼	2.3
Mwanza (MIXP)	Mwanza, Tanzania	3	386 kb/s ┴	533 kb/s	1.38 ┴
Zanzibar (ZIXP)	Zanzibar, Tanzania	3		326 kb/s	
Arusha (AIXP)	Arusha City, Tanzania	12	129 kb/s ┴	179 kb/s	1.38 ┴
Dodoma (DIXP)	Dodoma, Tanzania	3		144 kb/s	
Mombasa (MSIXP)	Mombasa, Kenya	7			

* Figures obtained from EACO report (EACO, 2017).

┤ Figures obtained at the various IXP statistics portals.

┴ Average growth figures for TIX/UIXP/RINEX used to extrapolate figures indicated (excl BDIXP).

┼ Figure obtained from the BDIXP website (no statistics portal available).

Table 5 lists the existing IXPs in East Africa. There appears to be many; however, only five of these IXPs carry traffic over 1 Gb/s. The larger IXPs in the region house various CDNs for ICPs to keep international traffic local and thereby reducing demand on international transit as well as transit costs.

As part of an overall trend towards more control of digital and social media platforms by regional governments, IXPs have come under scrutiny too. The UCC proposed a framework (UCC, 2019) to create a National Designated IXP and link any new IXPs in the country to it, effectively creating one distributed national IXP similar to that in place in Tanzania. In Tanzania pressure has been exerted on TIX to move to a TTCL site in Dar-es-Salaam and even in Kenya where Asteroid have tried to establish an IXP in the iColo IDC in Mombasa they have found a reluctance from the Communications Authority to issue a license (Asteroid, no date).

2.7.9 Content provision

Global multinational ICPs have a weak presence on the continent due to the relatively few

IDCs available. Akamai, for example, has no location PoP on the African content, Cloudflare has eight PoPs on the continent but only one in East Africa (Cloudflare, 2015). Netflix no longer supports their own IDCs and relies on Amazon Web Services (AWS). AWS currently do not have an IDC on the African continent which means that Netflix has little penetration in this market. AWS are planning one in Cape town in the near future (AWS, no date). Facebook, Microsoft, and Google have no IDC presence in Africa, though Microsoft have plans for two in South Africa.

Despite this, many of these providers have CDNs located at both ISPs and IXPs in the region. For example, in East Africa, Akamai has at least five CDNs in Kenya as well as CDNs in Uganda, Rwanda and Tanzania. Facebook has node appliances in Kenya, Uganda, Rwanda and Burundi. Netflix has CDNs in Kenya and Rwanda and with the uptake of residential fibre-optic services in the larger cities there is more interest from them in the distribution of video content in the region (EACO, 2017). Google has edge PoPs in Mombasa, Kenya and Kampala, Uganda as well as Google Global Caches (GGC) located in Dar-es-Salaam, Tanzania, Nairobi, Kenya, Kigali, Rwanda to supplement the PoPs in Mombasa and Kampala (Google, no date).

Local Content provision has been weak in the region and the majority of content available is provided by international ICPs. Even local companies with content tend to host it outside the region, mainly due to price. Local content, particularly content in local languages, can attract a new members to the Internet and the vast majority of the relatively small volume of local content that does exist within the region is of Kenyan origin, services such as the SportsPesa online betting platform (*SportsPesa*, no date) as well as the ViuSasa Video on Demand (VoD) service delivers video, music and live television channels as a subscription service (*Viusasa*, no date). It is important for local hosting providers to work with local content providers to ensure a rich selection of content is available and even governments can have a role in the promotion of local content development, particularly when it preserves national and regional heritage as well as addressing any economic/business or technical/skills issues that are identified (Kende and Rose, 2015).

2.7.10 Internet Access

The vast majority of Internet users in East Africa access services via their mobile handset, the MNOs that serve them therefore become an essential cog in the Internet ecosystem. It is also a reality that mobile Internet costs in Africa are the most expensive in the world. Within Africa there is a significant variance between countries, even between neighbouring countries, with the continents current price range per 1 GB of data downloaded between €1.80 in Mozambique and €30 in Equatorial Guinea with a median price across the continent of €6.10. The prices in Uganda is on the continental median at €6.20 with Kenyan, Rwandan and Tanzanian prices all falling well below the median at €3.35, €2 and €3.9 respectively.

It can be demonstrated that there is an inverse relationship between the number of MNO operating in any African country and the cost of data charged to the consumer. The exception to this rule is Ethiopia where there is only one state monopoly MNO and pricing is government controlled (Ecobank, 2018). Because Africa is starting from a low base, it is forecast to increase mobile penetration faster than any other region over the next half decade and it is estimated that there will be 300 million additional subscribers as well as 44 million IoT devices to come online across the region by 2025 (GSMA, 2018b).

2.7.11 The Internet and business

Globally, links have been shown between mobile penetration, Internet access and innovation. ICT provides both direct opportunities for innovation as well as a vehicle for knowledge dissemination that plays a pivotal role for innovation (OEDC, 2012).

In Uganda for example, Small/Medium Enterprises employ more than 2.5 million people, making up 90% of those employed in the private sector and contributing over 70% to total GDP (Asiimwe, 2017). Yet according the World Trade Organisation (WTO), SMEs perceive, *unreliable and/or low band Internet access and inadequate national telecommunications networks* within their top 5 constraints to entering, establishing or moving up value chains (WTO, 2016).

It is therefore essential that governments focus on innovation development and SMEs as an

engine for job creation. A major indicator of a healthy nurturing business environment is an increase in mobile penetration rates. Actions by governments that can facilitate this include reducing mobile licensing and spectrum costs which encourage new entrant MNOs into the market, reducing sectoral taxation linked to incentivised market competition as well as the development of regulations and policies that encourage the reduction of pricing by MNOs. A positive example from Kenya described in sub-section 2.7.2 demonstrates this. When the government removed VAT and other charges, handset sales doubled and penetration jumped to 70% (WEF, 2018). However, there are negative examples too such as the introduction by the government of Uganda of two new controversial taxes in July 2018 (PWC, 2018). These included an excise duty on MoMo transactions on receiving, payments and withdrawals at 1% of the transaction value and an OTT tax that levies UGx 200 (50c) per day to access social media sites such as Facebook and WhatsApp. Both of these caused major controversy and a backlash from the public. The first impacted significantly on the most vulnerable who rely on MoMo for banking and transactions declined by €160 million in the first two weeks of enforcement and by the end of August 2018, MTN Uganda, has reported a 30% decline in revenue over the previous two months which resulted in a backtrack by government who subsequently reduced the tax to 0.5% and only on withdrawals (Kafeero, 2018). The second impacted many Ugandans who rely on WhatsApp for Short Message Service (SMS) style services, it was viewed as a means of controlling communications between citizens and as an attack on free speech. The Uganda Revenue Authority (URA) targetted €6 million revenue per quarter but collected only €4.7 million in the first quarter of the tax (Kasemire, 2019). In January 2019, 6 months after the introduction of the taxes it was reported that Internet users had fallen by five million and penetration was down from 47% to 35%. The negative impact of the taxes disproportionality impacted on the most vulnerable with the cost of 1 GB of data jumping to nearly 40% of their average income (Nanfuka, 2019).

2.7.12 Internet and digital politics

Administrations across the region have a tight control of print, broadcast an social media. Even in Kenya during the political unrest of 2007 the media broadcast and printed, pacifying, pre-

approved news while the international media was providing real time reporting of what was occurring on the ground. It is understandable then that citizens, particularly the youth, turned to the Internet and social media as a means to digest and disseminate news. For example 8.6% of twitter activity in Africa contains political messages compared to 2% in the U.K. and U.S. (Nyabola, 2018).

Governments are therefore in a conundrum, on one hand the Internet and digital platforms allow them to present Government services in a streamlined way. For these systems to be effective, Governments need to address the low levels of literacy to facilitate citizens access to the digital platforms. But the opposite side of this coin is that more citizens are now capable of being better informed and educated through the Internet. As mobile Internet penetration rates rise, so too does the level of uncontrolled information that citizens can access which means that Governments no longer enjoy the levels of control they had in the past. This leads to Governments fear of the digital space and a longing for control or at least curtailing the influence of the Internet at key times.

In 2016, 11 African countries exercised Internet shutdowns, including Uganda which restricted access to social media platforms by ordering the shutdown of Facebook, Twitter, WhatsApp, and mobile money services on the eve of the 2016 elections for four days starting from the 17 February 2016. There was a second shutdown on during the hours before the inauguration ceremony of President Museveni on the 11 May 2016 (Freedom House, 2016).

The government has also cracked down on local content production by requiring all online publishers to obtain a license from UCC, ISPs were ordered to block unregistered sites. This directive strengthened the power of UCC to limit speech online. In February 2019 UCC ordered the Monitor newspaper to suspend its online newspaper (Daily Monitor, 2019).

The OTT Tax that was imposed, originally proposed by President Musevini, as a measure to curb "lugambo" (Lugwere language meaning 'gossip') online and improve the country’s tax base. This can also be viewed as a mechanism to stifle citizens’ online activities, particularly the most vulnerable who cannot afford the tax (Freedom House, no date).

Kenyan have been proud of their rapid digitisation since the landing of the submarine fibre-optic cables in 2009, rapidly progressing to the top of the mobile penetration rates on the continent and their very high Internet penetration rates. The Kriegler commission reforms after the election violence during the 2007 elections led to the development of the Kenya Integrated Election Management System (KIEMS). However, during the 2017 election KIEMS failed, in the main due to human interference. Over 25% of the KIEMS systems sent to polling stations were not used, ostensibly because of network issues, however, these reports were challenged by political and civil society organisations. The Kenyan Government considered enforcing an Internet shutdown but to their credit it never materialised (Nyabola, 2018). The Kenyan Supreme Court considered that these irregularities tainted the election result and ordering that a new vote be held within 60 days. This was also a real demonstration of the independence of the judiciary in Kenya as they resisted intense pressure from political leaders (de Freytas-Tamura, 2017). In order to reconfirm the results of the second election the back-up paper ballots were also checked to ensure the result was fair. In conclusion, while the digital voting system in the first election came in for significant criticism, it is clear that what was really at fault was the manner in which the system was manipulated.

The Internet and social media have had impact on recent elections in the region. Governments across the region have tried, to varying extents, to gain control through Internet blockages, surveillance mechanisms, use of false social media accounts to spread false information as well as to attack opposition politicians is common. There has also been some moves towards a Chinese model of Internet censorship as a way for governments to control their citizens through technology. Chinese cyber companies and officials have been active in Africa which preceded the passage of restrictive cybercrime and media laws in Uganda and Tanzania in 2018 (Grant Gross, 2018) (Shahbaz, 2018).

2.8 Research gap

As can be seen from this literature review, research in the field of IXPs is largely focused on the creation of scale and more recent research considers such questions in terms of the potential to implement SDN at IXPs thereby creating SDXs. Furthermore the SDX research is targetted at very large scale implementations, focused on European and North American IXPs to create iSDXs. The research has concentrated, in the main, on trials but also in some live deployments as demonstrated at TouSIX. Research activity surrounding SDX is directed towards the potential for more secure and flexible traffic engineering techniques to inter-domain and multi-domain routing problems, DDoS mitigation as well as mechanisms that can enable the interconnection of SDN islands, in future scenarios where SDN will replace the traditional AS.

By contrast, this research focused on enabling affordable IXPs through the simplification of design, automation of build and containerisation of functions. In addition the research aimed to develop a set of remote IXPs managed centrally through models that can be selected for and applied to sites based on location suitability, affordability and demonstrating how the models are scalable, in particular in the context of resource constrained developing countries. Considering the trust of research focus on SDN to enable large scale iSDX as well as the potential for IXP applications at SDXs to solve other problems, it also made sense to consider SDN in the context of the PoC to develop SDX for medium and small sized sites in developing countries.

2.9 Summary

Goal 9 Industry, innovation and infrastructure, of the SDGs identifies that Internet access has a direct impact on GDP. Internet can also be a tool to enable governments to support the redistribution of jobs from capital cities that in many cases cannot absorb the level of rural migration witnessed today. This is particularly pertinent in East Africa where cities such as Kampala are expanding, in a largely unplanned way, due to excessive migration. To solve this dilemma it is incumbent upon governments to develop effective spatial strategies that offer alternatives for citizens looking for opportunities. Such strategies involve the planned development of alternative,

competing cities through the building of the necessary infrastructure from roads, water and sanitation, electricity and communications. A part of this involves Internet development through connections to the submarine fibre-optic cables at the coast as well as the IDCs and IXPs to facilitate technology businesses.

ISPs connect to the Internet via purchased Internet transit and through settlement free Internet peering with other ISPs. Internet peering is very attractive because not alone does it reduce Internet transit costs, it improves customer experience which increases traffic consumption and offers a real means of traffic control for ISPs within the regional Internet peering ecosystem. Another key element of a regional Internet peering ecosystem are IDCs which offer the infrastructure to house computer servers, storage, networking and telecommunications infrastructure. ISCs often house an IXP which offers a key point in the ecosystem for both bilateral and multilateral Internet peering.

Internet quality in a regional Internet peering ecosystem is measured by the volume of traffic that remains local due to peering as well as the technologies that are used to supply services to businesses and end-users. Five maturity levels gauge the maturity of the Internet in a region and the services that can be supported. In East Africa, as fibre-optic services are restricted to small areas of major cities, the maturity level is largely indicated by the mobile service generation as well as the quality of such mobile services. As Internet traffic today has switched from static text and picture based content there is increasing demand for the ecosystem to attain higher maturity levels.

IXPs provide ISPs, MNOs and ICPs the facility to interconnect locally which serves to retain content locally and therefore provides a positive effect on traffic latency, hop count, packet loss and jitter. It has also been demonstrated in other jurisdictions, such as Brazil, that increasing the number of IXPs can further serve to increase network speed and reduce costs within the locale of each exchange.

SDN and NFV are two disruptive technologies that are changing the nature of networking in much the same way as virtualisation and containerisation did through elastic cloud and storage

almost a decade ago. It is therefore clear that these technologies can have a place in the future of IXPs. Projects such as the Google Cardigan and the TouSIX as well as academic works and prototyping have pointed the way to a future for SDX and the potential for new approaches to inter-domain routing leading the way to the potential for more secure and flexible traffic engineering techniques in the future.

East Africa is the newest region of the world to be connected to the Internet via submarine fibre-optic cables. Their landing in 2009 has triggered development of infrastructure and co-operation right across the region. It has been identified that while infrastructure is an important foundation it is not the major barrier to increased Internet penetration. To further penetration, and by linkage GDP, the barriers of device and access affordability as well as skills and awareness among citizens is key. Kenya has led the way and now its citizens now enjoy the highest rate of Internet penetration on the continent. However, content is also a major driver and East Africa has been very slow to build IDC infrastructure. Such infrastructure is the foundation required to host content both local and international. There has also been unwelcome evidence of regional governments deploying more effective mechanisms to control Internet activity, by limiting connectivity using technology and taxation tools, controlling infrastructure as well as blocking and filtering content. Governments have also imposed temporary shutdowns of Internet networks during sensitive political events as has been the case in East Africa on a number of occasions in recent times, particularly during elections.

3. Methodology

3.1 Introduction

The methodology underpinning this research can be described in two parts. The initial part consisted of a mixed methods political economy study and survey which informs and underpins the PoC development. The study commenced by exploring the recent history as well as the impact of Internet development in Africa, particularly East Africa, over the last decade, since the landing of international submarine fibre-optic cables on the coast. It then proceeded to garner the thoughts of the Subject Matter Experts (SME) engaged in the development of the Internet ecosystem to Internet eXchange Points (IXP), the impact of IXPs on the region to date as well as the potential direction they will take in the future. The study assessed the attitudes of SMEs to the disruption potential of new software-defined paradigms such as Software Defined Networking (SDN) and Network Functions Virtualisation (NFV) and concluded by eliciting their thoughts on Internet evolution and ecosystem change into the future.

The second part involved the development of a Proof of Concept (PoC) system design to investigate models for the automated build of IXPs of various roles such as a core IXP (cIXP) or local level mini IXPs (mIXP). Each IXP build is characterised by the schema of information that is supplied to it. Furthermore this component investigated the integration of an SDN framework within the PoC and the development of functionality from the framework to form a Software Defined eXchange (SDX). This SDX exposes an interface that can deliver future new functions, services and processes as the technology matures. Finally, the PoC includes an integrated mechanism of management for remote mIXPs directly from the cIXP, without compromising the independent peering nature of each IXP.

A PoC is *an experiment or a pilot project that demonstrates that a design idea or concept will work*. In information systems research Nunamaker et al (Nunamaker Jr, Chen and Purdin, 1990) first articulated this approach by positing that system development in information systems research should be the result of observation, theory building and experimentation applied to a research

domain. The design is intended to incorporate elements of social and engineering approaches. It was further developed by Burstein (Williamson, 2002) who said that a system development methodology comprises three steps, (1) Concept development, (2) System building and (3) System evaluation. The major emphasis in the system development approach is on setting out the concept that must be illustrated. These research methodologies were developed together as Action Design Research (ADR), the merger of the ideas of intervention focused, iterative and participatory Action Research (AR) that link theory with practice (Petersen *et al.*, no date) (Santos and Travassos, 2009) and the *build and evaluate* cycle of Design Research (DR) giving a new design that *reflects the premise that IT artefacts are ensembles shaped by the organisational context during development and use* further elaborates on the earlier work (Sein *et al.*, 2011). There were further refinements of these research design principles through Laboratory based ADR (LADR) which links laboratory based design where the research exerts significant control over the PoC that will then intervene in a real-world situations. LADR also promotes the combination of mixed methods research to facilitate data triangulation (Ralph, 2014). Another influence on the research design of this work is the Design Science Research Methodology (DSRM) which considers the creation and evaluation of IT artefacts intended to solve organisational problems as three activities, problem identification and motivation, define solution objectives; design and development, demonstration, evaluation and communication (Peffers *et al.*, 2007).

3.2 Research Design

Building on the methodological principles defined in ADR, LADR and DSRM this study took a multi-methodological approach to research design within the research domain of the *Internet and IXPs in East Africa* as illustrated in Figure 12.

The *project context* was extrapolated through literature review and a mixed methods political economy study. SMEs from across the East African Community (EAC) participated in qualitative interviews with a set of questions assembled in an Interview Guide (Appendix B1). The SME set consisted of business, political and technical leaders who have participated in the

development of the Internet in the region since the fibre-optic cables landed. Between the 25 participants they have a combined 550 years of experience. The qualitative interviews were triangulated by a survey (Appendix B2) undertaken of engineers and academics who are currently working in the Internet sector at the African Network Operators Group (AfNOG), the Information Society and Technology (IST) Africa (ISTA) 2018 conference and the African Peering and Internet Forum (AfPIF) 2018 conference. Taken together, the mixed method study elements informed the project context laying the foundations for the technical solution.

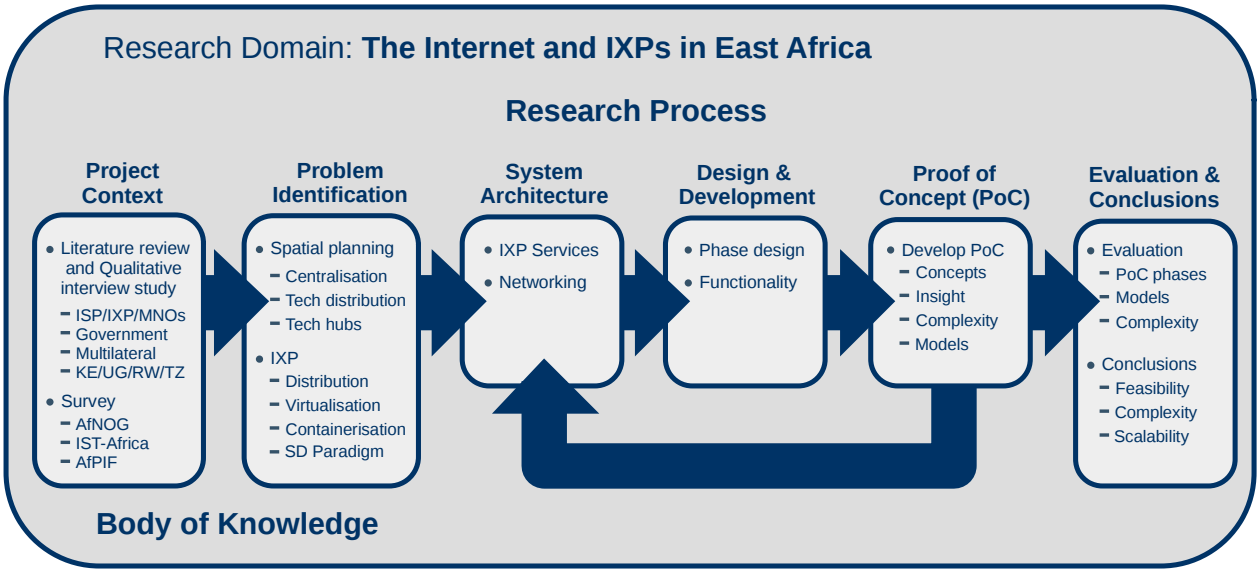


Figure 12: Research design

The *problem identification* was informed by the need for improved spatial planning in order to address the problem of migration as characterised in *Goal 9: Industry, innovation and infrastructure* of the Sustainable Development Goals (SDG).

In the context of this larger problem, this research has focused on the IXP as a key component of delivering the Internet outside of capital cities. In preparation for the design and building of PoC iterations, consideration was given to technologies that could form part of the PoC, such as the Operating System (OS), virtualisation and containerisation technologies, orchestration, potential SDN Controllers (SC) and OpenFlow (OF) switch hardware and software options.

The next three steps in the process were iterative, functionality was layered on that developed in the antecedent loop. The *system architecture* step involved the consideration of the

architecture from the previous loop as a two simple questions; (1) *were all the elements in the previous loop appropriate for the current iteration?*, (2) *how can the next set of requirements be accommodated in the architecture?*. Having answered these questions the *design and development* was considered within each iteration, the appropriate hardware was acquired and configured along with the development of the software to deliver a working PoC version.

The final step in the process, *evaluation and conclusions* considered the achievements of the PoC overall. The PoC was documented in the IXPBuilder operations manual (Appendix A) as well as further documentation of the internal workings of the IXPBuilder PoC software. Conclusions and recommendations were made from the overall research.

3.3 The PoC System Architecture

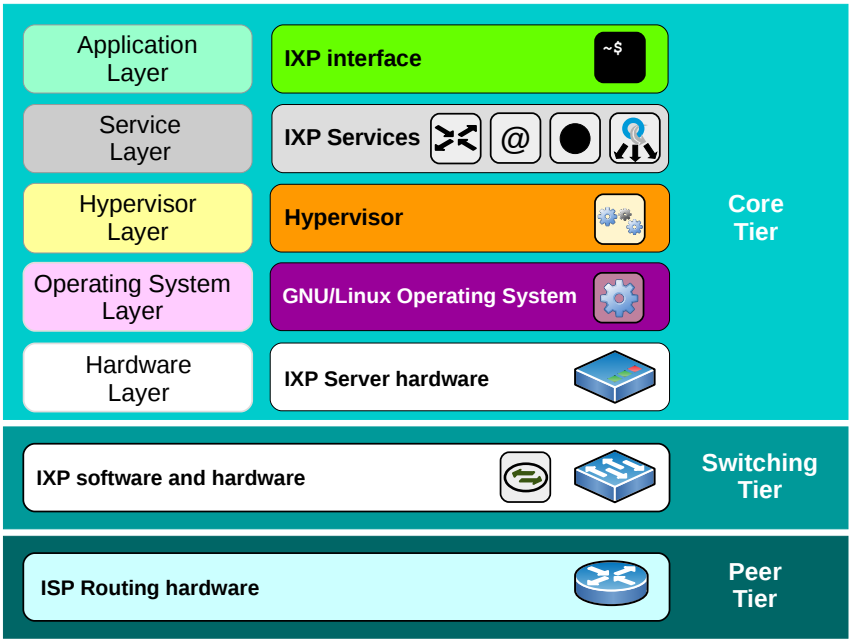


Figure 13: PoC System Architecture

The PoC system architecture as illustrated in Figure 13 has been divided into three main tiers. The lower tier, the *peer tier* which consists of IXP member equipment and has largely remained outside the scope of this work. The middle tier, the *switching tier* would traditionally have consisted of Ethernet switches connecting IXP members in order to permit them to peer. It was proposed that as part of the design of the PoC that this tier would cater for both software and

hardware based Ethernet switches. The software switching element was installed on the IXP server hardware at the *OS layer*; however, as the element responsible for peer switching it is logically considered to be part of the *switching tier*. The upper tier, the *core tier* provides the OS which supports the LinuX container hypervisor Daemon (LXD) upon which LinuX Containers (LXC) are build to house the various IXP services. This tier also provides a Command Line Interface (CLI) application which interfaces with an IXP library module to build and manage the IXP.

3.4 The PoC build phases

As has been described there are many interdependent elements to the IXP and the PoC must build IXP models in order to facilitate various IXP sizes and configurations as well as the consideration of the centralised management of each mIXP in the dIXP from the cIXP while maintaining their peering independence.

3.4.1 The baseline (v1.0, v1.1, v1.2)

The initial iterations of the development of the PoC have involved the building of a basic IXP system. The focus of this iteration has been the configuration of three IXP peering members, indicated as *ISP1*, *ISP2* and *ISP3* in Figure 14. The computer with a GNU/Linux OS hosts a Route Server (*RS*) to exchange routes between members with an open peering policy (Jasinska *et al.*, 2017). The BIRD Internet Routing Daemon (BIRD) (Filip *et al.*, no date) is a popular choice among IXPs worldwide with about two-thirds of all IXPs using it (Offerman, 2016). It also has a GNU General Public License (GPL) so it was open for inclusion in the PoC, free of cost. It therefore made sense to use BIRD as the *RS* function.

The objective of these initial design iterations was the recording of the *ISP1*, *ISP2* and *ISP3* router configurations as well as the BIRD *RS* configuration to operate as a simple IXP. Each ISP router supports hosts connected on ISP networks to validate the configuration. Configurations were developed for a number of popular routers from Cisco, Juniper, MikroTik and Netgear at the *peer tier (*Ó Briain, 2018) (Section 16, Appendix A). This established a baseline peer tier for later iterations when the focus shifted to development at the switching and core tiers.

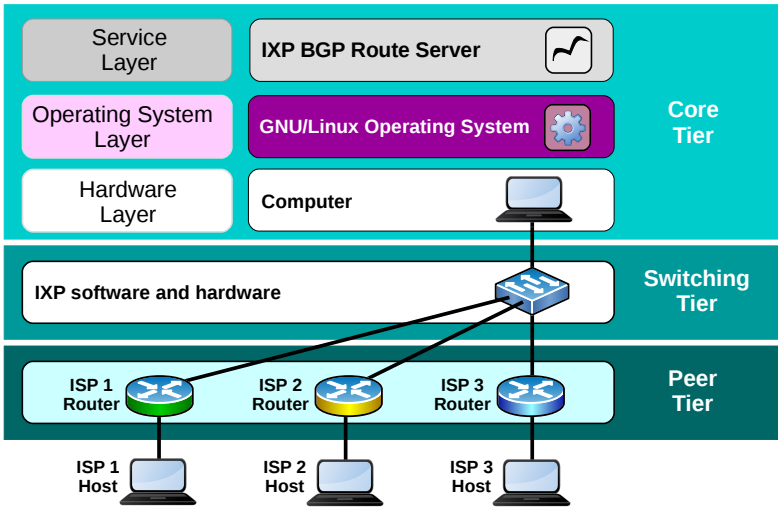


Figure 14: Initial PoC build

The BIRD software configuration was adapted to test configurations in order to provide both the *RS* function and a Route Collector Server *(CS)* function, which serves to collect routes for statistical purposes. The *CS* function operates in a similar way to that of the *RS* except that the *CS* only *imports* routes from peers, it does not *export* routes to them. This means that an IXP can require members to peer with the *CS* no matter their peering inclination and therefore peering with the *RS* maybe optional in the IXP terms, conditions and policies.

The second iteration of this phase, version 1.1 was focused on the virtualisation of the *service layer* of the *core tier* using the VirtualBox Virtual Disk Image (VDI) format. Virtualisation in the form of Virtual Machines (VM) like the Kernel VM (KVM) offer value to IXPs keen to reduce their hardware outlay. While this version was functional future requirements for more than two VMs was considered a potential risk.

In the context of the PoC these VM image sizes to host the *RS* and *CS* functions as well as the impact of VM instances drawing on host memory resulted in a cost performance on the host. Considering that future iterations would need to virtualise additional functions it made sense to consider other options like containerisation. As the host of the PoC uses Ubuntu 18.04 as its OS and the *RS* and *CS* functions run on GNU/Linux it made sense to experiment with containerisation in the testbed and thereby avoid a potential problem later in the research. LXD and LXC are shown to reduce the hardware specification of hosts as LXD can achieve 14.5 times greater density when

compared to virtualisation using KVM, instances launch up to 94% faster and there is 57% less latency experienced (Canonical, 2015). LXD/LXC are also open source technologies under the Apache 2.0 license. These tests proved successful and the final baseline iteration, version 1.2, migrated the *service layer* of the *core tier* to LXC hosts on the LXD and for version 1.2 the host *OS layer* was then virtualised using the VirtualBox VDI format for portability.

At the end of the baseline phase, peer router configurations as well as BIRD configurations for *RS* and *CS* were confirmed. The testbed was transportable and usable for training and functionality testing (Ó Briain, 2018).

3.4.2 IXP Services & Virtual Local Area Networks (v2.0, v2.1 & v2.2)

The next set of iterations, as illustrated in Figure 15, were characterised by significant experimentation, particularly at the *switching tier* and *core tier*, building on the baseline phase. In version 2.0, two Linux Bridges were added to facilitate the creation of two Virtual Local Area Networks (VLAN), one for management and the second for peering. This has facilitated the separation of the management traffic and member devices at the *peer tier*. Data from these bridges are combined on one VLAN trunk interface to the managed Ethernet switch with VLAN tags identifying the source of the different frames.

At the managed Ethernet switch, the frames are separated such that some interfaces on the switch can be established as peering access interfaces, while others can configured as interfaces for management. These pathways are illustrated as solid (blue) and dashed (red) lines between the managed traditional Ethernet switch and the two bridges in Figure 15. ISP routers from the peer tier connect to the managed traditional Ethernet switch interfaces configured as peering access interfaces. The LXC hosts have internal ports on both LANs, connection to the peering LAN so ISP routers have connectivity to their IXP functions while a connection to the management LAN is maintained for back-end operations. During this phase configurations were produced for a number of popular managed Ethernet switch OS, at the *switching tier*, from Cisco, Juniper, MikroTik and Netgear were produced (Section 15.1, Appendix A).

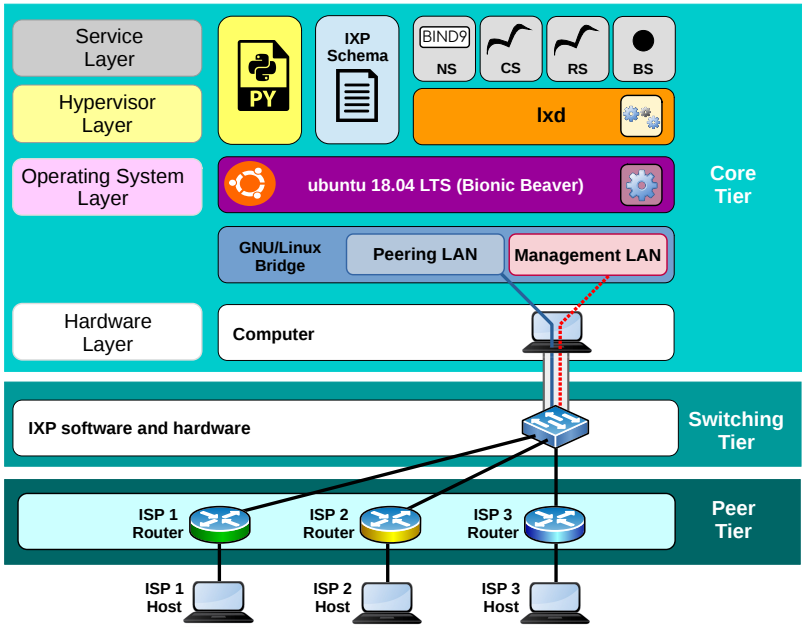


Figure 15: PoC v2.0 - IXP Services and VLANs

There were also new functions hosted in LXCs which increase the services offered by the PoC. In addition to the *RS* and *CS* functions, the IXPs are expected to have have a Domain Name System (DNS) server (*NS*) (Mockapetris, 1987) and an AS112 Blackhole Service (*BS*) (Abley and Sotomayor, 2015) to deal with DNS *reverse lookup* queries for IP addresses which are part of private address space.

Each of four LXCs host a *NS*, a *CS*, a *RS* and a *BS*. In the case of the *NS*, the Berkeley Internet Name Domain (BIND) from the Internet Systems Consortium (ISC) is the de facto standard DNS server. The current version BIND version 9 (BIND9) has an open source Mozilla Public License (MPL 2.0) which is compatible with the GNU GPL and Apache open source licenses. It has market share of between 55% and 70% of the DNS servers worldwide (Huston, 2015). BIND9 is therefore the ideal choice of DNS software for inclusion in the PoC.

The *BS* acts as a distributed sink service for *reverse-lookup* DNS queries corresponding to private local scope IP addresses in order to reduce the load on upstream DNS servers (Abley and Sotomayor, 2015). In terms of software the *BS* nameserver and routing functions are implemented using a combination of BIND9 and BIRD software as have already been described.

To develop automation within the PoC, a python3 function library module called `ixp.py`

was developed as well as a python3 based CLI in order to implement its functionality. During operation this program assists with the development of the IXP schema which is used to identify the required software packages to install. It created LXCs to host functions, installs the required BIND and BIRD software and configures the various IXP elements in the PoC with the aid of templates. Schema and other data is stored in Yet Another Markup Language (YAML) files. The IXP library module installs the elements required by the ecosystem and configures each based on the variables within the IXP schema and then presents a CLI interface that will allow peers to be added, deleted and monitored.

A limitation identified during this phase was the single interface between the computer and the managed *traditional* Ethernet switch. It was necessary to acquire a Common Of The Shelf (COTS) server hardware with multiple Ethernet interfaces before considering further development of the testbed.

An installation tool `ixp-install.sh` was developed and added to version 2.1. This tool automates the installation of the PoC on the Ubuntu 18.04 LTS OS server hardware and version 2.2 replaced the Linux Bridge functionality with Open virtual Switch (OvS). At that point it was understood that OvS would become a necessary part of the development of an SDX variant of the PoC; however, OvS could also be used in the PoC variant interfacing with managed *traditional* Ethernet switches therefore introducing it at this early stage made sense.

3.4.3 IXP models to operate with traditional switches (v3.0)

The next iteration of the research design focused on the development of new models of operation. These models allow the PoC to operate with *traditional* managed Ethernet switches. Earlier version 1 and 2 iterations had been based on single and low Ethernet interface numbers on the hardware so peers were connected to external managed Ethernet switches. These models had been labelled ‘A’ for a single interface and ‘B’ for a two interface computer. As the complexity of the IXP schema and the IXP library module `ixp.py` increased it was considered necessary to replace the YAML files with a Structured Query Language (SQL) database.

The COTS server, as illustrated in Figure 16, at the hardware layer incorporated eight Ethernet interfaces and facilitated the development of new models that could expand IXP operations at various IXP site types. It was considered necessary to reserve two interfaces for In Band and Out Of Band (OOB) management functions.

Model ‘C’ was developed with the COTS hardware and PoC software providing the complete IXP solution for a small site, an IXP in a box. This model has been designed to support sites where the number of peers are small.

Model ‘E’ provided for the connection of managed *traditional* Ethernet switches to each of its non management interfaces to cater for a larger number of peering members at large IXPs, for example at core sites. Separation of each LAN was facilitated by each of these interfaces acting as VLAN trunk interfaces.

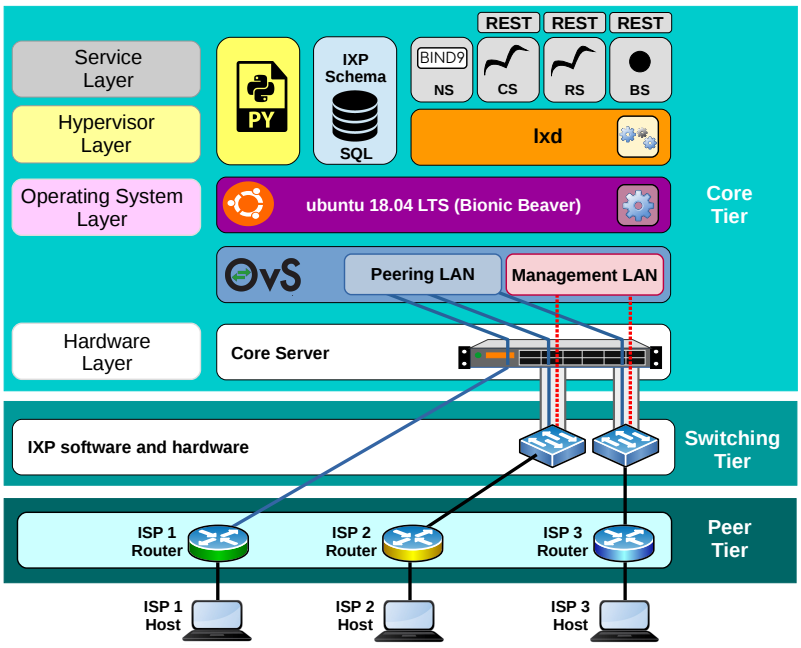


Figure 16: PoC v3.0 - Traditional IXP models

Model ‘D’ has provided for flexibility, some interfaces provide for the connection of managed *traditional* Ethernet switches in a similar way to Model ‘C’ while also facilitating the direct connection of peers as in Model ‘E’. Such a model is considered as a mechanism to future proof growing sites.

3.4.4 Object Oriented Programming (OOP) paradigm (v4.0)

After the previous iteration it became clear that the IXP library module `ixp.py` had become quite large and the number of functions within the library difficult to manage. To this end the library module was rewritten using the Object Oriented Programming (OOP) paradigm to leverage the organisation and object nature of class and method. Classes were created which included methods to replace many of the functions from the earlier versions.

3.4.5 Software-defined switching models to create an SDX (v4.1)

This iteration facilitated the incorporation of SDN functionality via three additional models to turn the PoC into an SDX. To facilitate this, an SC function was added at the *service layer*. There are a number of potential open source controller options; however, *Ryu*, a component-based SDN framework freely available under the GPL compatible Apache 2.0 license, was chosen. It is lightweight and has a python3 codebase making it ideal for integration within the PoC.

As well as class developments within the IXP library module `ixp.py` there was a need to develop an SDX sub-class of the Ryu framework `ryu.base.app_manager` class called `ixp_switch_13`. This sub-class provides the SDN functionality through communication with the IXP library module `ixp.py` as well as with both internal OvS and external OF Ethernet switches.

The *SC* was given controlling access to the OF switches in the *switching tier* as shown by the thick dashed black lines in Figure 17. These lines represent South Bound Interfaces (SBI) using the OF protocol to form a control-channel over Transmission Control Protocol (TCP) port 6633 on the management LAN. This allows the *SC* to manage both the internal peering LAN bridge in the OvS and any external OF compliant Ethernet switches. The *SC* has no port on the peering LAN, it controls OF switches via the control-channel over management LAN.

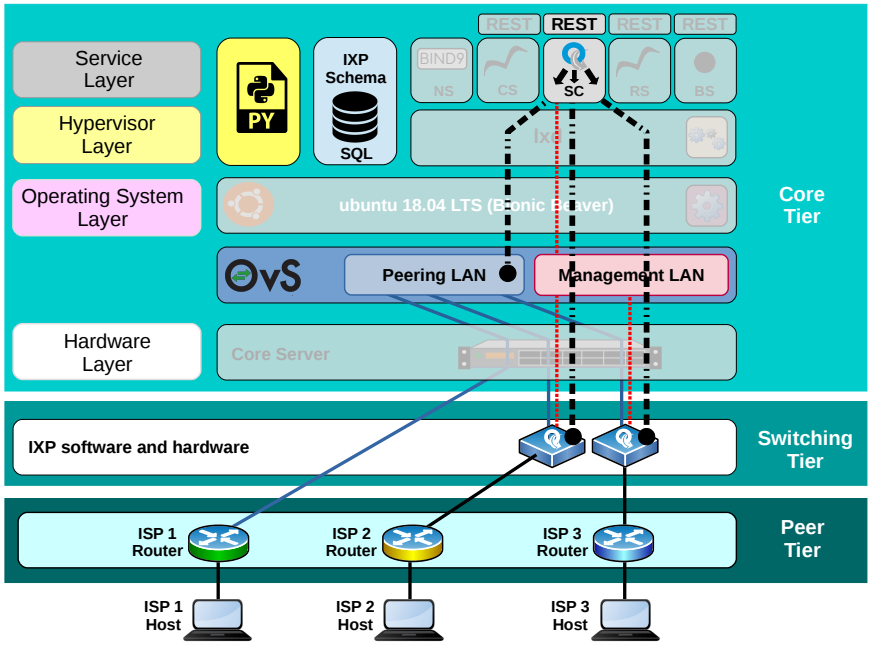


Figure 17: PoC v4.1 - SDN Controller managing the switching tier

New models S – U adopt the basic logic of models A – C . Model 'S' offers a similar level of support for smaller sites with limited numbers of peers. Model 'U' is considered similar to model 'E' as it facilitates the connection of external OF Ethernet switches and model 'T' offers a mix of options in a similar way to model 'D'. However, these SDX models offer further functionality. The *SC* hosts a Representational State Transfer (REST) Application Programming Interface (API) `ryu.app.ofctl_rest` which permits the manipulation of OF tables in both internal and external OF Ethernet switches. The sub-class `ixp_switch_13` has been developed to automatically block non configured interfaces on either the internal OvS or external OF Ethernet switches. External OF Ethernet switches require no separate configuration as they are managed directly by the SC function.

3.4.6 Remote management of mIXPs from the core (v5.0, v5.1, v5.2)

The final research design iterations were used to developed a facility to manage remote mIXPs from the cIXP within the dIXP set. As management traffic levels are small, a facility that securely traverses any network connection, such as the open Internet, was created within the the IXP library module `ixp.py`. The functionality has provided for additional commands to be made available on the cIXP PoC instance CLI that monitor and manage the day-to-day operations of local

mIXP PoC instances. The PoC leverages the existing Rivest, Shamir and Adleman (RSA) pubic/private key authentication mechanism within the GNU/Linux OS to secure management traffic. An `IxpRemote` class was added to the IXP library module `ixp.py` which allows for the generation of a key pair on the cIXP. The public key is distributed to each mIXP and this key combination protects the confidentiality and integrity of the management data passing between the sites. Commands at the cIXP, but associated with the remote mIXPs, provide the dIXP administrator with the mechanisms to add, delete and list peering members at each remote mIXP as well as review their BGP status and routes tables.

3.5 PoC testing

3.5.1 Functionality tests

End-users on any Internet Service Provider (ISP) should be able to achieve connectivity with any end-user or service of other peering members who have an open peering policy through the IXP. Connectivity with other nodes on the Internet is achieved by end-users over their ISPs transit connections via upstream providers. In the case of this PoC and any testbeds that are generated from it, the ISPs are added such that they appear to only have connections to the IXP. When there are multiple IXPs within a dIXP, connectivity between end-users on different ISP networks should be only possible if both of the ISPs are peering on the same IXP whether that be the cIXP or one of the local mIXPs.

As a result of this, end-users who demonstrate connectivity on a testbed generated by the PoC must be connected to end-users of ISPs who are also connected to the same IXP. The lack of transit connections in the testbed prevents any alternative connection paths. Verification of PoC model functionality can therefore be determined by conducting a matrix of connectivity tests between all end-users.

3.5.2 Usability tests

The PoC software, IXPBuilder, along with the IXPBuilder manual (Appendix A), was given to two separate groups of undergraduate BSc in Telecommunications Engineering students at the

College of Engineering, Design, Art and Technology, Makerere University to test the install-ability and the ease of build of the PoC.

3.6 Summary

This chapter had set out the overall design of this research as well as the methodology involved in the development of the architecture of the PoC system.

It described the initial part which consisted of a mixed methods political economy study and survey that exploring the recent history, the impact of Internet development in East Africa since 2009, specifics on the impact of IXPs to date and the potential impact the Internet, IXPs and new technologies into the future. This informs the design parameters for the PoC.

The chapter then progressed to consider a PoC, that in the context of the larger problem of spatial planning, could deliver IXPs in hub towns and cities as a key component of the Internet ecosystem. Drawing on AR, DR, ADR, LADR and DSR methodologies a multi-methodological approach to research design within the research domain of the Internet and IXPs in East Africa was presented. A description of the *build/evaluate* looped nature of the development with each iteration built upon its antecedent was described.

A PoC three tiered system architecture was described as the peer, switching and core tiers. The core tier was further described as the hardware, OS, hypervisor layer and service layer. Each set of build phases from the baseline through the development of IXP Services and VLANs, IXP models to operate with *traditional* switches, the switch to OOP paradigm, *software-defined* switching models to create an SDX and remote management of mIXPs from the core were described. The PoC section concluded with a description of the functionality and usability testing.

4. The Internet in East Africa, a mixed methods study

4.1 Introduction

East Africa is the last major area of the world to gain access to the Internet when submarine fibre-optic cables landed at Mombasa, Kenya and Dar-es-Salaam, Tanzania in 2009. The region previously relied on satellite communications to individual Internet Service Providers (ISP). This presented a unique opportunity to acquire and document the thoughts of key business, political and technical leaders who were, and continue to be, an integral part of the development of the regional Internet ecosystem from the 23 July 2009, the first day of operations of the SEACOM cable (SEACOM, 2010). This prompted a mixed methods political economy study of the Internet in East Africa to gain an understanding of why the regional Internet infrastructure developed as it did, the vision of these leaders as to the future direction of the regional Internet, their view of the potential disruption of new networking technologies such as Software Defined Networking (SDN) and Network Functions Virtualisation (NFV) as well as the growth of the Internet's multinational online infrastructure, software and service companies that have begun to dominate the Internet. This chapter presents an extract of the complete study outputs as they apply to this research.

4.2 The study area, the East African Community

The East African Community (EAC) as illustrated in Figure 18 is an inter-governmental organisation in the Great Lakes region of East Africa. The EAC partner states of Burundi, Kenya, Rwanda, South Sudan, Tanzania and Uganda co-operate in matters of politics, economy and societal development for their mutual benefit. The EAC partner states share both a customs union and a common market. The member states have a common passport and are working towards monetary union. The ultimate aim of the EAC is a political federation of East African States (EAC, no date). The capital of the EAC is located at Arusha City in Tanzania.

The EAC is the subject of this mixed methods study and there was particular focus on Kenya, Rwanda, Tanzania and Uganda. The selection criteria for the four countries was based on the fact that both South Sudan and Burundi have ongoing Irish Department of Foreign Affairs and

Trade travel advisories against all travel by Irish citizens, in the former case due to an ongoing armed conflict and in the latter due to a general deterioration of the security situation in Burundi, since an attempted coup d'état in May 2015

Figure 18: East African Community

In this study Uganda is considered the inland country which has a dependency on access to the submarine fibre-optic cables at the coast. For the research there was a choice to be made between Kenya and Tanzania as the countries with submarine fibre-optic cable landing stations and interview participants from both countries were included. However, as the Ugandan Internet traffic uses the Kenyan route and Rwandan ISPs generally prefer the Uganda/Kenya route over Tanzania, priority was given to Kenya as the main coastal country for the project. This is because Kenya has more submarine fibre-optic cable options, as well as a very liberalised telecommunications market. The final country of the four is Rwanda, which was selected due to the unique dependence it has on multiple countries to reach the submarine fibre-optic cables at the coast. While it borders Tanzania and has links to the submarine fibre-optic cables landed there, the Rwandan providers, in general, choose the longer route through Uganda and Kenya to access submarine fibre-optic cables in Mombasa. This is due to the fact that the Tanzanian link is owned and managed by a single

incumbent Tanzania Telecommunications Company Limited (TTCL) and is far more expensive than the links through the liberalised markets in Uganda and Kenya. Added to this is the greater choice of submarine fibre-optic routes in Kenya. For these reasons the three countries Kenya, Rwanda and Uganda are termed the sub-region.

4.3 Study Interviews and study

Qualitative interviews were carried out with 25 business, political and technical leaders who are a Subject Matter Expert (SME) group in the telecommunication sector in East Africa (‘*the participants*’). These interviews were semi-structured in which participants were given open-ended focused questions from an interview guide (Appendix B1) on a series of topics relating to core Internet provision using well defined interview techniques (Edwards and Holland, 2013) on the:

- state of the Internet today,
- changing nature of Internet traffic,
- function and benefits of Internet eXchange Points (IXP),
- future *software-defined* networking paradigms,
- expected evolution of future networks,
- Internet ecosystem changes that are necessary in the future.

The study employed a purposive sampling technique to select the participants. Purposive sampling is a technique where the participants were selected based on their usefulness to collect focused information. The selected participants were those with many years experience in the sector (Flick, 2014). 27 participants were invited to participate and 25 agreed giving a response rate of 93%.

The study was conducted with the aim that the outputs meet the credible, dependable, confirmable and transferable criteria expected to ensure rigour (Houghton *et al.*, 2013). To obtain credibility, the participant list was given a good geographical spread across the sub-region with 24%

of participants from Kenya, 24% from Rwanda, 44% from Uganda and 8% from other parts of the region. This was achieved by the researcher visiting each participant at their offices in Dar-es-Salaam, Kampala, Kigali and Nairobi. Consideration was also given to ensure that there was adequate representation from those involved in policy, in this case the multi-lateral and government ministries; those involved in regulation, in this case telecommunications regulators, and finally those involved in implementation, namely the ISP, Internet Data Centres (IDC) and IXPs.

Dependability and confirmability criteria of the data from the interviews was established by setting a baseline for each theme and category. To warrant inclusion in the analysis it need to be agreed by more than one participant and the more participants that made similar points the more dependable the data was considered. However, considering the expert nature of the participants, individual comments were, in some cases, included where they offered rich information and it was possible to corroborate the information. Transferability determines if the findings from the interviews can be transferred to other similar contexts or situations, while still preserving the meanings and inferences (Leininger, 1994). The information was categorised, ordered with detailed descriptions provided for each theme and category such that a reader considering another context or situation can determine the transferability of each point.

The outputs were also corroborated using across-method methodological triangulation via an Internet Survey of engineers and academics who also work in the sector delivering the Internet across Africa (‘*the respondents*’). The survey form (Appendix B2) was distributed electronically to the African Network Operators Group (AfNOG) membership, made available in hardcopy at the Information Society and Technology (IST) Africa (ISTA) 2018 conference in Gaborone, Botswana from the 09 - 11 May 2018 and at the African Peering and Internet Forum (AfPIF) 2018 conference in Cape Town, South Africa from the 21 - 23 August 2018 as summarised in Table 6. Respondent numbers in each group are within statistical norms for a small survey when Hoyle’s Finite Population Correction Factor (FPCF) (Hoyle, 2018) is applied to the ideal sample size from the Krejcie and Morgan research tables (Krejcie and Morgan, 1970).

Table 6: Internet survey participant statistics

Group	N (pop)	n_{km} (K&M)	n_h (FPCF)	n (Actual)
AfNOG	49	43.56	23.31	34
ISTA	41	37.13	19.74	21
AfPIF	54	47.45	25.51	28
Total	**144**	**104.93**	**60.95**	**83**

An anonymised version of the dataset was used for analysis using a custom program written in the R statistical programming language (R Project, no date). Each question was assessed for statistical significance between the three groups using Pearson's χ^2 test of homogeneity with positive evidence of homogeneity indicated by a p-value greater than 0.05 and weak or no evidence of homogeneity if the p-value returned is less than or equal to 0.05.

4.4 Regional Internet, improvements over the last decade

All participants were clear that the Internet and Information and Communications Technology (ICT) in general has been transformed in East Africa over the last decade. They also pointed out that the landing of the submarine fibre-optic cables connected to Europe, the Middle East and India at Mombasa and Dar-es-Salaam has triggered this transformation moving the region from a satellite based, low bandwidth, high latency Internet to a rapidly developing terrestrial fibre-optic network interconnecting the towns and cities of the region and connecting them to the submarine fibre-optic landing points at the coast. As a result the volume of data that can be processed from the region without adding more fibre-optic cables exceeds 20 Tb/s.

> To me the changes are unbelievable not just for Internet but for ICT technology as a *whole, but still I wish it would be more reliable because I guess I am bit of a perfectionist even when things are working I wish they would work even better. ("Paul", Chief Executive Officer (CEO) at an ISP)*

> *The advent of the fibre-optic cables also coming into through Mombasa through to Uganda as well and coming inland also transformed the Internet space. ("George", Director at an Internet related organisation)*

It has become like electricity, it's a medium that's happening in the background without having you pay attention to the fact that yes, that's happening because of that connectivity or that linked up because of the Internet. ("Ambrose", Senior government official)

The survey respondents were asked two related questions to accurately elicit their thoughts on how the Internet is delivered across the region today.

The state of the Internet in the region over the last 10 years is:

Statistically there was strong evidence of homogeneity between these groups. Considering the raw data analysed directly and illustrated in Figure 19, a relationship can be seen with between 52% - 68% of all three groups selecting *very improved* and between 32% and 43% of the three groups selecting *slightly improved* indicating that all respondents considered the Internet to have improved in the region over the past decade.

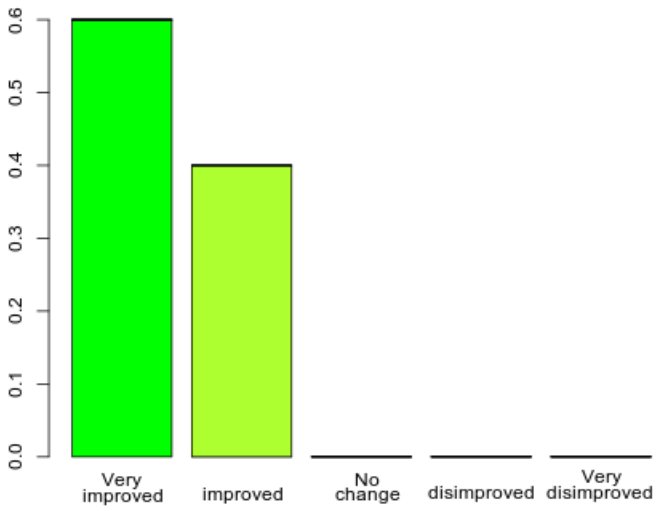


Figure 19: The state of the Internet in the region over the last 10 years is

The second question explored the respondents understanding of the reason for that change.

The key catalysts for change on the Internet has been:

There was no statistical evidence of homogeneity between the groups on this question, particularly in the first and second preference selections. This is evidenced graphically in Figure 20 for first preference choices. It can be seen, however, that respondents from each group has a strong

preference for *Submarine cables landing in Africa* and *Mobile Network Operators (MNO)* at the top of their choices.

48% of the ISTA group chose *MNOs* as the key catalyst whereas for this group *Submarine cables landing in Africa* was the second choice at 43%. 46% of the AfPIF group also selected *Submarine cables landing in Africa* as the key catalyst and 14% selected *MNOs;* however, the second choice at 18% from this group was actually *National terrestrial fibre networks*.

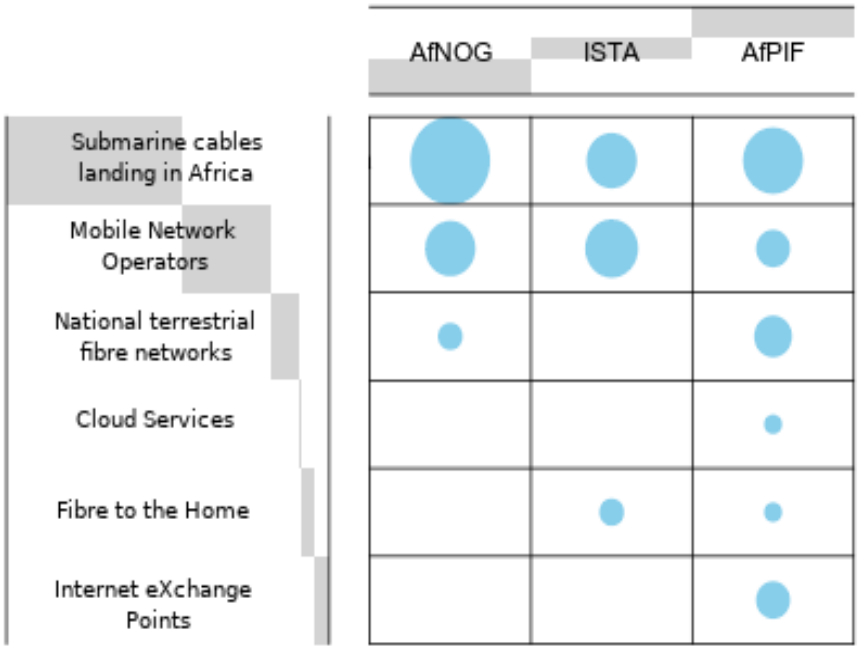


Figure 20: χ^2 test; the key catalysts for change on the Internet has been

While the null hypothesis of homogeneity was rejected statistically, the data demonstrates that the top two reasons for all three groups was the same. It could be surmised that the reason for this small discrepancy between the groups relates to the fact that 88% of the AfNOG group are from East Africa and as this region was only connected to the submarine fibre within the last decade it is not surprising that the effect of *Submarine cables landing in Africa* rated so highly.

Interestingly, the third choice votes in this ranked question showed moderate evidence of homogeneity between the groups.

4.5 Internet eXchange Points

> *The role of the IXP is to make the end-user experience a better experience, because if they are able to access content faster then they consume more, this means more revenue for the ISPs, because the ISPs will have to sell bigger pipes. ("Robert", Principal Engineer at an ISP)*

There was general consensus from participants that IXPs have been a key part of the Internet ecosystem since the mid 1990s. IXPs were originally formed to allow ISPs in a country or region to pass traffic between them for mutual benefit instead of passing the traffic over Internet transit adding unnecessary cost. The IXP has also become a point of data exchange between ISPs, MNOs and Internet Content Providers (ICP) typically at IDCs. It has, and continues to provide an important tool in the ecosystem to keep local data, local and prevent hairpinning of data over transit.

> *Before we had IXP connections I saw huge amounts of traffic going from one local network to another local network, going all the way out to the Internet and all the way back in again. Ridiculous situations. ("Doreen", Chief Technical Officer (CTO) at an ISP)*

> *Without IXPs the growth of the Internet in the country cannot be achieved to the fullest potential, the reason for that as we know the Internet eXchanges are developed or created for the basic concept of keeping the local traffic local. ("Joel", CEO at an ISP)*

Active participation and strategic routing policies allow members to avoid significant Internet transit, as well as the problems associated with traffic hairpinning via the Middle East, Europe and even the U.S. IXPs in East Africa have developed significantly from very low traffic volumes 2001. Traffic across the switching fabrics at individual IXPs today is measured in Gb/s, for example Uganda IXP (UIXP) regularly handles in excess of 6 Gb/s.

> *There were only 160Mb/s in traffic going through eXchange Points across all of Africa. ("Assumpta", CEO at an ISP)*

Some participants outlined that in Kenya the IXP initially came online at the end of 2001 but after two weeks the Communications Authority of Kenya powered it down and confiscated the equipment after a complaint received from the then incumbent operator Telecom Kenya. They

claimed it was illegal and they, as the national incumbent, were the only body allowed to provide an IXP under Kenyan law.

> *In February 2002 there was an agreement and a license was issued to the IXP and we went in and switched it on. It was actually Valentine's day of 2002. ("Anne", Director at an Internet related organisation).*

The Kenyan exchange grew slowly at first but in 2010 traffic, having started at 400 Mb/s, passed the 1 Gb/s mark for the first time. Today Technology Service Providers of Kenya (TESPOK) maintains two IXPs, Kenya IXP (KIXP) in Nairobi at the East African Data Centre (EADC) and the Mombasa IXP (MSIXP) at the Telephone House Data Centre (THDC). KIXP regularly exchanges over 6 Gb/s of traffic today.

> *In 2010 within a period of 7 months KIXP grew traffic exchanged from 400 Mb/s to 1 Gb/s of traffic. ("Robert", Principal Engineer at an ISP)*

Both KIXP and MSIXP are managed by the TESPOK as a neutral, non-profit IXP for its members. KIXP does not impose restrictions upon the types of organisation or individual who may become members and currently supports peering between ISPs, content providers, government institutions, banks and other companies that wish to peer with each other.

> *The eXchange is owned by TESPOK but the mandate is really open to everyone, anybody can be a TESPOK member. What is needed [to join] is an Autonomous System (AS) number (ASN), this makes you a carrier or an ISP in that sense. ("Robert", Principal Engineer at an ISP)*

UIXP also started in 2001 and it is the third oldest IXP in Africa. It started with 20 Mb/s share of the African traffic, whereas, today it regularly peaks over 7 Gb/s. UIXP has 27 networks connected and it includes ICPs such as Google and Akamai, as well as, eGovernment services to each of the ISPs and MNOs in the country.

> *At UIXP, today we have a lot of diversity, we have 27 networks connected and each one of them, generally speaking, brings something to the table, and some provide more or less value to some of the other networks, there is a little interesting market going on at the eXchange Point. ("Assumpta", CEO at an ISP)*

The addition of Content Delivery Networks (CDN) by ICPs have been instrumental to the development of the IXP. With each addition there was a noticeable step in traffic throughput.

> *In Uganda right now having a cache at the eXchange Point, the networks have a really strong desire to come because the only other way to get that Google content is through the expensive transit upstream. ("Assumpta", CEO at an ISP)*

The Rwanda INternet EXchange (RINEX) was founded in 2004 and was initially located in the IDC of the incumbent Rwandatel. Initially it was ran by volunteers from MTN, Rwandatel and from industry but in 2014 RINEX was formalised under the Rwanda Internet Community and Technology Alliance (RICTA). RICTA had already taken responsibility for the country code Top Level Domain (ccTLD) .RW in 2012. Since 2015 RICTA is funded from revenue received from both the .RW domain and RINEX. Typical traffic volumes between the RINEX 15 peering members is 2 Gb/s.

> *RICTA took responsibility for the IXP [RINEX] it was 2014 but the IXP itself had been operational since 2004. It had been people from mainly MTN, Rwandatel, as well as from industry who were managing it on a day to day basis and it was located at the incumbent [Rwandatel]. Today it is ran by RICTA employees. RICTA is a private organisation limited by guarantee paying taxes, staff, etc... Revenue traditionally comes from the domains and since 2014/2015 both the domains and the IXP are generating revenues. ("Ismail", CEO at an ISP)*

> *Once we connected to the eXchange [RINEX] within four months we offloaded 100 Mb/s of traffic, we offloaded roughly about 15 to 20% of the traffic and now it is moving up. We were able to cut our upstream providers from 1 Gb/s, over six Synchronous Transport Module 1 (STM-1) to just four STM-1s. ("Robert", Principal Engineer at an ISP)*

> *IXPs are very important, very vital in the development of the future Internet in Africa. Without the IXP the whole digital economy is chopped. There is no innovation, there is no financial services, no government services that can happen. ("Ismail", CTO at an ISP)*

The Tanzania Internet Service Provider Association (TISPA) formed the Tanzania Internet eXchange (TIX) at Postal House in Dar-es-Salaam in 2003. The government through the Tanzania Communication Regulatory Authority (TCRA) built a fibre-optic ring called the National ICT

Broadband Backbone (NICTBB) and convinced TISPA to develop smaller IXPs at sites in other cities along the path of the ring. They also directed ISPs to connect to these other sites. The long-term goal of TCRA is to create a national distributed IXP through the NICTBB which the ISPs and TISPA are not comfortable with; however, TCRA have issued decrees which are forcing their compliance. TCRA are also requiring TISPA to migrate the original TIX in Dar-es-Salaam to the TTCL site.

> *TIX was established in 2003 by the ISPs themselves through the association TISPA then later on the government through TCRA donated equipment and convinced TISPA to establish regional [local] IXPs, the goal was to, at a later stage to have a ring to connect all these IXPs so that we have a sort of a national Internet eXchange Point connected through the national ICT broadband. ("Gabriel", CTO at an ISP)*

While an overall goal of this project is the consideration of distributed IXPs (dIXP) in the context of developing countries, there is little point in establishing them without a business case or a clear plan that has the support of stakeholders such as ISPs. In Tanzania for example there are five IXPs but only TIX, the IXP not currently under government control, is a viable IXP with over 4 Gb/s of traffic on average on its switching fabric. The other IXPs on the NICTBB at Zanzibar, Arusha, Dodoma and Mwanza were established by government directive and while ISPs have connected in order to be in compliance, the level of traffic is measured in kb/s. At Arusha, the capital city of the EAC, both Google and Akamai initially installed caches; however, since then Akamai have withdrawn their cache due to lack of traffic (in 2018 it carried less than 250 kb/s on average). The remaining IXPs at Dodoma, Manwazi and Zanzibar have average traffic levels in the region of 140 kb/s, 500 kb/s and 325 kb/s respectively in 2018.

A similar situation has existed in Mombasa where a joint project partnership between TESPOK and the Amsterdam IXP (AMS-IX) to develop and run MSIXP at the SEACOM PoP in 2010. AMS-IX withdrew from the partnership citing lack of traffic as the reason. TESPOK relaunched MSIXP in THDC 2016.

The first point of learning from this is that before considering the development of additional

IXPs there needs to be a business case.

The second point to consider is the interconnection of the dIXPs via the NICTBB. This has been resisted by the Tanzanian ISPs as it directly impinges on their ability to deliver transit services themselves. It also counteracts any initiatives that reduce costs in the transit network. ISPs are forced to participate in the interconnected national dIXP and therefore have no reason to develop their own competing infrastructure. This is clearly demonstrated by the choices made by Rwandan ISPs to choose a route to the submarine fibre-optic cable landing stations in Mombasa via a two country route, through Uganda and Kenya, rather that the more direct route to Dar-es-Salaam through Tanzania.

> *Considering the different IXPs there currently is not a link between them, some are quite small, for example in Zanzibar IXP (ZIXP) there are three peers, Arusha City IXP (AIXP) they have 12. Take Dodoma IXP (DIXP) which currently has three members, as government headquarters has moved to Dodoma, the government is pushing us to improve it so that more could be connect there but currently there are only three peers. Dar-es-Salaam by contrast has 33 peers, Mwanza IXP (MIXP) is the final one and they also have only three peers. Most of the peers in the smaller IXPs are MNOs. They are really not getting any benefit from peering at these sites, they are doing it just to comply with the government because the numbers of peers there are very small so they are doing it just to comply. In terms of benefit it does not make sense really. ("Gabriel", CTO at an ISP)*

Survey respondents were asked a number of questions to understand how they see the IXPs within the Internet ecosystem today and if they see a place in the future for them.

To the question:

IXP have had a positive effect on the Internet in the region over the last 10 years*:*

The responses as illustrated in Figure 21 indicate strong statistical evidence of homogeneity between the three groups; however, judging by the level of *undecided* votes it appears that the role of the IXP is not as widely understood within the community as it should or could be.

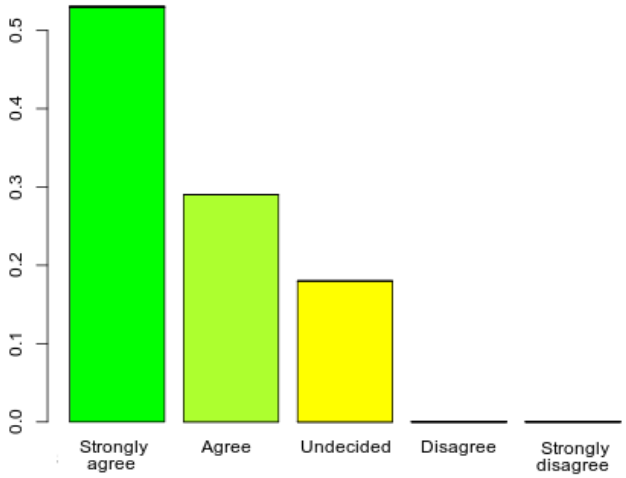


Figure 21: IXPs effect on the Internet over the last 10 years

Having said that, there is a clear majority opinion that the IXP has a positive effect across all regions and there is no opinion disagreeing with this statement.

When the survey respondents were asked for their opinion on IXPs place in the future there was no evidence of homogeneity between the groups as illustrated to the right of Figure 22. It can be seen from the horizontal bar graph to the left of the same figure that both *a future continuing to act as a point where ISPs* and *Application Service Providers (ASP) peer* and *a future to share local content* feature high for all three groups and it is possible that the respondents were confused as to the subtle difference between *a future to share local content* and *a future as a location for CDNs* as the latter is rated highly with the AfNOG group. In fact, it is surprising that *a future as a location for CDNs* did not rate more favourably among the respondents from the ISTA and AfPIF groups considering the current trend for CDNs to locate at IXPs.

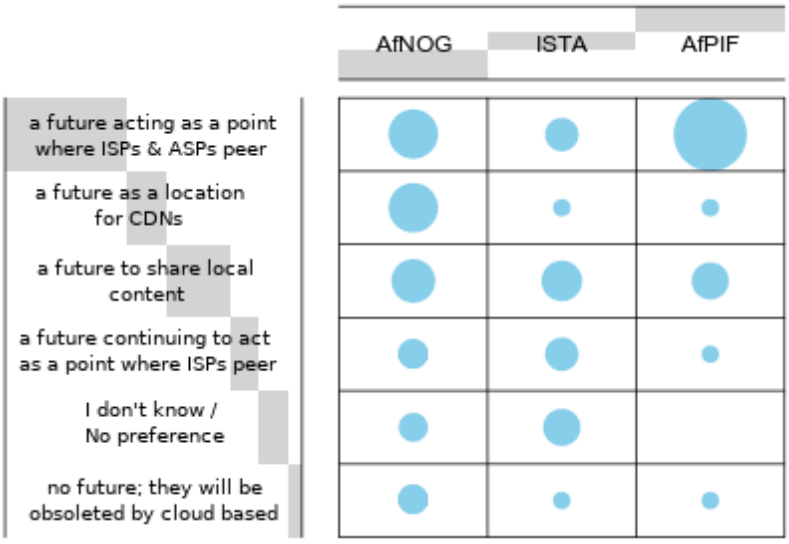


Figure 22: χ^2 test; Internet eXchange Points have:

As there is currently no universal model for IXP management and ownership it was interesting to note the responses from the survey respondents on who should run IXPs. When analysed using Pearson's χ^2 test, a p-value of 0.23 demonstrates strong evidence of homogeneity between groups. While there are differences between the choices in all three cases, AfNOG at 74%, AfPIF at 71% and IST-Africa 48%, the vast majority consider "*an organisation/entity of peering members at the IXP*" the preferred option and "*by a local ISP association*" in second place as is illustrated in Figure 23. This verifies that the European open, co-operative, cost sharing model of IXP is preferred among the East African technical community. This is not surprising given that the participants have a industry bias.

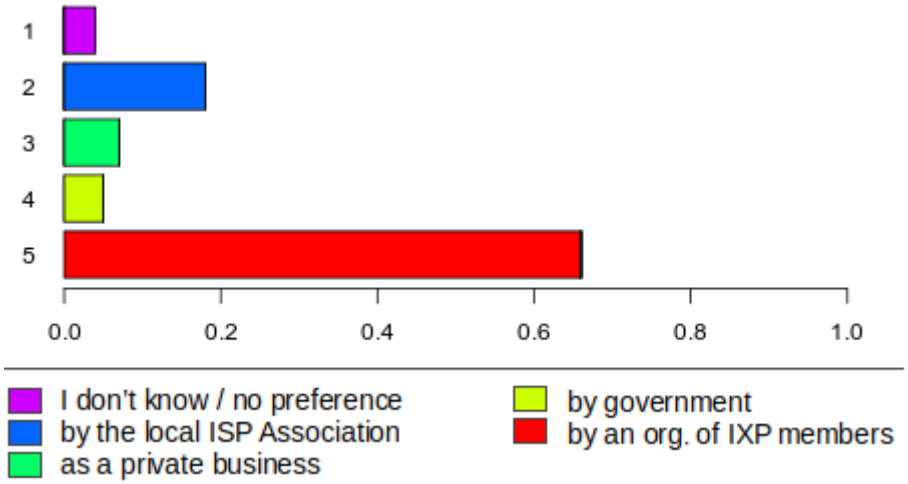


Figure 23: IXPs should be ran by

4.5.1 Content delivery networks

An IXP staff member characterised the most significant reason for the relatively recent transformation of IXPs as the incorporation of CDNs for ICPs at what is a very convenient and neutral point in the regional Internet peering ecosystem.

> Content trying to come close to the users that makes exchange points basically distribution points for content rather than a more diverse healthy robust ecosystem of varying size networks with different things to offer. So you get Google plugging into an exchange point because having central distribution points may be the most efficient way to distribute all that stuff instead of a mesh of inter co*nnected links. ("Assumpta", CEO at an ISP)*

> *What has also helped the IXP to grow is the CDN, because we are getting quite a number of CDNs posted at the exchange now, so all of us want access to those CDNs. But also the fact that we can basically keep our traffic local and cut down on all our upstream costs is a good thing for us. So yes I totally see IXPs being a very important focal point in the Internet ecosystem. When it comes to East Africa I am not of the view that we should build an East Africa IXP but I am of the view that the different IXPs need to peer each other. That makes sense. ("Denis", Senior government official)*

However, others when asked if this is healthy for the local Internet ecosystem, were non committal suggesting that *this has yet to be seen*. While it currently suits the ICPs to locate at IXPs, they could also set this up themselves and force the ISPs to come to them for connections. There is a real possibility of a monopolist position by these companies as content becomes the driver for exchange and not a healthy ecosystem of peering ISPs as well as, international and local ICPs.

> *In terms of the potential for monopolies on content, that is certainly going to be a problem, it is already a problem, any time you have a monopoly. It is a fact that is here and looks like it is here to stay because it is Facebook that is doing acquisitions at alarming rates, it is the same Google that is doing acquisitions at alarming rates. ("Lena", Director at an Internet related organisation)*

When the survey respondents were asked the question:

> ***In your opinion what has had the most significant impact on video services in the region over the last 10 years:***

There was strong statistical evidence of homogeneity between the groups for first and second preference selections. Caching and CDNs received just under half of the first preference selections and had a strong showing among the 2nd and 3rd options of the other voters. However, the very strong 2nd preference result for ISPs meant it was given almost equal significance with MNOs and IXPs also showing strong support overall. While credit must be given to ISPs and MNOs for making improvements on their networks, the fact that ICP have located CDNs on their networks and at IXPs to bring their content closer to their customers is seen as the major contributing impact to video services in the region.

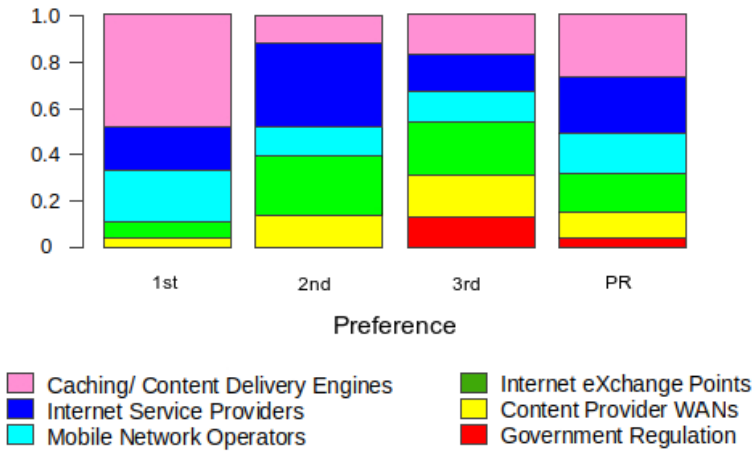


Figure 24: Most significant impact on video services over the last 10 years

4.5.2 Pan Regional eXchange Point

The EAC members came together as early as 2007 to implement the East Africa IXP (EAIXP), an initiative of the East African Communications Organisation (EACO) Working Group 5 for Broadcasting Development, Spectrum Management and Media Services Regulations on IP networks. The idea was to connect the IXPs of the EAC states into a single regional IXP such that a member in Kampala could peer with an ICP cache in Nairobi or an ISP in Juba could access services hosted in Kigali.

> The idea behind this project [EAIXP] was to have the IXPs in the region interconnect either logically, physically whichever way to trunk and exchange traffic within the region without traversing outside of the East African region. It was to improve the Internet experience of users and to reduce on the costs and latency. The EAIXP project

> changed to the East Africa Peering and Interconnection project because the experts who were involved in that project thought that conne*cting IXPs was not viable but we need to talk of peering and Interconnection in the region. ("Justine", Senior Manager at a Multi-lateral organisation)*

This raises the question as to which ISPs would get to carry the traffic between the elements of this regional IXP. Would ISPs play along and pass routing information to make it work? The EAIXP would replace some of the function that transit carrying ISPs do today. It is not surprising therefore that Simbanet who were initially awarded a contract to make the inter-connections between the IXPs were unable to implement the project due to several challenges including the unwillingness of ISPs to participate. There were many questions as to the feasibility of the project, despite the intentions behind it and this resulted in the concept being transformed into a peering forum to discuss best practice.

> It is not a case of let's wake up one day and interconnect the IXPs, we need to think about who does *it affect and who gains the most from it and what is the current structure, as we have it based on who connects to which IXP and how?. ("Denis", Senior government official)*

While the idea of the pan-EAC EAIXP did not take off, the idea that a country or region will require more than one IXP in the future was raised by a number of the participants. To develop industry and technology hubs away from the main cities will eventually require the development of multiple IXPs and/or a dIXP. Considering the EAIXP experiment, the reasons for not directly interconnecting the national IXPs apply at least equally, if not more so.

4.5.3 distributed Internet eXchange Points

> If you grow the local infrastructure, creating distributed IXPs around the country. Create a village network which can actually then be rated differently so if I am just sending mail using the village mail server and that's all I am connecting to, then I can participate in the Internet and I am not drawing down *on my data rates if I am just communicating with the local servers sitting at the local village post office ("Dickson", Director at an Internet related organisation)*

> *The IXPs have an ecosystem into the Internet infrastructure that is not going anywhere. I think what we have to do is enhance it and build more of it. A country should have at least two or three IXPs to be specific, one IXP is not enough. ("Robert", Principal Engineer at an ISP)*

Some participants visualised a future where national IXPs need to consider developing more local IXPs in other cities, where industrial hubs can form and thrive. It makes sense for government to promote such a de-centralised policy to develop other parts of the country and reduce urban sprawl around the capital. For this to be successful it is important to understand how such local IXPs can achieve transit cost reductions and improved latency times to help the local Small Medium Enterprises (SME) and industry without negatively impacting the ISPs transit business. For example questions such as *how can the local universities use the local IXP to help or to revamp their Research and Education Network (REN) to interact with the SMEs and industry locally?* or *what help can national and local IXPs get that will help them grow as an ecosystem while partnering with other entities that can be beneficial?*

> *In East Africa we are growing so give it another five years you will actually have enough technical capacity for people to actually manage an IXP in Gulu. ("Denis", Senior government official)*

> *I think the national IXP needs to put together a system that is going to help them to build better local ones because I think we need to build the IXPs locally, we need to understand how the cost reduction in terms of international capacity is going to help in the sense that how can we help local businesses, local SMEs to reach other local SMEs, we need to build a certain infrastructure that is going to help maybe a better education system for our universities for example. How can we help IXPs grow as an ecosystem while partnering with other entities that can be beneficial, like the REN or even the private sector. ("Bruno", Network Operations Manager at an ISP)*

A small number of participants consider that local IXPs or dIXPs are unlikely to happen. The reason given was the lack of trained personnel. In some ways; however, it has already happened in both Kenya and Tanzania. In Kenya KIXP has PoPs in Mombasa and Nairobi, while in Tanzania there are IXPs in Dodoma, Dar-es-Salaam, Arusha, Mwanza and on the semi-autonomous island of Zanzibar off the Tanzanian coast. It is also fair to say that specific circumstances apply in both of these cases. Mombasa is a submarine landing site and therefore an obvious place where

ISPs converge while Nairobi is the capital city and centre of industry in Kenya. While there are similar reasons in Tanzania, where the submarine fibre-optic cable lands in Dar-es-Salaam, Dodoma is the capital city, Zanzibar is semi-autonomous, and Arusha is home of the EAC, the current traffic patterns show the Dar-es-Salaam IXP is currently the only viable one.

> I also don't believe that you will have a situation here where distributed IXPs can b*e run in remote areas of certain countries any time soon. That has a lot to do with lack of qualified personnel to be able to run these IXPs and that is why I think the IXPs will stay in the main cities. ("Kenneth", Senior Manager at an ISP)*

"Lena", an executive board member of a multi-lateral organisation, cited the case of Brazil which has implemented a model of distributed IXPs around the country. In fact it is the largest distributed IXP network in the world. The goal behind the large number of IXPs in Brazil is to attract tier two and three ISPs who provide local Internet access in the isolated municipalities to expand to areas currently lacking connectivity by offering to locate an IXP in the municipalities with the associated benefits of a full Network Information Centre (NIC) service. These benefits include free co-location, a local peering ecosystem and other Internet related benefits not normally associated with IXPs in other counties.

> *Of course IXPs in places like Brazil have taken on a whole complete model of a Network Information Centre (NIC) covering everything Internet related, the IXP, the national Internet Registry, etc... ("Lena", Director at an Internet related organisation)*

When the survey respondents were asked:

> **In the future there will be the need for the establishment of IXPs in regional [*local*] towns apart from the national capital*:***

While the question used the word *regional* the meaning was regional within each country and not the EAC, within this thesis the meaning is substituted with the word *local*. The responses demonstrated strong evidence of homogeneity between groups. The results as outlined in the pie chart in in Figure 25 demonstrate a strong *yes*. Overall only 16% of respondents disagree with the statement.

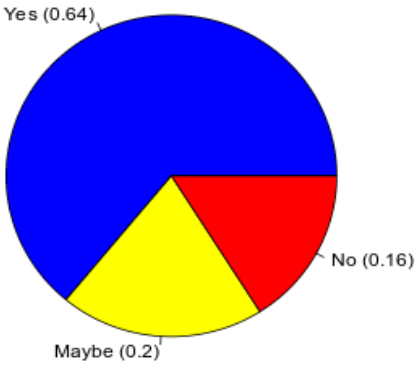


Figure 25: Will there be a need for the establishment of IXPs in regional [local] towns

4.6 Software-defined disruptive technologies

The software-defined paradigm that started with servers, then storage and now networks through SDN and with functions through NFV was recognised by many participants as having the potential to really disrupt how networks evolve and how functions are delivered over the next few years. SDN and NFV are considered technologies that will transform management and move network configuration away from manual configuration to automation.

> [Software-defined is] making it easier to integrate intelligence in centralised manag*ement and all kinds of other sophisticated stuff and you can automate it but does that actually create new ways for the networks to interact? ("Assumpta", CEO at an ISP)*

Adding intelligence to networks, making them smart, will also make them more efficient. This will reduce costs while increasing capabilities, perhaps generating or acquiring content that will change how service providers view their place in the market of the future.

> The move to software defined means there is a lot more automation, there is a lot less understanding of what goes under the hood because there is a lot more intelligence built into the system and the system is able to automatically make decisions as to how to route traffic. I fear that there are certain skill-sets which we are likely to loose and fewer engineers that actually understand. *("Anne", Director at an Internet related organisation)*

> *Today a lot of cost of the Internet is the infrastructure, expensive hardware, expensive backend infrastructure. NFV, SDN is giving us the promise of delivering all that with software. We as the service providers must think more about the applications needed to deliver a type of service wherever we want it, so they [SDN, NFV] have a huge impact*

on the Internet that way. ("Janet", CTO at an ISP)

Traffic on the Internet has been defined by the users on it. Automation through SDN and NFV will make the back-end more efficient but it will not have an impact on user behaviour.

SDN is a fancy way to say having automated Local Area Networks (LAN) and Wide Area Networks (WAN). Traffic has always been pushed by the users and I don't think that will change much. ("Sydney", CEO at an ISP)

What SDN and NFV are offering operators is the ability and flexibility to reduce CAPital EXpenditure (CAPEX) through the employment of Common Off The Shelf (COTS) instead of expensive function specific hardware. It also allows them leverage open source software projects such as OpenStack to focus their resources to tailor services for their customers needs. They can also reduce OPerational EXpenditure (OPEX) through automation and network intelligence.

SDN for operators gives flexibility and bring down the cost of OPEX, it reduces OPEX. The CAPEX may not have that huge impact now because every technology provider is taking huge advantage of the high costs right now. ("Joel", CEO at an ISP)

As the industry and the economic outlook for the region grows, there is increased demand for digital services. ISPs must scale up to meet this demand and software-defined technologies such as SDN and NFV offer an architecture that has the agility to scale with demand.

To scale up to the demands and requirements of the future generations, there is no other way besides SDN solutions because of the architecture flexibility and the scalability that it brings to us as an operator. The agility that it will bring in terms of how we respond to c*hange in capacity requirement, change in future requirements from the consumers. ("Calvin", CTO at an ISP)*

The survey respondents were asked to consider the trend towards Software-defined technologies and how this will shape the Internet of the future. To garner there view to the potential of disruptive technologies in the Internet ecosystem the survey asked:

SDN is pretty much hype and nothing will change:

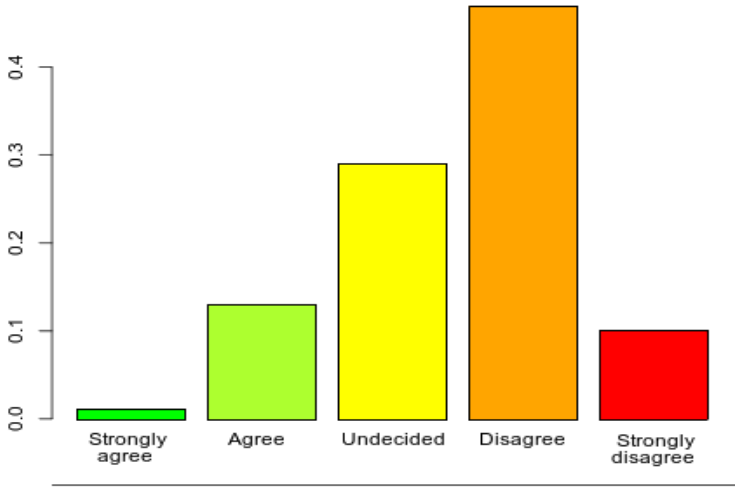


Figure 26: SDN is pretty much hype and nothing will change

The results are illustrated in Figure 26. With a large body disagreeing with the statement while many are undecided or agreeing with the statement. There was strong evidence of statistical homogeneity and for the AfNOG group 56% selected *strongly disagree* or *disagree* with a significant 29% *undecided*. A slightly larger 67% of the ISTA group *strongly disagree* or *disagree* and again 29% were *undecided*. Half of the AfPIF group either *strongly disagree* or *disagree* with the statement and again 29% were *undecided*. It is clear that the groups were in broad agreement, in fact the AfNOG and AfPIF groups had much the same response while the ISTA group were a little over 10% more in disagreement with the statement that the other two groups. This may be reflected in the fact that this group included more academics who may be part of research groups studying SDN and NFV.

The second survey question asked:

NFV is the last throw of the dice for operators, they have lost the application layer battle:

This question, as is evidenced from the results displayed in Figure 27, demonstrates some confusion among the respondents with over half undecided and practically no respondent having a strong opinion either way. When the three groups are compared there is strong evidence of homogeneity between them with a strong undecided vote in each.

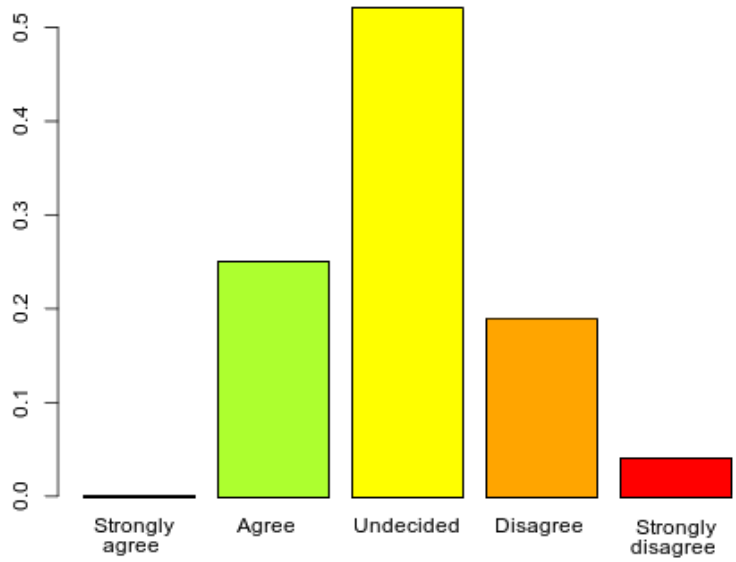


Figure 27: NFV is the last throw of the dice for operators, they have lost the battle

Considering that respondents are the employees of ISPs (31%) in the main, it is difficult to see NFV assume the future role for which it was originally envisioned. NFV may find a stronger use case as a virtual Customer Edge (vCE) providing a platform for Global Service Providers (GSP) to offer their services via partner Local Service Provider (LSP) networks (Weldon, 2015) and within the core network of MNO Fifth Generation (5G) New Radio (NR) networks.

4.7 Internet Regulation

The rapid change in the nature of the Internet is and will continue to present significant challenges to the national regulator system. This is particularly marked by user behaviour which demonstrates a move towards large global multi-national providers like Google and Facebook. Such migration is attributed to the consolidation of traffic which has driven ICPs to locate CDNs at local IXPs and/or ISPs which in turn has led to changed traffic patterns across the global Internet as more traffic is retained within the region.

> Unfortunately we seem to be going back to the same thing, we are having now dominant players taking over again, becoming almost de-facto telecoms without much hindrance and that unfortunately that means that the version of the Internet is what they are selling in their flavour because this thing makes some money. *("Janet", CTO at an ISP)*

Almost 70% of the survey respondents did not see a potential problem with this, however, there was a more marked difference of opinion among the interview participants, some who considered this convergence as a potential risk of the emergence of a monopolist or oligopolistic situation to develop in the future. While some said that the regulatory mechanisms can deal with this, many were of the considered opinion "*that regulators today seem ill equipped to deal with multinational monopolies as they gain dominance over user behaviour and control of content*".

4.8 The future of the Internet in East Africa

The evolution of the Internet can be considered in terms of connectivity and infrastructure on one hand, and services on the other. Wired connectivity will see fibre-optic cables getting closer and closer to the customer, first to towns with Metro Ethernet style products, then to street cabinets as Fibre To The Kurb (FTTK) and finally into homes via Fibre To The Home (FTTH). Wireless extensions to the fibre-optic tails in the home via WiFi to give customers the flexibility and convenience of access will become common as is the case in developed countries. The main wireless technology of the new decade will be 5G NR. 5G NR is being designed to accommodate the non human, Machine to Machine (M2M) communication to Internet of Things (IoT) devices and it is expected that 5G will in fact have far more M2M devices connected than humans over time. Handsets and other access devices will have increased intelligence for making the switch between available networks based on quality and costs. For example the handset may use 5G while in the car on the way home but upon arrival offload to the WiFi connected to the FTTH connection, without dropping the data stream or ongoing call.

> The connectivity part [of Internet evolution], you find that there is the 5G coming, but obviously the role of fibre in terms of connectivity is going to increase because when you talk 5G its becoming wire more than wireless because you have to do fibre to each and every site to be able to cater for things like IoT. So I think the import*ance of fibre in all East African countries is increasing and we see more and more players probably rolling out FTTH and fibre to everywhere. ("Calvin", CTO at an ISP)*

The other dimension of the evolution is services. With functions virtualisation it is expected that the number and breath of services available to consumers over the Internet will increase and

become far more intelligent. These services are expected to bring people, functions, processes, data, and IoT machines together as the Internet of Everything (IoE).

> I think Internet is enabling lot of services that are coming up and we are going to see a lot of IoT and IoE basically, which will enable services that we have not imagined but will basically become part of our lifestyle. Things like WhatsApp they have changed the way we communicate so I am sure there will be a lot more applications th*at are going to come that will just be a part of our lives. ("Calvin", CTO at an ISP)*

The survey respondents were asked how they see the evolution of the Internet in the future in East Africa via two questions.

> ***The top three technology drivers of the Internet evolution over the next 5 to 10 years in the region will be:***

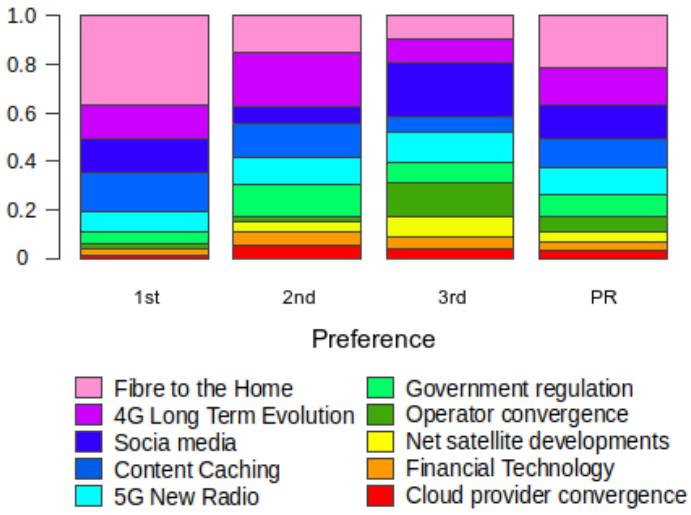


Figure 28: What will the top drivers of the Internet evolution be over the next decade

Considering the future of the Internet in Africa and the expected drivers that will take it forward the respondents gave a mixed response.

This question shows strong evidence of statistical homogeneity among first and second preference choices. The third choice option demonstrates weak evidence of homogeneity which is not surprising considering the number of options presented. There is clear agreement among all three groups that *FTTH* is the top technology driver, as each give this option between 32-39% for this selection. While AfNOG and AfPIF groups give *content caching* as their second choice, the

ISTA group consider *social media* and *Fourth Generation (4G) Long Term Evolution (LTE)* as as their equal second choice.

The second question posed to the survey respondents:

What are the top three changes you think are necessary in the ecosystem to facilitate Internet evolution in the region in the next 5 to 10 years:

The current dependence on off continent content is raised as a concern in conversations at many industry conferences in Africa currently and this is further highlighted by the responses to the question posed as visualised in Figure 29. The responses from the three groups to the question demonstrates strong evidence of homogeneity across all three preferences.

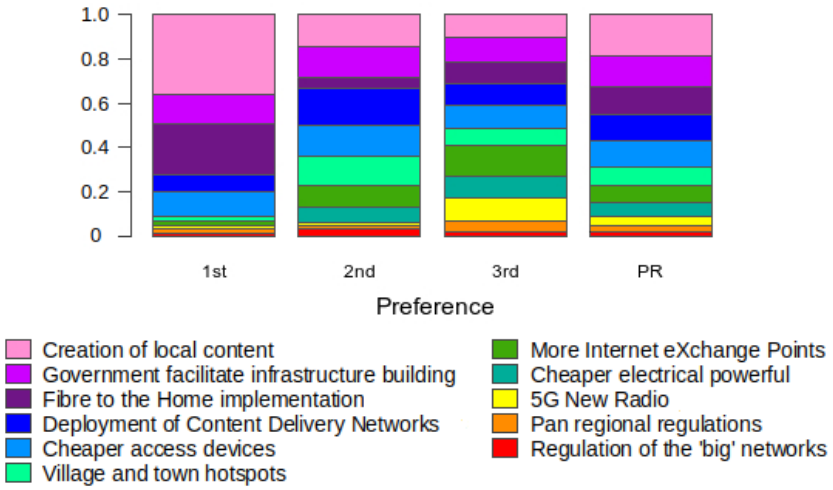


Figure 29: Changes necessary to facilitate Internet evolution over the next decade are

Taking a deeper look into the results there is agreement that *the creation of local content* is the number one ecosystem change necessary with selections from 32 – 39%, *FTTH* is given strong prominence too; however, in the case of the ISTA group it is given an equal score with the need for *cheaper access devices*.

4.9 Summary

The two mixed method components of the study, the qualitative part of open-ended focused questions in interviews and the semi-quantitative part of the Internet study in the form of a questionnaire, proved to be complimentary, in the main. The purpose of carrying out both was to

apply across-method triangulation (Boyd, 1993) so as to ensure completeness of the dataset and reach data saturation (Fusch and Ness, 2015). Similarities, as well as, variances in thinking were identified between the main expert group from the interviews and the engineer/academic group from the Internet survey.

The results of the across-method triangulation of these two elements, the qualitative interviews and the Internet study with two different sets of people, the interviewing of leaders in their respective organisations and the survey of engineers and academics were interesting in that there was broad agreement on most topics within the study. The interviews produced a much richer source of data; however, it is a far more difficult format to analyse. While, the survey is much easier to setup and analyse, the received data is not as rich in detail. The rich data received from the qualitative interview process has been verified by the Internet survey.

4.9.1 Infrastructure

It was clear that both the participants of the qualitative interview process and the survey respondents agreed that the Internet experience and infrastructure was *much improved* in the past decade. *The landing of submarine fibre-optic cables* in Mombasa and Dar-es-Salaam was seen as the catalyst for this improvement. Both groups are also in full agreement that streaming video services have improved over the decade and attributed this to the addition of CDNs by ICPs, as well as, improvements in the Internet infrastructure in terms of terrestrial fibre and peering.

4.9.2 Internet eXchange Points

When IXPs are considered, the survey respondents agree that IXPs have had a positive impact on the Internet in the region. They was some uncertainty as to the next step for IXPs with *a future continuing as a point where IXPs and ASPs peer* being slightly ahead of the other choices selected. The qualitative interview participants were quite clear that IXPs had a positive impact, and the numbers given for traffic traversing the exchange switching fabric these days compared to their origin in 2001 for Kenya, Uganda and 2004 in the case of Rwanda confirm this. They also stated that the reason for such growth in IXP traffic levels was the addition of CDNs in the ecosystem,

whether they are located in the IXP directly or at a local ISP where other ISPs were either permitted access across the IXP switching fabric or via direct peering.

While most participants in the qualitative interviews and almost 70% of survey respondents could see that in the future there would be a need for IXPs to develop in cities and towns outside national capitals. However, there was a small number of interview participants who agreed it would be a good thing but were doubtful whether it would actually happen due to skill-set miss matches outside capitals as telecommunications and ICT professionals are attracted to roles in the capital.

4.9.3 Software-defined technologies

On the issue of software-defined technologies such as SDN and NFV many of the qualitative interview participants were sceptical that they would change the nature of the Internet in terms of users and their interaction with it; however, there was broad agreement that they have the potential to increase automation and deliver flexibility to ISPs to deliver services. There was a note of caution among some that network engineers could loose their understanding of what was happening behind the network and as a result reskilling would be necessary. It was raised by some of the qualitative interview participants close to MNOs, that both SDN and NFV concepts are being integrated into the 5G NR design philosophy and a core network with a Centralised - Radio Access Network (C-RAN) based on flexible functions with hardware confined to the Remote Radio Head (RRH) and COTS servers can be expected.

4.9.4 The future of the Internet in East Africa

In terms of what will drive the future of the Internet in the region, the survey respondents were of the opinion that *FTTH*, *LTE* and *content cashing* were the top three drivers with *social media* receiving worthy of mention. The interview participants more or less agreed that an increase in fibre-optic density in both linking rural areas, as well as FTTH deployments in urban centres were key drivers too. There was also an emphasis from the interviews that wireless remains the main access method to the Internet and will continue to do so into the future in East Africa.

When it comes to actions to support ecosystem transformations the survey respondents

consider the *creation of local content* and the need for government *to facilitate infrastructure build*, as well as *FTTH implementation* as priorities. The qualitative interview participants broadly agree and also consider that for rural citizens the development of hotspots in villages, as well as cheaper smartphone devices are also critical issues.

5. A Proof of Concept for cost effective models for IXPs

5.1 Introduction

The literature review in Chapter 2 highlights the importance placed on the impact to Gross Domestic Product (GDP) of increased levels of Internet penetration. Internet penetration can also act as a tool to enable governments to encourage the redistribution of jobs from capital cities. This can be achieved by spatial strategies which involve building hub towns and cities incorporating the necessary infrastructure such as roads, water and sanitation, electricity and communications. There are two key elements to such development. Firstly, Internet Data Centres (IDC) which offer a platform for software companies to host content and services and secondly, Internet eXchange Points (IXP) which can be conveniently located at the IDC. IXPs facilitate Internet Service Provider (ISP), Mobile Network Operators (MNO) and Internet Content Providers (ICP) to interconnect locally which serves to retain content locally and therefore provides a positive effect on traffic latency, hop count, packet loss and jitter. It has also been demonstrated that increasing the number of IXPs can further serve to increase network speed and reduce costs within the locale of each exchange.

It has also been illustrated in the literature review that there has been a reluctance among some of the most successful IXPs, particularly in Europe, to link distributed IXP (dIXP) nodes for the purpose of peering. This reluctance stems from the potential hazard it would create if the very ISPs that made these IXPs successful would now consider the IXP as competition in the transit and back-haul space, namely *the IXP interconnection hazard*. By remaining neutral, with a focus on the provision of mutual value through the sharing of costs, IXPs can grow while staying focused on the provision of core IXP services.

The literature has also demonstrated that Software Defined Networking (SDN) and Network Functions Virtualisation (NFV) are changing the nature of networking and that Software Defined eXchanges (SDX) have the significant potential to change the nature of IXPs into the future.

Drawing on the mixed methods study in Chapter 4, it can be concluded that IXPs have had a

positive impact on the Internet in the region. While most participants and survey respondents suggested that there is a need for IXPs to develop in hub towns and cities, some were doubtful that this would happen due to limited skills availability outside capital cities.

When new software-defined technologies were discussed there was some scepticism expressed by participants that they would change the nature of the Internet in terms of users. However, there was agreement that such technologies have the potential to increase automation and deliver flexibility to ISPs to deliver services. This agreement is exploited in the development of the PoC giving definition to the scope and specification.

5.2 High level functional specification

When considering the design of a Proof of Concept (PoC), drawing on the literature review as well as the mixed methods study of the Internet in East Africa the following points are extracted to from part of the functional specification. The PoC must:

- *be simple to build and maintain*. Each IXP should be simple to install and given basic schema data, automate the build process of the required exchange.
- *be capable of supporting different site scenarios*. IXP builds should support models that are appropriate to difference scenarios. For example, a minimal IXP for small hub towns with a small number of peers, while larger towns and cities with a larger number of peers must be supported by the addition of extra switching hardware as required.
- *support a distributed model for management while maintaining independence for peering*. In this way the remote IXPs can be managed from the core site and offset any skills deficiencies that may exist.
- *incorporate SDN such that the IXP can operate as a SDX*. Explore mechanisms to support future applications that can increase automation and deliver flexibility into the future.

Taking these high level function specifications into consideration, this research has

developed a PoC testbed consisting of a containerised IXP in order to explore models suitable for different site sizes, technologies and configurations. The IXP build is largely automated through the development of a Python3 IXP module called `ixp.py` as well as an SDX sub-class of the Ryu `ryu.base.app_manager` class called `ixp_switch_13`. This module and SDX subclass, developed through this research, form the core of the functionality that builds and operates the PoC.

This PoC demonstrates five new models for operation that can operate with *traditional* managed Ethernet switches and three further new models that leverage *software-defined* switching paradigms interfacing with Ethernet switches that support the OpenFlow (OF) protocol at the control plane. Also considered is a method of day-to-day management of remote mini IXPs (mIXP) from a centralised core IXP (cIXP). This chapter details how the PoC software demonstrates these.

5.3 IXPBuilder, the PoC testbed

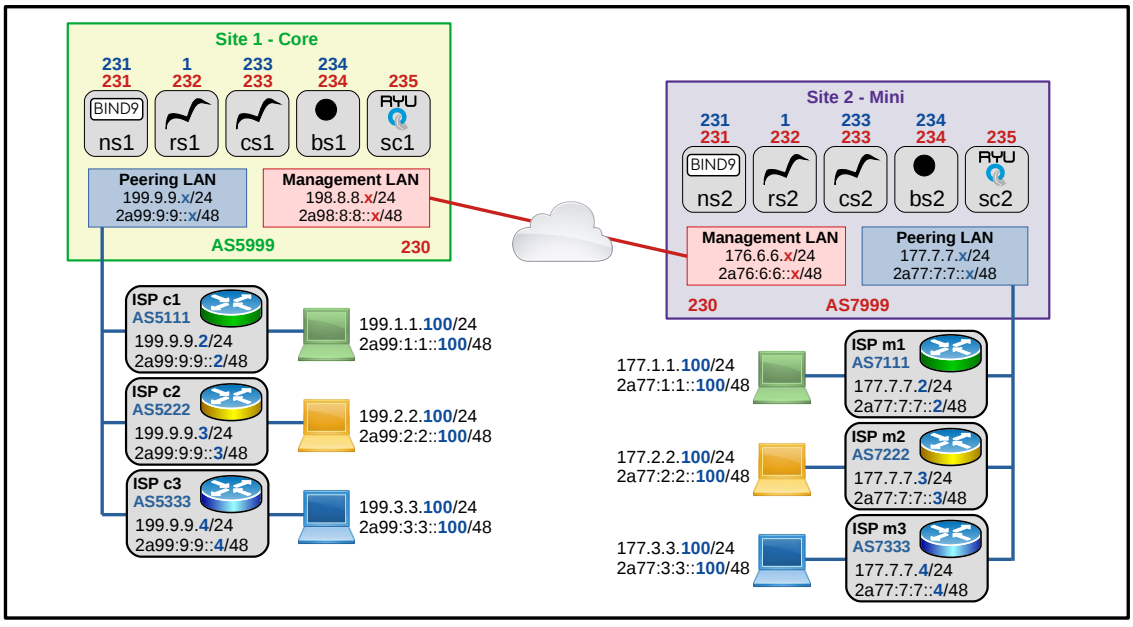


Figure 30: PoC testbed

Figure 30 is a reference tool to aid description. It outlines two sites, a cIXP and a remote mIXP with the various Internet Protocol (IP) prefixes and addresses used at each site. In the case of models employing *traditional* Ethernet switches there is no SDN Controller (SC), the schema has addresses assigned but no *sc1* or *sc2* containers are built except in *software-defined* models.

5.4 Models employing traditional Ethernet Switching

Traditional Ethernet switches are those whose hardware, Operating System (OS) and applications are closed, except for configuration, and under the control of the vendor. The focus of this research is at the *core tier* and while the models define ecosystem configurations they must be compatible with *traditional* Ethernet switches located at the *switching tier*. It is for this reason that the first set of models developed are referred to as *traditional*.

For IXP models employing *traditional* Ethernet switching the PoC software builds an embedded stand-alone Open virtual Switch (OvS) and reserves trunk interfaces to connect to external *traditional* Ethernet switches where necessary. The control and configuration of any external *traditional* Ethernet switches at the *switching tier* rests with the IXP engineer and is not a function of the PoC. During the build the PoC software detects the number of available Ethernet interfaces (ε) on the server hardware and the IXP build engineer specifies the number of *traditional* Ethernet switches (τ) required at the *switching tier* for the IXP ecosystem. The model to be applied is then selected as follows;

- IF $\varepsilon = 1$, model A is automatically selected.
- ELIF $\varepsilon = 2$, model B is automatically selected.
- ELIF $\varepsilon = 3$, model E is automatically selected.
- ELIF $(\varepsilon - 2) \geq \tau$, there is an option to select between models C, D and E.
- ELSE too many *traditional* Ethernet switches have been selected, τ must be reduced.

In model A the server hardware has a single Ethernet interface. Such a hardware configuration is not ideal for an IXP system as it cannot support Out of Band Management (OOB). To separate peering and management Local Area Networks (LAN) the single interface must be established as a Virtual LAN (VLAN) trunk in order to connect to an external *traditional* Ethernet switch. The external switch must be configured to have the first interface as a trunk interface passing the VLAN tags 100 and 900, some access interfaces with VLAN tags of 900 to the

management LAN and the other interfaces are employed for access with VLAN tags of 100 to the peering LAN.

In model B the server hardware has two Ethernet interfaces. The backup OOB management interface is crucial in an operational environment and therefore the first Ethernet interface is reserved for this function. This interface should be connected to a network with a Dynamic Host Configuration Protocol (DHCP) Server to assign an IPv4 address, a stateful DHCPv6 server to assign IPv6 addresses or a network device that will supply IPv6 prefix for configuration with IPv6 Stateless Address Autoconfiguration (SLAAC). Alternatively manually configured addresses using the Linux shell *iproute2* commands or persistent addresses via the '`/etc/netplan/01-netcfg.yaml`' file can be used. The second Ethernet interface is used in an identical fashion to that described for model A. Like model A, this is considered a suboptimal model for an IXP server to support the PoC software.

To support model C the server must have more than three Ethernet interfaces. The first interface is used for OOB, the second as a direct connection to the management LAN and subsequent interfaces are used to connect peers to the peering LAN. To do justice to this model there should be six or more interfaces. The ideal scenario is a deployment as a small remote mIXP where the number of peering members are few. For example a town with up to six peers could be served using a server with eight Ethernet interfaces and no external *traditional* Ethernet switch is required.

Model D is employed when there are more peering members than there are Ethernet interfaces, less the two interfaces for OOB and management. Like model C, the first interface is used for OOB, the second for the management LAN. There are two sub-models of this model, in the sub-model D_1 scenario the third interface is configured as a VLAN trunk interface to be connected to an external *traditional* Ethernet switch and the remaining interfaces are configured for peers. So in an eight interface scenario there are five remaining peer interfaces. The second sub-model D_2 allows for a mix of trunk and peer interfaces. For example interfaces three and four could be

configured as trunk interfaces to connect two external *traditional* Ethernet switches leaving four interfaces remaining for peers.

Model E has the first two Ethernet interfaces for OOB and management functions. Sub-model, E_1 is the only scenario suitable for a three interface server. In this case the third interface is configured as a trunk interface. For sub-model E_2, all interfaces except the first and second are established as trunk interfaces to connect *traditional* Ethernet switches. This sub-model is ideal for larger sites where a number of external *traditional* Ethernet switches are required.

Table 7: Traditional model summary

Model	Interfaces	Interfaces			
		1	2	3	4+
A	1	Trunk interface			
B	2	OOB	Trunk interface		
C	> 3	OOB	Management interface	Peer interfaces	
D_1	> 3	OOB	Management interface	Trunk interface	Peer interfaces
D_2	> 3	OOB	Management interface	Mix of Peer and Trunk interfaces	
E_1	3	OOB	Management interface	Trunk interface	
E_2	> 3	OOB	Management interface	Trunk interfaces	

5.4.1 Structure of the PoC for traditional models

Many IXPs, particularly those that are price conscious within developing countries either leverage the second hand market in high quality core switches or simply purchase off the shelf *traditional* Ethernet switches. This makes sense as IXP organisations tend to be members of larger IXP associations and with traffic growth rates in Europe there are often opportunities to access very high specification core switches cheaply or even free of charge when swap-outs are being carried out.

Figure 31 is a functional diagram of the optimum system for switching models employing *traditional* Ethernet switches. It consists of three tiers, a *peer tier* that belongs to the ISPs themselves and includes routing devices that act as Border Gateway Protocol (BGP) peers for the ISPs Autonomous System Number (ASN) within the IXP ecosystem. Suggested configuration notes for *traditional* Ethernet switches and ISP peering routers for various switching and routing devices

is given the IXPBuilder manual (Sections 15.1 and 16, Appendix A).

The *switching tier* consists of IXP owned switching devices. These include a combination of internal OvS switching functions within the PoC as well as external *traditional* Ethernet switches which operate together to provide the switching service.

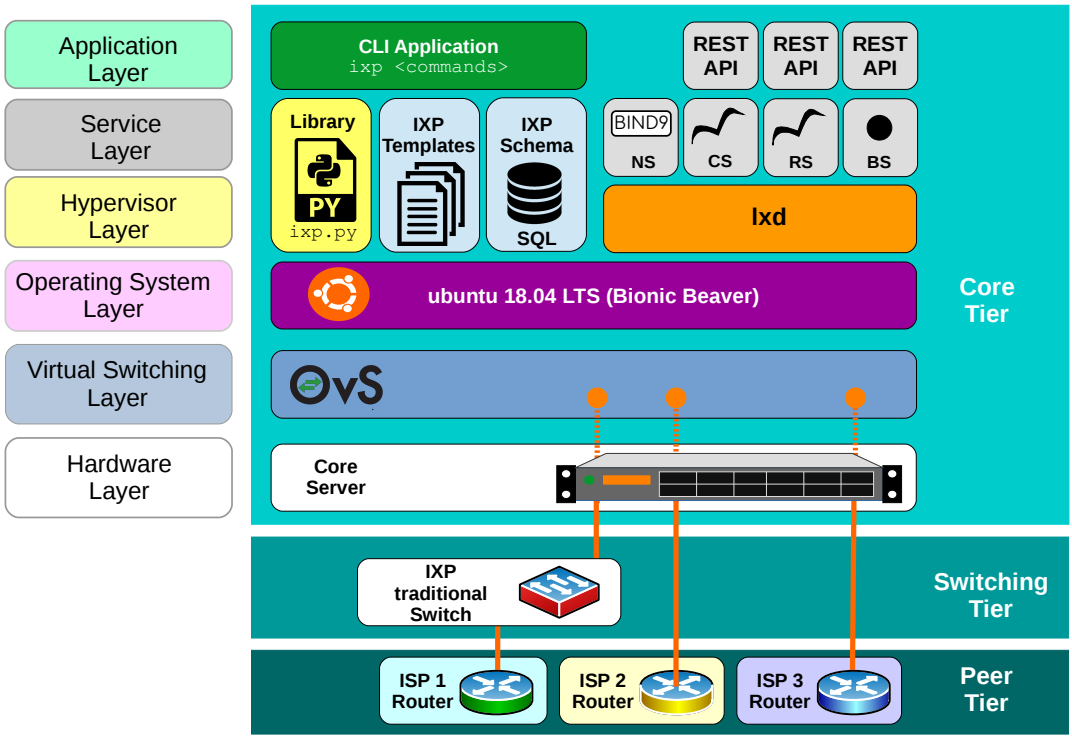


Figure 31: IXPBuilder functional diagram – traditional

The *core tier* provides most of the IXP functionality and apart from the physical Common off the Shelf (COTS) server providing a *hardware layer* within the tier, the remaining layers are software based whose function is to provide;

- an Internal Ethernet switching service,
- a LinuX container hypervisor Daemon (LXD) platform,
- LinuX containers (LXC) for services,
- a database to store operational data,
- a library of python functions, classes and methods, the PoC software,
- a Command Line Interface (CLI) application to allow access to the service layer functions,
- IXP services.

5.4.2 IXP Schema for traditional models

The design of the PoC introduces the concept of an IXP schema and it is the foundation upon which the IXP is built. It contains essential information for the development of the IXP in two parts. The first is *site* information, this is used by the software to identify itself during the software build phase, for example, to configure the Name Server (*NS*). Attributes stored in the schema *site* table is illustrated in Figure 32.

- ***Site type*** : Is the site a cIXP or mIXP, *core* or *mini*
- ***Switching type*** : Type of switching, *traditional* or *software-defined*
- ***Country*** : The country name where the IXP sits
- ***Town*** : The city or town where the site is located
- ***Elevation*** : The height of the IXP above sea level
- ***Organisation*** : The organisation who runs the IXP
- ***Latitude*** : Site degrees, minutes and seconds and North or South
- ***Longitude*** : Site degrees, minutes and seconds and East or West

Figure 32: IXP Schema site table

The second table of information is specific to the configuration of the site and is based on a unique site number. Typically the cIXP is site number 1 and the remaining mIXP sites are numbered 2, 3, … The IXP build engineer specifies the IXP ASN, IPv4 and IPv6 peering and management LAN prefixes as well as an IXP domain-name. From this the schema is built, this process extrapolates individual IPv4 and IPv6 addresses for the core server, the *NS*, Route Collector Server (*CS*), Route Server (*RS*), AS112 Blackhole Server (*BS*) and *SC* on both the peering and management LANs as well as assigning the ASN and domain-name to them. This address schema can be visualised in Figure 30 on page 100. Database entries are generated for each IP address in the IPv4 and IPv6 tables for both the peering and management LANs. Take for example the IPv4 peering LAN. The table attribute names are listed in Figure 33.

- ***Address*** : The IP address and subnet mask
- ***Switch number*** : The port assigned to a switch, internal or external
- ***Port number*** : Port number on the switch
- ***Name*** : Name assigned to the device this IP address represents
- ***Assigned*** : The core device assigned to the IP address
- ***ASN*** : ASN associated with the IP address
- ***Domain name*** : Domain name is associated with this IP address
- ***Route server*** : Does the ISP with this IP address want to peer with the *RS*
- ***AS112 server*** : Does the ISP with this IP address want the AS112 service
- ***Function*** : Is this IP address associated with core or peer devices
- ***Enabled*** : 0 = not enabled, 1 = enabled, 2 = peer assigned
- ***Reserved*** : 0 = not reserved, 1 = reserved

Figure 33: IP peer database structure

After the IXP schema is built, there are a number of IP addresses reserved and configured for internal *core* functions as is demonstrated in Figure 34.

```
cIXP SQL> select * from ipv4_peer where function = 'core';
199.9.9.1/24|100||rs1|rs1|5999|netlabs.tst|||core|1|1
199.9.9.230/24|100||lxd1|lxd1|5999|netlabs.tst|||core|1|1
199.9.9.231/24|100||ns1|ns1|5999|netlabs.tst|||core|1|1
199.9.9.232/24|100|||||||core|0|1
199.9.9.233/24|100||cs1|cs1|5999|netlabs.tst|||core|1|1
199.9.9.234/24|100||bs1|bs1|5999|netlabs.tst|||core|1|1
199.9.9.235/24|100||sc1|sc1|5999|netlabs.tst|||core|0|1
```

Figure 34: IPv4 peering LAN, Core schema

The remaining ports are given a function of *peer* and are divided into two groups, the larger group from host 2 to 229 remain unassigned and are not reserved, while a second group from host 236 to 254 are unassigned or reserved as demonstrated in Figure 35. This second group of IP addresses will remain unused because their corresponding addresses in the IPv4 management LAN are reserved for external switches that maybe connected to the site to add additional capacity as can be seen in Figure 36.

```
cIXP SQL> select * from ipv4_peer where function = 'peer';
199.9.9.2/24|||||||||peer||0
199.9.9.3/24|||||||||peer||0
.. .. .. .. .. .. .. .. .. ..

199.9.9.253/24|||||||||peer||0
199.9.9.254/24|||||||||peer||0
```

Figure 35: IPv4 peering LAN, peer schema

```
cIXP SQL> select * from ipv4_man where reserved = 1;
198.8.8.230/24|100||lxd1|lxd1|5999|netlabs.tst|||core|1|1
198.8.8.231/24|100||ns1|ns1|5999|netlabs.tst|||core|1|1
198.8.8.232/24|100||rs1|rs1|5999|netlabs.tst|||core|1|1
198.8.8.233/24|100||cs1|cs1|5999|netlabs.tst|||core|1|1
198.8.8.234/24|100||bs1|bs1|5999|netlabs.tst|||core|1|1
198.8.8.235/24|100||sc1|sc1|5999|netlabs.tst|||core|0|1
198.8.8.236/24|||||||||core||1
198.8.8.237/24|||||||||core||1
.. .. .. .. .. .. .. .. .. ..

198.8.8.253/24|||||||||core||1
198.8.8.254/24|||||||||core||1
```

Figure 36: IPv4 management LAN, core schema

5.4.3 IXP Host

The IXP host that supports *traditional* Ethernet switches is built using one of five models. Some of these models are based on the information supplied in the IXP schema while others are defined by the IXP build engineer during the host build step. As an example consider the build of an IXP with a server incorporating eight physical Ethernet interfaces and a single external *traditional* Ethernet switch connected to it.

In this case the IXP build engineer is offered three models;

- ***Model C*** : OOB, Management, and remaining peer interfaces,
- ***Model D*** : OOB, Management, one trunk interface and remaining peer interfaces,
- ***Model E*** : OOB, Management, and remaining trunk interfaces.

For the purpose of this description, model D as illustrated in Figure 37, has been chosen for demonstration purposes as this model includes both trunk and peer interface elements and therefore aids in the full understanding of system functionality. Model D breaks the Ethernet interfaces into

four groups, the first interface is considered an OOB management interface, it is the only interface remaining within the Ubuntu *Netplan* and is configured for DHCP and SLAAC configuration by default. This interface is designed to operate as a backup connection to the IXP site using a secondary communications medium like 4G Long Term Evolution (LTE) or 5G New Radio (NR) for remote access when the management LAN is unavailable. The second interface is reserved for connection to the management LAN. The IXP management transit provider provides service on it.

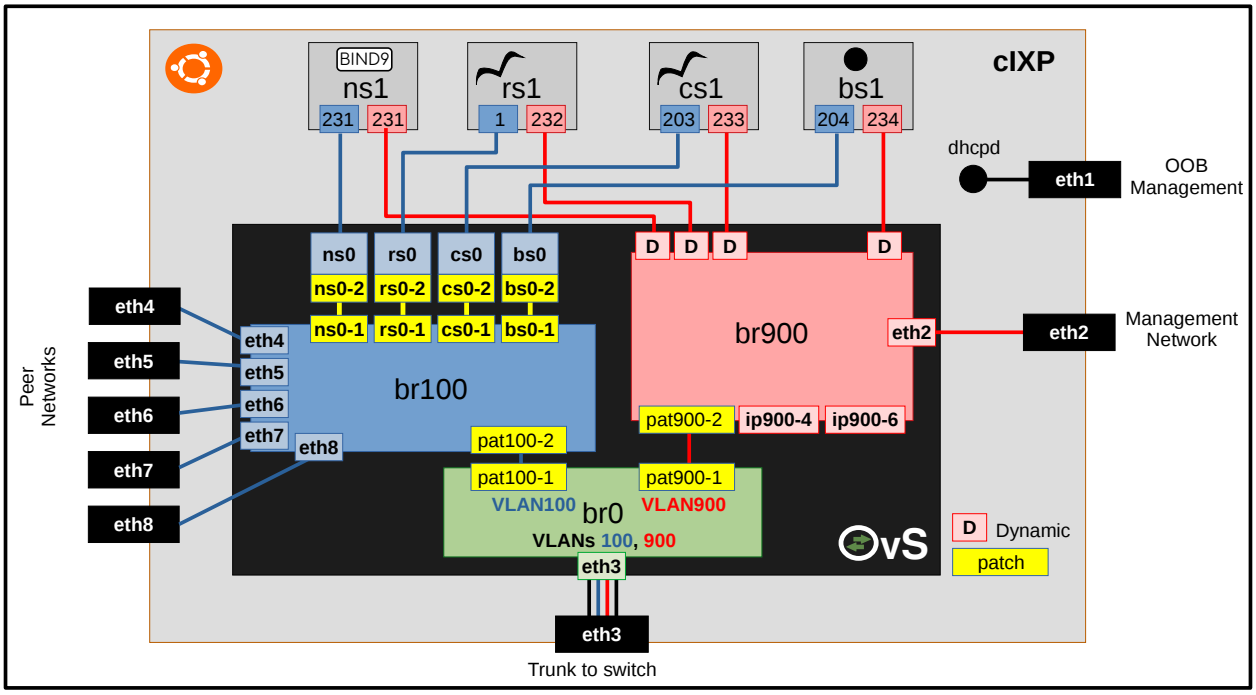


Figure 37: Traditional switching, Model D block diagram

The third interface is configured for VLAN trunk operation with an external *traditional* Ethernet switch. The VLAN trunk will pass over frames with embedded VLAN tags 100 and 900 for the peering and management LANs respectfully.

The remaining interfaces are configured as interfaces connected to the peering LAN and each are assigned to an IP address in the database starting with the first free address, these addresses become *reserved* and the *enabled* value is is set to 2. Switch 0, represents the internal OvS and the ports represent the actual position of the interface on the server, ports 4 – 8 are demonstrated in Figure 38.

```
cIXP SQL> select * from ipv6_peer where function = 'peer' and reserved = 1;
2a99:9:9::2/48|0|4|||||||peer|1|1
2a99:9:9::3/48|0|5|||||||peer|1|1
2a99:9:9::4/48|0|6|||||||peer|1|1
2a99:9:9::5/48|0|7|||||||peer|1|1
2a99:9:9::6/48|0|8|||||||peer|1|1
```

Figure 38: Model D, IPv6 peering ports

Additionally during this build process a table called *host* is added to the database that maintains a description of the host layout, post the build, this information can be recalled at any time by the IXP build engineer (Illustration 9, Appendix A).

As a single external *traditional* Ethernet switch was included in the build, a new *switch* table is created which includes minimal information, a default name *sw_1* and the first free address from the management LAN switch pool of IP addresses, host 241. This IP address also becomes both *enabled* and *reserved* in the IPv4 and IPv6 management LAN tables.

The *ixp host* process (Section 6, Appendix A) then builds start and stop services for LXD and OvS which become managed by the host OS system daemon (*systemd*) process. These services can be restarted in the case of a failure by *systemd*. The LXD start/stop scripts are simple in nature, allowing *systemd* to call on LXD to start the containers *ns1*, *rs1*, *cs1* and *bs1* which have yet to be built. However, the OvS start script in particular is more complex, it builds the logical networks demonstrated in Figure 30 on page 100. The OvS start script builds three main logical bridges plus four bridges associated with each LXC. These are listed in Figure 39.

Each of the peering interfaces are directly connected to the OvS bridge *br100*, the peering LAN as well as a patch connection to each LXC peering bridge. This gives a persistent name, *ns0-1*, *rs0-1*, *cs0-1* and *bs0-1* to access each LXC container, instead of the dynamic *veth* interface name which would be assigned by the LXD. The dynamic *veth* interface name is hidden within each LXC bridge *ns0*, *rs0*, *cs0* and *bs0*. The final patch connection from *br100* is *pat100-2* which connects to the corresponding *pat100-1* on *br0*. On bridge *br0* this port is configured as an access port with a VLAN tag of *100*.

- ***br100*** : Peering LAN
- ***br900*** : Management LAN
- ***br0*** : Trunking LAN
- ***ns0, rs0, cs0, bs0*** : Bridges to link the *veth* peering ports of LXC to br100

Figure 39: LXD bridges for LXC networking in traditional models

The management bridge *br900* also has a connection to each of the containers *ns1*, *rs1*, *cs1* and *bs1*; however, in this case there is no requirement for specific identifiers and therefore each are assigned dynamic *veth* interface names upon boot within bridge *br900*. The physical management interface is connected to this bridge also. Like the peering LAN, a patch to *br0* is created which also acts as a VLAN access port and traffic is given the VLAN tag *900*.

br0 has a third connection to the physical interface, *eth3* which is attached to the first interface on the external *traditional* Ethernet switch. This port is a VLAN trunk and will only pass VLAN tags *100* and *900*. In sub-model D_2 and model E this would be many interfaces.

5.4.4 IXP External Switch

During the *ixp host* process (Section 6, Appendix A) the external *traditional* Ethernet switch is added to the database; however, it is not defined in detail. The IXP switch process allows the IXP build engineer to further refine the switch definition. A more specific arbitrary *name*, a *manufacturer* name and *switch type* can be configured but the mandatory configuration for *traditional* models is the number of *switch interfaces*, for example *48* as the core tier has no way to obtain this information automatically. Additionally the IXP build engineer can define interface speeds which is important when peers are being added. While this information is updated in the *switch* table in the database, the main action from this process is updating the IPv4 and IPv6 peering LAN tables. The first interface on the switch is reserved for the trunk function, the second interface is connected to the management VLAN so the next free 46 IP addresses are *enabled*, *reserved* and associated with interface numbers 3 – 48 as demonstrated in Figure 40.

```
cIXP SQL> select * from ipv4_peer where switch_number is 0 or switch_number is
1;
199.9.9.2/24|0|3|isp1|isp1|5111|one.com|1G|yes|yes|peer|2|1
199.9.9.3/24|0|4|isp2|isp2|5222|two.com|1G|yes|yes|peer|2|1
199.9.9.4/24|0|5|isp3|isp3|5333|three.com|1G|yes|yes|peer|2|1
199.9.9.5/24|0|6|||||1G|||peer|1|1
199.9.9.6/24|0|7|||||10G|||peer|1|1
199.9.9.7/24|0|8|||||10G|||peer|1|1
199.9.9.7/24|1|3|||||10G|||peer|1|1
199.9.9.8/24|1|4|||||10G|||peer|1|1
.. .. .. .. .. .. .. .. .. ..

199.9.9.51/24|1|47|||||10G|||peer|1|1
199.9.9.52/24|1|48|||||10G|||peer|1|1
```

Figure 40: External switch ports on IPv4 peering LAN

5.4.5 IXP Server

The *ixp server build* process (Section 8, Appendix A) checks for a local image of the Ubuntu 18.04 Long Term Support (LTS), if it doesn't exist, it is downloaded and used as a template to clone the containers *ns1*, *rs1*, *cs1* and *bs1*. Each container is configured with peering and management IPv4 and IPv6 address extracted from the database. The OvS is restarted via *systemd* using *systemctl* and Internet connectivity from each is confirmed.

The OvS configuration is quite large so it is broken down into four parts for the purpose of explanation. As demonstrated in Figure 41, each of the containers *ns1*, *rs1*, *cs1* and *bs1* have associated bridges *ns0, rs0, cs0* and *bs0* built in OvS. Each of these bridges have a bridge internal port, a patch port that connects to a corresponding patch port in *br100* and a dynamic port that is associated with the first network interface, *ens3* on the LXC associated with that bridge. Take *ns1* for example, as demonstrated in Figure 42, the interface *ens3* has been assigned the IPv4 and IPv6 addresses reserved for it from the IXP schema. This interface is associated with the dynamic port *veth3KSB19* within the associated bridge *ns0*.

```
ubuntu@cIXP:~$ sudo ovs-vsctl show
[sudo] password for ubuntu:
e0c89665-cd2e-4cfd-b168-436f18ab05ba
    Bridge "ns0"
        Port "ns0"
            Interface "ns0"
                type: internal
        Port "ns0-2"
            Interface "ns0-2"
                type: patch
                options: {peer="ns0-1"}
        Port "veth3KSB19"
            Interface "veth3KSB19"
    Bridge "rs0"
        Port "rs0"
            Interface "rs0"
                type: internal
        Port "rs0-2"
            Interface "rs0-2"
                type: patch
                options: {peer="rs0-1"}
        Port "vethVVG8KR"
            Interface "vethVVG8KR"
    Bridge "cs0"
        Port "cs0"
            Interface "cs0"
                type: internal
        Port "cs0-2"
            Interface "cs0-2"
                type: patch
                options: {peer="cs0-1"}
        Port "veth3NEABE"
            Interface "veth3NEABE"
    Bridge "bs0"
        Port "bs0"
            Interface "bs0"
                type: internal
        Port "bs0-2"
            Interface "bs0-2"
                type: patch
                options: {peer="bs0-1"}
        Port "veth17A1QO"
            Interface "veth17A1QO"
```

Figure 41: OvS configuration for traditional models - Part 1

```
root@ns1:~# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever

65: ens3@if66: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default qlen 1000
    link/ether 00:16:3e:94:00:a1 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 199.9.9.231/24 brd 199.9.9.255 scope global ens3
       valid_lft forever preferred_lft forever
    inet6 2a99:9:9::231/48 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::216:3eff:fe94:a1/64 scope link
       valid_lft forever preferred_lft forever

67: ens4@if68: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default qlen 1000
    link/ether 00:16:3e:e9:c9:7a brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 198.8.8.231/24 brd 198.8.8.255 scope global ens4
       valid_lft forever preferred_lft forever
    inet6 2a98:8:8::231/48 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::216:3eff:fee9:c97a/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 42: ip address show on ns1

The second part of the OvS configuration, as demonstrated in Figure 43, develops the configuration of *br100*, the peering LAN. This bridge includes an internal port, patch ports to connect to the the bridges *ns0*, *rs0*, *cs0* and *bs0*, a patch connection to the trunk bridge *br0* plus individual ports associated with all the physical peering interfaces 4 – 8 on the server.

```
    Bridge "br100"
        Port "br100"
            Interface "br100"
                type: internal
        Port "ns0-1"
            Interface "ns0-1"
                type: patch
                options: {peer="ns0-2"}
        Port "rs0-1"
            Interface "rs0-1"
                type: patch
                options: {peer="rs0-2"}
        Port "cs0-1"
            Interface "cs0-1"
                type: patch
                options: {peer="cs0-2"}
        Port "bs0-1"
            Interface "bs0-1"
                type: patch
                options: {peer="bs0-2"}
        Port "pat100-2"
            Interface "pat100-2"
                type: patch
                options: {peer="pat100-1"}
        Port "eth4"
            Interface "eth4"
        Port "eth5"
            Interface "eth5"
        Port "eth6"
            Interface "eth6"
        Port "eth7"
            Interface "eth7"
        Port "eth8"
            Interface "eth8"
```

Figure 43: OvS configuration for traditional models - Part 2

The third part of the OvS configuration is the management bridge *br900* as shown in Figure 44. This bridge has a default internal port as well as two configured internal ports, *ip900-4*, *ip900-6*. These ports represent interfaces that are used for local server IP addresses, as demonstrated in Figure 45. Like *br100* this bridge has a patch connection to *br0* for trunk access. This gives bridge *br900* a connection to the second interface on external *traditional* Ethernet switches which are configured with a VLAN *900* tag. Each of the containers *ns1*, *rs1*, *cs1* and *bs1* have dynamic *veth* ports within *br900*. A mapping between the interface index for interface *ens4*, the management interface, on each container and the associated *veth* interface in br900 can be determined. Take *ns1* for example, see Figure 42, *ifindex 68* from *ens4@if68* can be mapped to the dynamic interface *vethF0X85V* as demonstrated in Figure 46. So in this example the interface *vethF0X85V* is associated with the container *ns1*. Finally the physical interface *eth2* has a port in *br900.*

```
    Bridge "br900"
        Port "br900"
            Interface "br900"
                type: internal
        Port "ip900-4"
            Interface "ip900-4"
                type: internal
        Port "ip900-6"
            Interface "ip900-6"
                type: internal
        Port "pat900-2"
            Interface "pat900-2"
                type: patch
                options: {peer="pat900-1"}
        Port "vethDYT0IH"
            Interface "vethDYT0IH"
        Port "vethF0X85V"
            Interface "vethF0X85V"
        Port vethFMWVGD
            Interface vethFMWVGD
        Port "vethS9CQXV"
            Interface "vethS9CQXV"
        Port "eth2"
            Interface "eth2"
```

Figure 44: OvS configuration for traditional models - Part 3

```
ubuntu@cIXP:~$ ip address show dev ip900-4
44: ip900-4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/ether 66:55:21:f8:ec:4f brd ff:ff:ff:ff:ff:ff
    inet 198.8.8.230/24 scope global ip900-4
       valid_lft forever preferred_lft forever
    inet6 fe80::6455:21ff:fef8:ec4f/64 scope link
       valid_lft forever preferred_lft forever

ubuntu@cIXP:~$ ip address show dev ip900-6
45: ip900-6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/ether 3a:08:d7:87:8b:1a brd ff:ff:ff:ff:ff:ff
    inet6 2a98:8:8::230/48 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::3808:d7ff:fe87:8b1a/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 45: IP addresses on lxd1

```
ubuntu@cIXP:~$ sudo ovs-vsctl find interface ifindex=68 | grep '^name'
name                : "vethF0X85V"
```

Figure 46: Map interface index to veth port

The final part of the OvS configuration is the trunk bridge *br0* demonstrated in Figure 47. Like the other bridges it has an internal port. It has two patch ports that connect to *br100* and *br900* respectfully. These are VLAN access ports and they strip VLAN tags for ingress frames while adding tags for egress frames, *100* on *pat100-1* and *900* on *pat900-1*. The last port connects to the physical *eth3* interface. This is a VLAN trunk interface for both VLAN tags *100* and *900* and it therefore passes the tags transparently via the first interface which is the trunk interface on the external *traditional* Ethernet switch. While in sub-model D_1 there is just a single trunk interface, in sub-model D_2 and model E there would be many trunk interfaces.

```
    Bridge "br0"
        Port "br0"
            Interface "br0"
                type: internal
        Port "pat100-1"
            tag: 100
            Interface "pat100-1"
                type: patch
                options: {peer="pat100-2"}
        Port "pat900-1"
            tag: 900
            Interface "pat900-1"
                type: patch
                options: {peer="pat900-2"}
        Port "eth3"
            trunks: [100, 900]
            Interface "eth3"
    ovs_version: "2.9.2"
```

Figure 47: OvS configuration for traditional models - Part 4

5.4.6 IXP Software

The *ixp software* process (Section 9, Appendix A) is handled in two process, one process to install the software and a second process to configure it. The PoC offers the option to reinstall and reconfigure the software on the core servers individually which is very helpful when testing. In a production environment, this functionality could be merged to simplify the processes. The initial step involves checking the Internet connectivity, this is because the overall process involves upgrading each container instance from the Internet repositories and then installing the software appropriate to them.

The following main packages are installed, either from the the IXPBuilder files or from the Ubuntu repository over the Internet along with their associated dependencies;

- ***NS*** : bind9, bind9utils,
- ***RS*** : bird, birdseye,
- ***CS*** : bird, birdseye,
- ***BS*** : bind9, bind9utils, bird, birdseye.

The *bind9* and *bind9utils* packages provide the Berkeley Internet Name Domain version 9 (BIND9) Domain Name System (DNS) while *bird* and *birdseye* packages provide the Bird Internet Routing Daemon (BIRD) router and a Representational State Transfer (RESTful) Application Programming Interface (API) Internet micro-service to access it.

5.4.6.1 The Nameserver

The initial process configures *ns1* with an upstream DNS address. In the PoC this is set to 8.8.8.8, the Google DNS server, by default. This default can be changed in the header section of the *ixp.py* class and function library module file if necessary. The network details for *ns1* are imported from the IXP schema and applied to the *Netplan* for the container along with the upstream DNS. The container is then upgraded and *bind9*, *bind9utils* and their dependencies are installed.

In the second step of the process, software configuration, the relevant data is extracted from the database. Tests are performed to ensure that *bind9* has been installed and that the */var/log/named/* directory exists. It then creates the */etc/bind/zones/* and */var/log/named/* directories as well as generating the *bind9.log* file in */var/log/named/* directory.

The */etc/bind/named.conf.options* and */etc/bind/named.conf.local* files are configured with the data from the IXP schema and the zone files *db.<prefix>* and *db.<domain>*, for example *db.198.8.8*, *db.2a98:8:8* and *db.netlabs.tst* are created with Pointer (PTR) records for the other LXCs within the IXP core.

When the configuration is complete the *bind9* daemon on *ns1* is restarted by *systemd* via

systemctl. Each of the other containers are then reconfigured to replace the default Google DNS server *8.8.8.8* with the newly configured DNS Server at the IP address, *198.8.8.231*. This configuration can be confirmed by connecting to any of the other containers, verify the DNS server IP addresses in use are the addresses of *ns1* and test connectivity as demonstrated in Figure 48.

```
root@rs1:~# systemd-resolve --status --no-pager | grep 'DNS Servers'
         DNS Servers: 198.8.8.231
         DNS Servers: 199.9.9.231

root@rs1:~# fping cs1
cs1 is alive

root@rs1:~# fping cs1.netlabs.tst
cs1.netlabs.tst is alive

root@rs1:~# fping www.ubuntu.com
www.ubuntu.com is alive
```

Figure 48: Test the DNS server ns1 from another container

5.4.6.2 The Route Collector and Route Server

There is very little difference between *cs1* and *rs1* servers except for some minor configuration in the *bird.conf* and *bird6.conf* files, so for the purpose of description they will be taken together. The network details for each are imported from the IXP schema and applied to the *Netplan* for the container along with the temporary global upstream DNS address. This is restored back to the local DNS at the end of the process. The container is updated from the repositories and the *bird* and *birdseye* packages are installed along with their dependencies, for example, *php* and *lighttpd* to support the *birdseye* micro-service.

The second step in the process, the configuration involves creating both */etc/bird.conf* and */etc/bird6.conf* files as well as *martian* filter files for IPv4 and IPv6 on the containers. These configurations define the local ASN, 5999 and the IP address of the containers interface within the peering LAN. They also link to uploaded import filter policies for IPv4 and IPv6 from RFC 6890 and RFC 8190 Special-Purpose Address Registries. The configuration file for the route collector *cs1* is configured to *import all* routes but not to *export* routes, whereas the route server *rs1* is configured to both *import* and *export* routes.

The *bird* and *bird6* daemons are then restarted on the containers by *systemd* via *systemctl*. Figure 49 shows the status for the *bird* daemon on the *RS* and the status of the *bird6* daemon on the *CS* as examples to demonstrate that both daemons are running on both the *RS* and *CS*.

```
root@rs1:~# birdc show status
BIRD 1.6.3 ready.
BIRD 1.6.3
Router ID is 199.9.9.1
Current server time is 2019-04-18 11:46:29
Last reboot on 2019-04-18 11:19:21
Last reconfiguration on 2019-04-18 11:19:21
Daemon is up and running

root@cs1:~# birdc6 show status
BIRD 1.6.3 ready.
BIRD 1.6.3
Router ID is 199.9.9.233
Current server time is 2019-04-18 11:47:21
Last reboot on 2019-04-18 11:19:24
Last reconfiguration on 2019-04-18 11:19:24
Daemon is up and running
```

Figure 49: BIRD daemon status

Additional to the BIRD daemons, the *Birdseye* application is also installed on the servers. This provides a REStful API micro-service to the BIRD daemons (Section 13, Appendix A). As there are daemon APIs on each container for each IP family type, IPv4 and IPv6, it is necessary to have Canonical NAME (CNAME) records added as shown in Figure 50 to the domain zone file. This maps alias names for each individual BIRD daemon to access them via the micro-service on each container *CS*, *RS* and *BS*.

```
root@ns1:~# tail -10  /etc/bind/zones/db.netlabs.tst

; Added for Birdeye
rs1-ipv4    IN    CNAME    rs1.netlabs.tst.
rs1-ipv6    IN    CNAME    rs1.netlabs.tst.
cs1-ipv4    IN    CNAME    cs1.netlabs.tst.
cs1-ipv6    IN    CNAME    cs1.netlabs.tst.
bs1-ipv4    IN    CNAME    bs1.netlabs.tst.
bs1-ipv6    IN    CNAME    bs1.netlabs.tst.
```

Figure 50: Birdseye additions to domain zone file

The API returns data in JavaScript Object Notation (JSON) format and can be accessed using any Internet browser that supports the display of JSON. However, the real power in the

RESTful API lies in the exposure of this data interface for the development of additional tools for automation and monitoring.

```
ubuntu@cIXP:~$ cat birdseye_rest_api.py
1  #! /usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import urllib.request
5  import json
6
7  str_ = str()
8  url_ = 'http://rs1-ipv4.netlabs.tst/api/status'
9  dict_ = dict()
10 tab = 15
11
12 f = urllib.request.urlopen(url_).read()
13 str_ = f.decode('utf-8')
14 dict_ = json.loads(str_)
15
16 print(f'\n{str_}\n')
17
18 for k, v in dict_['status'].items():
19    k = k.replace('_', ' ').title()
20    print(f'{k:<{tab}}: {v}')
```

Figure 51: Birdseye RESTful API status python snippet

As an example, consider the simple python3 program snippet in Figure 51. Now consider the output of direct access from the shell command *cURL* and the output from the program snippet in Figure 52. The program reads the string in JSON format in a similar way to the *cURL* command. Using the python JSON module `loads()` method the string is parsed into a dictionary and can therefore be manipulated easily as shown in this simple example. In line 16 the string is printed to verify that the program has the same data as that returned from the *cURL* command. The last three lines of the program output the same data in a more human readable format to STDOUT.

```
ubuntu@cIXP:~$ curl rs1-ipv4.netlabs.tst/api/status; echo
{"api":
{"from_cache":true,"ttl_mins":1,"version":"1.1.4","max_routes":1000},"statu
s":{"version":"1.6.3","router_id":"199.9.9.1","server_time":"2019-04-
20T08:42:49+00:00","last_reboot":"2019-04-
20T07:17:56+00:00","last_reconfig":"2019-04-
20T07:17:56+00:00","message":"Daemon is up and running"}}

ubuntu@cIXP:~$ ./birdseye_rest_api.py

{"api":
{"from_cache":true,"ttl_mins":1,"version":"1.1.4","max_routes":1000},"statu
s":{"version":"1.6.3","router_id":"199.9.9.1","server_time":"2019-04-
20T08:46:17+00:00","last_reboot":"2019-04-
20T07:17:56+00:00","last_reconfig":"2019-04-
20T07:17:56+00:00","message":"Daemon is up and running"}}

Version         : 1.6.3
Router Id       : 199.9.9.1
Server Time     : 2019-04-20T08:46:17+00:00
Last Reboot     : 2019-04-20T07:17:56+00:00
Last Reconfig   : 2019-04-20T07:17:56+00:00
Message         : Daemon is up and running
```

Figure 52: Compare cURL and the simple Birdseye python snippet output

There are many commands that can be extracted from the RESTful API in this manner and a list can be obtained via the root Uniform Resource Locator (URL) for the server. For example to extract the IPv4 daemon data from the *RS* container use the URL;

http://rs1-ipv4.netlabs.tst

Table 8 outlines some of useful examples:

Table 8: Birdseye RESTful API URL examples

RESTful API URL	Function
cs1-ipv4.netlabs.tst/api/protocols/bgp	Information on ISPs using BGP on *cs1*
rs1-ipv4.netlabs.tst/api/protocol/ISP1	Information about the ISP1 on *rs1*
cs1-ipv4.netlabs.tst/api/routes/protocol/ISP2	Routes for ISP2 in *cs1*
rs1-ipv4.netlabs.tst/api/routes/table/master	Routes table for *rs1*
cs1-ipv4.netlabs.tst/api/route/199.1.1.0	Routes for a specific network on *cs1*

5.4.6.3 The AS112 Blackhole service

Many sites connected to the Internet make use of IP addresses that are not globally unique. Examples include the well known 192.168.0.0/16 range from RFC 1918. Sometimes reverse-lookup

queries are directed to DNS servers outside local networks and it is obviously not possible for public DNS servers to respond with answers. Widespread deployment of private address space has increased the number of such queries. The AS112 Blackhole service was assigned the ASN 112 by Internet Assigned Numbers Authority (IANA) to support a distributed DNS sink service to handle misdirected DNS queries. The service is a distributed blackhole ran mainly in IXPs but also in some ISPs and other Internet organisations.

All AS112 servers are configured with the same IPv4 and IPv6 addresses as shown in Figure 53, such that, it appears on the Internet as an *anycast* type service.

```
root@bs1:~# cat /etc/netplan/10-netplan.yaml | grep -A11 'bridges:'
  bridges:
    as112_br1:
      interfaces: [dummy1]
      addresses:
          - 192.175.48.1/24
          - 192.175.48.6/24
          - 192.175.48.42/24
          - 192.31.196.1/24
          - 2620:4f:8000::1/48
          - 2620:4f:8000::6/48
          - 2620:4f:8000::42/48
          - 2001:4:112::1/48

root@bs1:~# ip address show dev as112_br1 | grep -E '(inet 1|inet6 2)'
    inet 192.175.48.1/24 brd 192.175.48.255 scope global as112_br1
    inet 192.31.196.1/24 brd 192.31.196.255 scope global as112_br1
    inet 192.175.48.6/24 brd 192.175.48.255 scope global secondary as112_br1
    inet 192.175.48.42/24 brd 192.175.48.255 scope global secondary as112_br1
    inet6 2001:4:112::1/48 scope global
    inet6 2620:4f:8000::42/48 scope global
    inet6 2620:4f:8000::6/48 scope global
    inet6 2620:4f:8000::1/48 scope global
```

Figure 53: AS112 container IP address assignments

The BIRD router daemon advertises these direct routes to the peering members as can be seen in Figure 54 while Figure 55 demonstrates that these routes are learnt by the peering routers.

```
root@bs1:~# birdc show route | grep as112_br1
192.175.48.0/24    dev as112_br1 [direct1 09:31:54] * (240)
192.31.196.0/24    dev as112_br1 [direct1 09:31:54] * (240)

root@bs1:~# birdc6 show route | grep as112_br1
2001:4:112::/48    dev as112_br1 [direct1 09:31:54] * (240)
2620:4f:8000::/48  dev as112_br1 [direct1 09:31:54] * (240)
```

Figure 54: AS112 routes in BIRD routing table of bs1

```
ISP3_RTR# show ip route bgp
B     192.31.196.0/24 [20/0] via 199.9.9.234, 01:36:41
B     192.175.48.0/24 [20/0] via 199.9.9.234, 01:36:41
B     199.1.1.0/24 [20/0] via 199.9.9.2, 01:36:14
B     199.2.2.0/24 [20/0] via 199.9.9.3, 01:36:42

ISP3_RTR# show ipv6 route bgp
B   2001:4:112::/48 [20/0]
     via FE80::216:3EFF:FECE:780C, GigabitEthernet0/0
B   2620:4F:8000::/48 [20/0]
     via FE80::216:3EFF:FECE:780C, GigabitEthernet0/0
B   2A99:1:1::/48 [20/0]
     via FE80::223:5EFF:FE0E:6816, GigabitEthernet0/0
B   2A99:2:2::/48 [20/0]
     via FE80::21E:BEFF:FE17:EB9A, GigabitEthernet0/0
```

Figure 55: AS112 routes at ISP3 router

IANA have established blackhole DNS servers as can be seen in Figure 56. The BIND service on *BS* is configured as an *authoritative-only* server with no recursion. It is also configured to *listen-on* the well known AS112 anycast nameserver addresses as established by IANA as well as its local addresses. The servers are configured to respond to any query with a *non-existent address* answer in a manner that is consistent with the DNS protocol such that it is cached by recursive DNS servers. This serves to both reduce lookup wait times as well as reduce network load.

```
root@bs1:/# less /etc/bind/named.conf.options | grep anycast
           192.175.48.1;      // prisoner.iana.org (anycast)
           192.175.48.6;      // blackhole-1.iana.org (anycast)
           192.175.48.42;     // blackhole-2.iana.org (anycast)
           192.31.196.1;      // blackhole.as112.arpa (anycast)
           2620:4f:8000::1;   // prisoner.iana.org (anycast)
           2620:4f:8000::6;   // blackhole-1.iana.org (anycast)
           2620:4f:8000::42;  // blackhole-2.iana.org (anycast)
           2001:4:112::1;     // blackhole.as112.arpa (anycast)
```

Figure 56: IANA authoritative DNS servers to support AS112 service

IP addresses from the following private address space blocks are directed to the zone *db.dd-empty* and receive a response consisting of the nameservers *blackhole-1.iana.org* and *blackhole-2.iana.org.*

- 10.0.0.0/8,
- 172.16.0.0/12,
- 192.168.0.0/16,
- 169.254.0.0/16.

Figure 57 demonstrates this for the 192.168.0.0/16 block.

```
root@bs1:~# cat /etc/bind/named.conf.local | grep 192
zone "168.192.in-addr.arpa" { type master; file "/etc/bind/zones/db.dd-
empty"; };

root@bs1:~# cat /etc/bind/zones/db.dd-empty | grep NS
        NS      blackhole-1.iana.org.
        NS      blackhole-2.iana.org.
```

Figure 57: AS112 Direct Delegation

It is essential for the AS112 Blackhole services to work successfully that any *NS* resolvers on ISP networks have connectivity to the *anycast* prefix instances *192.175.48.0/24* and *2620:4f:8000/48* as they exist at the IXP *BS*, otherwise *reverse-lookups* will be performed by other AS112 services elsewhere in the Internet ecosystem. The routing of these prefixes to each participating ISP should ensure this. Confirmation can be made using the network diagnostic tool *mtr* or *traceroute* from the *NS* to the private IP addresses configured on the *BS*. The response should be confined within the ISP and the IXP networks. The example in Figure 58 demonstrates this, it shows that only a single hop exists between the *NS* and *BS*. To further review, the DNS *dig* tool traces the delegation path from the root nameservers by makes iterative queries to resolve the name being looked up by following referrals and showing the answer from each server that was used to resolve the lookup. In each case a *reverse-lookup* for mapping the private address space addresses to names is carried out and in each case traces for the private IP addresses 192.168.1.10 and fd00:1.1::10 return a Non-eXistent DOMAIN (NXDOMAIN) as well as the fact that the response was received from the first DNS server as no other referrals are reported in the path. This indicates that the *NS* received the query result directly from the *BS*.

```
root@ns1:~# mtr --report 192.175.48.1
Start: 2019-04-27T12:07:36+0000
HOST: ns1                     Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- prisoner.iana.org     0.0%    10    0.1   0.1   0.0   0.3   0.1

root@ns1:~# mtr --report 2620:4f:8000::1
Start: 2019-04-27T12:08:47+0000
HOST: ns1                     Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- prisoner.iana.org     0.0%    10    0.1   0.1   0.1   0.5   0.1

root@ns1:~# dig -x 192.168.1.10 | grep status
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 61263

root@ns1:~# dig +trace -x 192.168.1.10
; <<>> DiG 9.11.3-1ubuntu1.7-Ubuntu <<>> +trace -x 192.168.1.10
;; global options: +cmd
;; Received 28 bytes from 127.0.0.53#53(127.0.0.53) in 0 ms

root@ns1:~# dig -x fd00:1.1::10 |grep status
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 18540

root@ns1:~# dig +trace -x fd00:1.1::10
; <<>> DiG 9.11.3-1ubuntu1.7-Ubuntu <<>> +trace -x fd00:1.1::10
;; global options: +cmd
;; Received 28 bytes from 127.0.0.53#53(127.0.0.53) in 0 ms
```

Figure 58: Confirm connectivity from NS to BS

When a reverse-lookup is performed for an address in private address space, for example *192.168.1.10*, a PTR lookup is performed for *10.1.168.192.in-addr.arpa*. The *168.192.in-addr.arpa* zone is delegated to *blackhole-1.iana.org*, and *blackhole-2.iana.org*, which have IP addresses in the *192.175.48.0/24* and *2620:4f:8000::/48* ranges.

To confirm that the AS112 Blackhole nameserver service is operational perform queries against the service as demonstrated in Figure 59. The *BS* nameserver responds with an authoritative NXDOMAIN response which indicates that the IP address is associated with a NXDOMAIN, i.e. it is invalid.

```
host1@ISP1:~# dig @199.9.9.234 -x 192.168.1.10

; <<>> DiG 9.10.3-P4-Raspbian <<>> @199.9.9.234 -x 192.168.1.10
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 49580
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;10.1.168.192.in-addr.arpa.   IN    PTR

;; AUTHORITY SECTION:
168.192.in-addr.arpa.   604800      IN    SOA   prisoner.iana.org.
hostmaster.root-servers.org. 20190424 604800 60 604800 604800

;; Query time: 1 msec
;; SERVER: 199.9.9.234#53(199.9.9.234)
;; WHEN: Thu Apr 25 11:46:39 UTC 2019
;; MSG SIZE  rcvd: 131
```

Figure 59: Test AS112 nameserver with reverse lookup for private IP address

As the *BS* nameserver is authoritative only, *status: REFUSED* responses are received when the requests require recursion as can be seen in Figure 60.

```
root@rs1:~# dig @199.9.9.234 netlabs.tst. A

; <<>> DiG 9.11.3-1ubuntu1.5-Ubuntu <<>> @199.9.9.234 netlabs.tst. A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 52540
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5290d7695f256f2a600cff195cbb1c035c66c03ba354c3fa (good)
;; QUESTION SECTION:
;netlabs.tst.                   IN    A

;; Query time: 1 msec
;; SERVER: 199.9.9.234#53(199.9.9.234)
;; WHEN: Sat Apr 20 13:17:55 UTC 2019
;; MSG SIZE  rcvd: 68

root@rs1:~# dig @2a99:9:9::234 netlabs.tst. AAAA

; <<>> DiG 9.11.3-1ubuntu1.5-Ubuntu <<>> @2a99:9:9::234 netlabs.tst. AAAA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 39942
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 37a6489d8c27f04ccb9eea1f5cbb1d8b8b0dcd33ce68652d (good)
;; QUESTION SECTION:
;netlabs.tst.                   IN    AAAA

;; Query time: 0 msec
;; SERVER: 2a99:9:9::234#53(2a99:9:9::234)
;; WHEN: Sat Apr 20 13:24:27 UTC 2019
;; MSG SIZE  rcvd: 68
```

Figure 60: Test that the AS112 service is authoritative only

5.5 Models employing Software-defined Switching

Both data path, packet forwarding, and control occur on the same device in t*raditional* Ethernet switches. OF switches however, separate these two functions, the OF switch retains the data path packet forwarding function while the control path function is handed off over a control channel to a separate *SC*.

This PoC software builds an SDX through an embedded OvS that is subject to the control of an *SC* and connected external Ethernet switches must be OF compliant. Apart from a minimal

configuration to establish the control-channel IP address and Transmission Control Protocol (TCP) port, external OF Ethernet switches are configured automatically by the *SC*. During the build of such a *software-defined* system the PoC software detects the number of available Ethernet interfaces (ε) on the hardware and the IXP build engineer specifies the number of OF Ethernet switches (σ) required for the SDX ecosystem. It is considered impractical to have *software-defined* models for less than four Ethernet interfaces on the server.

The model to be applied is then selected as follows;

- IF σ = 0, model S is automatically selected.
- ELIF (ε – 2) / 2 > σ , model T is automatically selected.
- ELIF (ε % 2) = 0 AND (ε – 2) / 2 = σ, model U is automatically selected.
- ELSE, too many OF Ethernet switches have been selected, σ must be reduced.

For model S and similar to previous models using *traditional* Ethernet switches C, D and E, the first interface is reserved for the OOB function and the second for connection to the management LAN. The remaining interfaces in model S are established as peer interfaces.

With model T, the first two interfaces are reserved for OOB and the management LAN functions. This model has a mix of connected OF Ethernet switches and local peer interfaces. Each OF Ethernet switch requires two interfaces, one to connect the OF Ethernet switches OOB port to the management LAN and a second to connect to the internal OvS peering LAN, these pairs are called the Control/Data Interface Pair (CDIP). The first interface of this pair of interfaces is used by the *SC* to manage the control plane of the OF Ethernet switch. It is connected to the external OF Ethernet switches OOB management port. While it is possible to incorporate the SDN control-channel in the peering LAN it was decided not to do so because in the case of peering LAN data plane issues arising it could affect control plane messages in the control-channel, leading to a slow or unresponsive control plane which would further aggravate the effect on the data plane. By keeping the control-channel separate in the management LAN this potential problem is avoided.

The second interface in the CDIP is used to connect the data planes of the internal OvS and the external OF Ethernet switch on its first interface. There are two sub-models, T_1 has a single CDIP and the remaining interfaces are peer interfaces whereas the T_2 sub-model refers to any mix where CDIP > 1 and there are some peer interfaces. In the case of a server with an odd number of interfaces and any number of CDIPs, then model T must be employed.

The final model, U also has the first two interfaces reserved for OOB management and the management LAN. This model has no local peer interfaces as all the remaining interfaces are used for CDIP pairs. Therefore this model can only apply to servers with an even number of interfaces.

All the software-defined models and sub-models are summarised in Table 9.

Table 9: Software-defined model summary

Model	Interfaces	Interfaces			
		1	2	3/4	5+
S	> 3	OOB	Management interface	Peer interfaces	
T_1	> 3	OOB	Management interface	OF 1 CDIP	Peer interfaces
T_2	> 3	OOB	Management interface	Mix of CDIPs and Peer interfaces	
U	> 3	OOB	Management interface	CDIPs	

5.5.1 The structure of the PoC as an SDX

Figure 61 is a functional diagram of the system for optimum *software-defined* switching models creating and SDX. The description of the tiers and the layers from the *traditional* PoC functional diagram holds true. However, the external *traditional* Ethernet switches at the *switching tier* are replaced with OF Ethernet switches. There is also an additional LXC container at the *service layer* for the *SC* function which controls OF connections to both the internal OvS and any external OF Ethernet switches in the *switching tier* over the management network. Suggested configuration notes for the OF Ethernet switches and the ISP peering routers for various devices is given in the (Section 15.2, Appendix A)

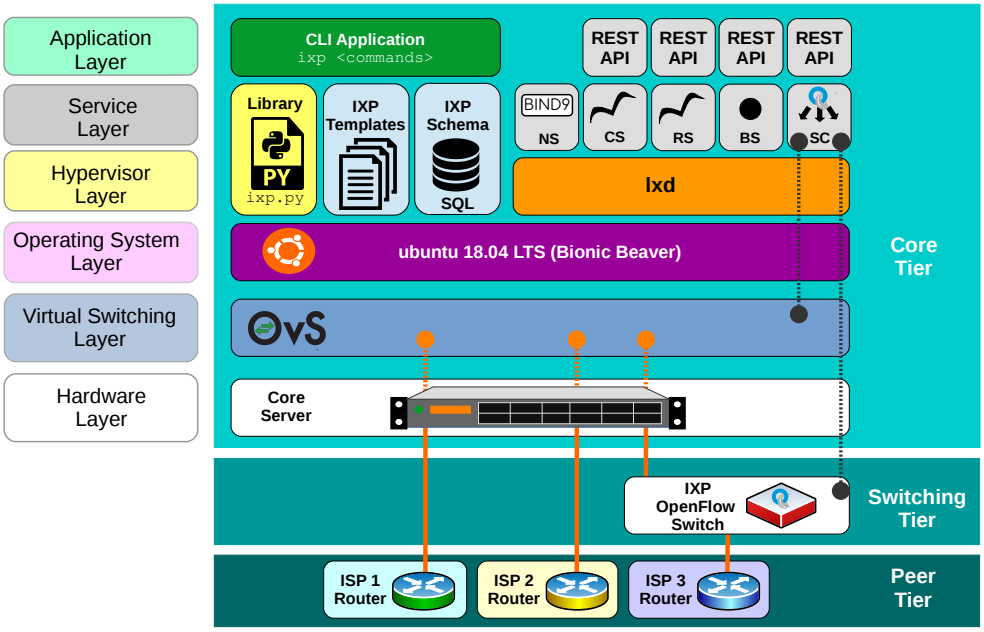


Figure 61: IXPBuilder functional diagram – software-defined

5.5.2 SDX Schema

The SDX schema is the foundation of the IXP, for *software-defined* models there is no difference in the SDX schema make-up as for models employing *traditional* Ethernet switches except that the switching type: *software-defined* in the *site* table as demonstrated in Figure 62.

```
cIXP SQL> select switching_type from site;
software-defined
```

Figure 62: Switching type in site table of schema

The SDX host is built using one of three models. All of these models are determined based on the information supplied in the SDX schema, the switching type: *software-defined* and the number of external OF Ethernet switches defined by the IXP build engineer. Considering the build of an SDX host with a single external OF Ethernet switch and a server with eight physical Ethernet interfaces as an example and outlined in Figure 63, the sub-model T_1 will be built.

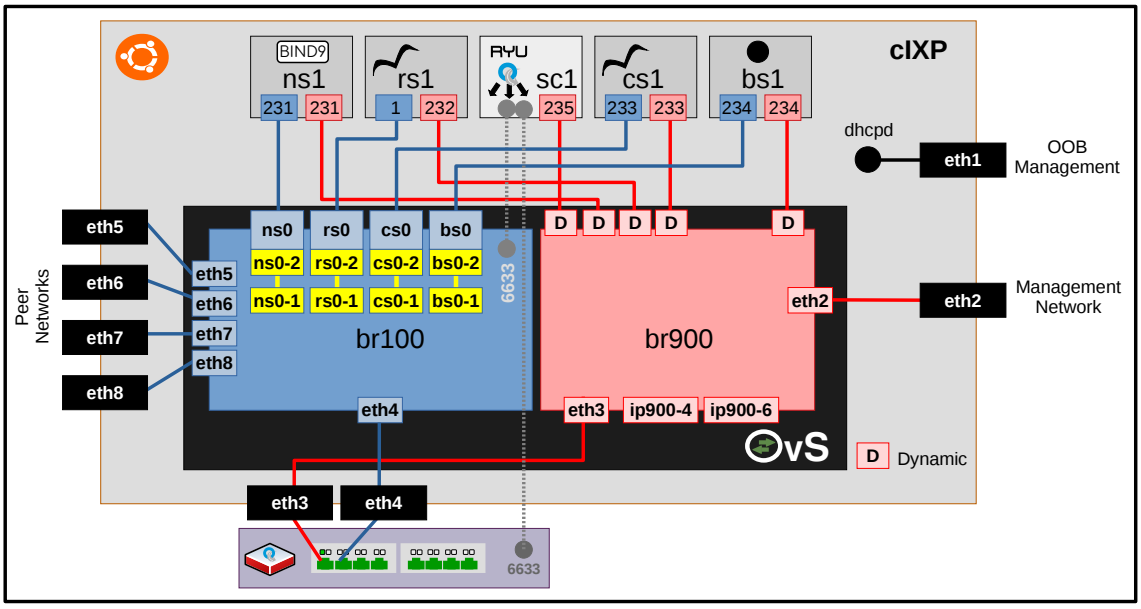


Figure 63: Software-defined switching, Model T block diagram

5.5.3 SDX Host

Model T breaks the Ethernet interfaces into four groups, the first interface is considered an OOB management interface, it is the only interface remaining within the Ubuntu *Netplan* and is configured for DHCP and SLAAC configuration by default. This interface is designed to operate as a backup connection to the SDX site using a secondary communications medium like 4G LTE or 5G NR for remote access when the management LAN is unavailable. The second interface is reserved for connection to the management LAN. The SDX management transit provider provides service on it.

The third and fourth interfaces form a CDIP with the third interface connecting the management bridge *br900* in the internal OvS to the OOB interface of the external OF Ethernet switch. The fourth interface connects the peering bridge *br100* in the internal OvS to the first interface on the external OF Ethernet switch. In this way the OF control-channel remains completely separated from the data plane.

From the database perspective the same tables are updated in an identical fashion to the

traditional model D except there is one less interface for peering as the CDIP takes up two interfaces. This is reflected in the database tables shown in Figure 64, note that there are only four ports reserved for peers to connect as the third (*eth3*) and fourth (*eth4*) interfaces are reserved as the *control* and *data interface* respectfully.

```
cIXP SQL> select * from ipv4_peer where function = 'peer' and reserved = 1;
199.9.9.2/24|0|5|||||||peer|1|1
199.9.9.3/24|0|6|||||||peer|1|1
199.9.9.4/24|0|7|||||||peer|1|1
199.9.9.5/24|0|8|||||||peer|1|1

SQL> select * from switch;
1|sw_1|Bare-metal|Bare-metal|eth3|eth4|0||198.8.8.241/24|2a98:8:8::241/48
```

Figure 64: Model T, IPv4 peering ports and external OF switch entry

As with *traditional* model D, the SDX host process builds start/stop services for LXD and OvS which operate identically to that described earlier, except the LXD script must also manage the additional *sc1* container too. There are similarities between the *traditional* and *software-defined* OvS start/stop scripts; however, there are significant differences to require explanation. The logical networks are demonstrated in the diagram at Figure 63.

The OvS start script builds two main logical bridges plus four bridges associated with each LXC as per the list in Figure 65.

- ***br100*** : Peering LAN
- ***br900*** : Management LAN
- ***ns0, rs0, cs0, bs0*** : Bridges to link the *veth* peering ports of LXC to *br100*

Figure 65: LXD bridges for LXC networking in software-defined models

Like model D, each of the peering interfaces are directly connected to the OvS bridge *br100*, the peering LAN as well as a patch connection to each LXC peering bridge. *ns1*, *rs1*, *cs1* and *bc1* are contained within their own bridges *ns0*, *rs0*, *cs0* and *bc0* which have patch connections to *br100*, for example *ns0* is connected to *br100* via the patch pair *ns0-2*, *ns0-1*. Note that the new container *sc1* is not connected to the peering LAN in any way and therefore does not have an IP address from the peering prefixes.

The management bridge *br900* has a connection to each of the containers *ns1*, *rs1*, *cs1*, *bs1* and *sc1* as well as interface *eth3* from the CDIP is connected to *br900* while interface *eth4* is connected to *br100*.

5.5.4 SDX External OF Switch

During the *ixp host build* process (Section 6, Appendix A) the external OF Ethernet switch was added to the database; however, it was not defined in detail. The *SC* exercises control of this external OF Ethernet switch via a South Bound Interface (SBI) control-channel that didn't exist in the *traditional* models. It is therefore possible to define *enabled* interfaces and the *SC* will block those remaining interfaces that can be considered as *not enabled*, i.e. *disabled*. This is something that cannot be achieved in *traditional* models without manual intervention at each external *traditional* Ethernet switch or by employing significant system administration scripts. Additionally in the *traditional* models the external *traditional* Ethernet switch is outside the control of the PoC so defining the *switch type* is arbitrary. With *software-defined* models this is not the case and it must be specified. For example, an external OvS style OF Ethernet switch does not have a specific OOB interface so the PoC is required to reserve one interface on the switch for that purpose. Other manufacturers of OF Ethernet switches, like Netgear, may have a specific physical OOB interface and therefore there is no requirement to reserve a regular interface. For a production system either tested manufacturer tables or a system to manually input OF Ethernet switch types are required (Section 7, Appendix A).

The example in Figure 66 demonstrates this. An external OvS OF Ethernet switch and a Netgear OF Ethernet switch are compared from the database perspective. In the OvS case, the first interface is reserved for the OOB function and the second interface for the connection to the data interface of the CDIP pair on the server. Therefore interfaces 3 – 48 are available for peers. In the case of the Netgear switch it has a separate physical OOB interface and therefore the first interface connects to the data interface of the CDIP pair on the server and interfaces 2 – 48 are available for peers.

OvS external switch

```
cIXP SQL>  select * from switch;
1|sw_1|ovs|Bare-metal|eth3|eth4|24|3-48|198.8.8.241/24|2a98:8:8::241/48

SQL> select * from ipv4_peer where switch_number = 1;
199.9.9.6/24|1|3|||||||peer|1|1
199.9.9.7/24|1|4|||||||peer|1|1
.. .. .. .. .. .. .. .. .. ..

199.9.9.50/24|1|47|||||||peer|1|1
199.9.9.51/24|1|48|||||||peer|1|1
```

Netgear switch

```
cIXP SQL>  select * from switch;
1|sw_1|Netgear|Bare-metal|eth3|eth4|24|2-48|198.8.8.241/24|2a98:8:8::241/48

SQL> select * from ipv4_peer where switch_number = 1;
199.9.9.6/24|1|2|||||||peer|1|1
199.9.9.7/24|1|3|||||||peer|1|1
.. .. .. .. .. .. .. .. .. ..

199.9.9.50/24|1|47|||||||peer|1|1
199.9.9.51/24|1|48|||||||peer|1|1
```

Figure 66: External switch ports on IPv6 peering LAN

5.5.5 SDX Server

In a similar way to the *traditional* models, the *ixp server build* process (Section 8, Appendix A) clones the containers *ns1*, *rs1*, *cs1* and *bs1* from a template image. Each container is configured with peering and management IPv4 and IPv6 address extracted from the database. For *software-defined* models an additional container *sc1* is added but is only configured with management LAN IPv4 and IPv6 addresses on interface *ens3* as can be observed in Figure 67. Note that no *ens4* interface exists on the *SC* container. The OvS start/stop service is restarted by *systemd* via *systemctl* and Internet connectivity from each is confirmed.

```
root@sc1:~# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
15: ens3@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP group default qlen 1000
    link/ether 00:16:3e:75:79:19 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 198.8.8.235/24 brd 198.8.8.255 scope global ens3
       valid_lft forever preferred_lft forever
    inet6 2a98:8:8::235/48 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::216:3eff:fe75:7919/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 67: IP addresses on sc1

Again the OvS configuration is quite large so it is broken down into two parts for the purpose of explanation. The OvS configuration for the containers *ns1*, *rs1*, *cs1* and *bs1* have associated bridges *ns0*, *rs0*, *cs0* and *bs0* built in the OvS in identical fashion to the *traditional* models so they require no further explanation.

The first part of the OvS configuration, as demonstrated in Figure 68, develops the configuration of *br100*, the peering LAN. This bridge includes an internal port, patch ports to connect to the the bridges *ns0*, *rs0*, *cs0* and *bs0*, a port associated with physical interface *eth4* which forms part of the CDIP to the external OF Ethernet switch and the remaining physical interfaces 5 – 8 to connect to peers on the server. Additionally *br100* has a controller statement that defines the control-channel over the management LAN and a fail mode should the controller link fail. In such a case the OvS resorts to *traditional* switching.

```
    Bridge "br100"
        Controller "tcp:198.8.8.235:6633"
        fail_mode: standalone
        Port "br100"
            Interface "br100"
                type: internal
        Port "ns0-1"
            Interface "ns0-1"
                type: patch
                options: {peer="ns0-2"}
        Port "rs0-1"
            Interface "rs0-1"
                type: patch
                options: {peer="rs0-2"}
        Port "cs0-1"
            Interface "cs0-1"
                type: patch
                options: {peer="cs0-2"}
        Port "bs0-1"
            Interface "bs0-1"
                type: patch
                options: {peer="bs0-2"}
        Port "eth4"
            Interface "eth4"
        Port "eth5"
            Interface "eth5"
        Port "eth6"
            Interface "eth6"
        Port "eth7"
            Interface "eth7"
        Port "eth8"
            Interface "eth8"
```

Figure 68: Software-defined OvS configuration - Part 1

The second part of the OvS configuration is the management bridge *br900* as shown in Figure 69. Like its *traditional* switching counterpart it has a default internal port as well as two configured internal ports, *ip900-4*, *ip900-6* for local server IP addresses. Additional to the containers *ns1*, *rs1*, *cs1* and *bs1*, the new container *sc1* also has a dynamic *veth* port *vethVPEXSG* as demonstrated in Figure 70. The first interface in the CDIP, *eth3*, has a port in the bridge *br900* as does the physical interface *eth2*.

5.5.5.1 OF port numbers

Typically the OvS picks arbitrary OF port numbers to match ports within its switches. This is not useful for programming purposes so the PoC assigns specific OF numbers according to the rules outlined in Table 10.

```
    Bridge "br900"
        Port "br900"
            Interface "br900"
                type: internal
        Port "ip900-4"
            Interface "ip900-4"
                type: internal
        Port "ip900-6"
            Interface "ip900-6"
                type: internal
        Port "vethPN3330"
            Interface "vethPN3330"
        Port "veth8Y8HJ2"
            Interface "veth8Y8HJ2"
        Port "vethAD4205"
            Interface "vethAD4205"
        Port vethVPEXSG
            Interface vethVPEXSG
        Port "vethDALF79"
            Interface "vethDALF79"
        Port "eth2"
            Interface "eth2"
        Port "eth3"
            Interface "eth3"
```

Figure 69: Software-defined OvS configuration - Part 2

```
root@sc1:~# ip link | grep ens3 | awk '{print $2}'| awk -F '@' '{print $2}'
if150:

ubuntu@mIXP:~$ sudo ovs-vsctl find interface ifindex=150 | grep '^name'
name                : vethVPEXSG
```

Figure 70: Map SC interface index to veth port

Table 10: OF determined port numbers

OF port number	Function
10yy	Peering – Peer, i.e. 1003 = eth3 as a peer interface
15yy	Peering – External OvS peer, i.e. 1502 = eth2 as an external peer
20xx	Peering – Patch 1
30xx	Peering – Patch 2
41yy	Peering – physical CDIP, 4106 = eth6 in CIDP pair
70yy	Management – physical, i.e. 7002 = eth2 as man interface
71yy	Management – physical CDIP, 4105 = eth5 in CIDP pair
8001	Management – internal IPv4
8002	Management – internal IPv6

yy – physical port number, i.e eth4 = 04, xx – arbitrary number picked by OvS

5.5.6 SDX Software

The *ixp software* processes (Section 9, Appendix A) installs the software packages and then configures the installed software on the containers for *NS*, *RS*, *CS* and *BS* as described for the *traditional* models. The processes also installs and configures the following main packages and their associated dependencies on the additional *SC* container;

- ***SC*** : python3-ryu, flowmanager.

The *python3-ryu* package is an SDN framework which provides software components with a well defined API to allow for the creation of new network management and control applications. Ryu fully supports OF v1.0, v1.2, v1.3, v1.4 and Nicira Extensions. For the purpose of the PoC, OF v1.3 is employed as there are very few implementations of OF versions beyond v1.3 in switching hardware today.

The second package *flowmanager* was used in earlier stages of the work to interact with the Ryu flow tables; however, it is not in general use within the operation of the PoC. It remains included as part of the install as a useful tool for troubleshooting.

Having installed the packages, the PoC *ixp_switch_13.py* sub-class of the Ryu `ryu.base.app_manager` class is installed. This provides contexts as well as routing of messages between Ryu applications like the `ryu.app.ofctl_rest` RESTful API with which the `ixp_switch_13` is ran. The final installation step is the creation of the Ryu service that is used by *systemd* to manage starting and stopping the *SC*.

5.5.7 SDN Controller

5.5.7.1 Initialisation

During *SC* initialisation the Ryu main class loads the following;

- the `ixp_switch_13` sub-class from the *ixp_switch_13.py* file,
- the Ryu OF protocol handler, `ryu.controller.ofp_handler`,
- the Ryu RESTful API, `ryu.app.ofctl_rest`.

Each of these elements are instantiated. During the instantiation of the `ixp_switch_13` sub-class the *Datapath Identifier (DPID) to Port map* file that supplies a dictionary of DPID keys, each with a list of ports that need to be blocked, is read. This map may not exist at an initial start-up event but will be generated shortly thereafter. If the map exists it is read by the *SC* for future use. The example in Figure 71 shows the ports 1005 – 1008 should be blocked for DPID, *69131984840* which is the internal OvS switch and ports 3 – 8 on the external OF Ethernet switch (which in this case is the example external OvS switch).

```
root@sc1:~# cat /srv/ryu/.dpid_to_port.map; echo
{"69131984840": [1005, 1006, 1007, 1008], "69133577248": [3, 4, 5, 6, 7, 8]}
```

Figure 71: Datapath to port map

5.5.7.2 OF Switch registration

Shortly after initialisation, the OF switches send an OF event *OFPSwitchFeatures* to the *SC*, this event triggers the `_switch_features_handler()` method which (1) opens the *DPID to IP map* and adds the DPID key with the IP address of the device the registration message came from. This is demonstrated in Figure 72 where it is clear that DPID, *69131984840* refers to the internal OvS switch and DPID, *69133577248* refers to the external switch by their respective IP addresses. These entries were generated at the point when the OF switches reported initially to the *SC*.

```
root@sc1:~# cat /srv/ryu/.dpid_to_ip.map; echo
{"69133577248": "198.8.8.241", "69131984840": "198.8.8.230"}


cIXP:~$ ip address show dev ip900-4 | grep 'inet '
    inet 198.8.8.230/24 scope global ip900-4


CIXP SQL> select * from switch;
1|sw_1|ovs|Bare-metal|eno3|eno4|8|3-8|198.8.8.241/24|2a98:8:8::241/48


OvS:~$ ip address show dev eth1 | grep 'inet '
    inet 198.8.8.241/24 brd 198.8.8.255 scope global eth1
```

Figure 72: Datapath to IP map

(2) It then clears any existing flows that exist in the switch (see `_clear_flows()` method below) and (3) installs a *table-miss* flow entry. This instructs the OF switch to send frames that have no other match to the *SC* via an *OFPPacketIn* message for processing as illustrated in Figure 73.

- ***DPID*** : Datapath Identifier
- ***Match*** : OFPMatch(), empty set {}, match all frames
- ***Priority*** : 0, only matches frames not already matched
- ***Actions*** : OFPP_CONTROLLER, send packet to the *SC*
  OFPCML_NO_BUFFER, entire packet sent to the *SC*

Figure 73: Table-miss flow entry

(4) Finally it blocks any ports noted in the *DPID to port map* with a drop frames action message as can be seen in Figure 74.

- ***DPID*** : Datapath Identifier
- ***Match*** : in port from the *DPID to port map*
- ***Priority*** : 99, higher than table-miss [0] or injected flows [1]
- ***Actions*** : Empty set {}, drop frames

Figure 74: Drop frames flow entry

These three steps can be observed in the logs and flow outputs in Figure 75. The first log is the Ryu `ixp_switch_13` sub-class log generated as the flows were added to the two OF switches. The next two logs show the internal OvS and the external OF Ethernet switch flow dumps. In the latter two cases the flows are ordered by priority. The initial *clear flow* modification is not shown as these instructed the OF switches to actually clear the flows from their flow tables so there were none to log. The priority 99 flows are the blocking flows generated from the entries in the *DPID to port map* and the lowest priority is the *table-miss* entry which sends frames that are not currently matched to the *SC* for processing, a default flow. At this stage the only frames being processed are between the *NS*, *RS*, *CS* and *BS* containers.

```
root@sc1:~# tail -f /var/log/ryu/ryu.log
CLEAR FLOW: 69131984840 (198.8.8.230)
ADD FLOW: 69131984840 (198.8.8.230) | Table miss
ADD FLOW: 69131984840 (OFPMatch(oxm_fields={'in_port': 1005})) | A:
ADD FLOW: 69131984840 (OFPMatch(oxm_fields={'in_port': 1006})) | A:
ADD FLOW: 69131984840 (OFPMatch(oxm_fields={'in_port': 1007})) | A:
ADD FLOW: 69131984840 (OFPMatch(oxm_fields={'in_port': 1008})) | A:
CLEAR FLOW: 69133577248 (198.8.8.241)
ADD FLOW: 69133577248 (198.8.8.241) | Table miss
ADD FLOW: 69133577248 (OFPMatch(oxm_fields={'in_port': 3})) | A:
ADD FLOW: 69133577248 (OFPMatch(oxm_fields={'in_port': 4})) | A:
ADD FLOW: 69133577248 (OFPMatch(oxm_fields={'in_port': 5})) | A:
ADD FLOW: 69133577248 (OFPMatch(oxm_fields={'in_port': 6})) | A:
ADD FLOW: 69133577248 (OFPMatch(oxm_fields={'in_port': 7})) | A:
ADD FLOW: 69133577248 (OFPMatch(oxm_fields={'in_port': 8})) | A:


cIXP:~$ sudo ovs-ofctl dump-flows br100 | awk '{print $7 " --> " $8}' | grep -E
'(priority=0|priority=99)'
priority=99,in_port=1005 --> actions=drop
priority=99,in_port=1006 --> actions=drop
priority=99,in_port=1007 --> actions=drop
priority=99,in_port=1008 --> actions=drop
priority=0 --> actions=CONTROLLER:65535


OvS:~$ sudo ovs-ofctl dump-flows br0 | awk '{print $7 " --> " $8}' | grep -E
'(priority=0|priority=99)'
priority=99,in_port=3 --> actions=drop
priority=99,in_port=4 --> actions=drop
priority=99,in_port=5 --> actions=drop
priority=99,in_port=6 --> actions=drop
priority=99,in_port=7 --> actions=drop
priority=99,in_port=8 --> actions=drop
priority=0 --> actions=CONTROLLER:65535
```

Figure 75: Clear flows, install a table miss flow entry & block ports

5.5.7.3 Packet-in handling

After initialisation and switch registration the *SC* waits for an OF event *OFPPacketIn* which triggers the `_packet_in_handler()` method. This method performs the following actions. (1) It drops any DECnet Maintenance Operation Protocol (MOP) messages based on the destination Medium Access Control (MAC) multicast address *ab:00:00:02:00:00*. This is a rarely used data link layer remote management protocol from the DECnet protocol suite, it is recommended to drop such frames as they are typically not required on modern networks. (2) It drop any frames that have the same source and destination MAC address as such frames are considered illegal. (3) The source MAC address of the remaining frames are added to the *MAC to port map* table. (4) It checks the destination MAC address against the *MAC to port map* table and if there is an entry, the *out port* is noted as the *action*. (5) It sends an *OFPPacketOut* back to the OF switch with the *DPID*, *message buffer ID* and the *out port* as the *action* and (6) adds a flow to the switch to match subsequent frames with the same *in port*, *source MAC* and *destination MAC* address with the *out port* as the

action (see `_add_flow()` method below). (7) If there is no matching *out port* in the *MAC to port map* table then the *OFPP_FLOOD* flag is set. This indicates that the packet should be sent out on all ports and an *OFPPacketOut* message is returned to the OF switch with the *DPID*, *message buffer ID* and *out all ports* as the *action*.

5.5.7.4 Add Flow

The `ixp_switch_13` sub-class handles the addition of flows to OF switches with the `_add_flow()` method which sends a modify flow entry message, *OFPFlowMod* to the OF switch as demonstrated in Figure 76.

- ***DPID*** : Datapath Identifier
- ***Match*** : set of match conditions {*in port*, *source MAC*, *destination MAC*}
- ***Command*** : *OFPFC_ADD* message instructing the OF switch to add the flow
- ***Priority*** : Priority level of flow entry, the higher number, higher priority
- ***Buffer ID*** : Default is *None*, or the ID of the switch buffer holding the frame
- ***Out port*** : Action port to forward matching frames
- ***Actions*** : *OFPIT_APPLY_ACTIONS*, applies the action immediately

Figure 76: Add flow OFPFlowMod message

5.5.7.5 Clear Flows

The final method in the `ixp_switch_13` sub-class is the `_clear_flows()` method which is called when the `_switch_features_handler()` method is triggered initially. This method sends an OF flow modification message *OFPFlowMod* to the switch as per Figure 77.

- ***DPID*** : Datapath Identifier
- ***Match*** : Empty set {}, so all are matched
- ***Command*** : *OFPFC_DELETE*, delete all matching flows
- ***Priority*** : Set to 32768
- ***Out port*** : *OFPP_ANY*, matching all ports
- ***Out group*** : *OFPG_ANY*, matching all groups
- ***Actions*** : Empty set {}

Figure 77: Clear flow OFPFlowMod message

5.5.7.6 RESTful API

The other sub-class of the `ryu.base.app_manager` class is the `ryu.app.ofctl_rest` sub-class that provides a RESTful API access to the *SC* (Section 11, Appendix A). Like the *Birdseye* API on the containers with the BIRD daemon, this API also returns data in JSON format and can be accessed using an Internet browser that supports the display of JSON. This API also exposes a powerful North Bound Interface (NBI) for the development of additional tools for automation and monitoring.

As an example consider the simple python3 program snippet in Figure 78. The code actually makes two GET requests to the RESTful API. The first GET request retrieves a list of the switch DPIDs by passing `/stats/switches/` as part of the URL while the second request takes each of these DPIDs in turn and passes `/stats/desc/<dpid>/` to retrieve each OF switch description in turn which are then displayed via STDOUT at the terminal.

Consider the output of direct access from the shell command *cURL* and the output of the program snippet in Figure 79. Like the *Birdseye* example, the program reads and processes the input using the python JSON module method `loads()`. The strings are parsed into a list and a dictionary so they can be easily manipulated within python as demonstrated by this example.

There are many commands that can be extracted from the Ryu RESTful API in this manner. The Ryu RESTful API has also mechanisms for modifying flows. A sample set of these are outlined in Table 11.

```
cIXP:~$ cat ryu_rest_api.py
#! /usr/bin/env python3
# -*- coding: utf-8 -*-

import urllib.request
import json

list_ = list()
dict_ = dict()
tab = 10
host = 'sc1.netlabs.tst'
http_port = '8080'
url_list = ['/stats/switches',
            '/stats/desc/']

def get_rest(u):
    ''' Grab data from the RESTful API '''
    f = urllib.request.urlopen(u).read()
    g = f.decode('utf-8')
    print(f'\n{g}')
    h = json.loads(g)
    return(h)

def switch_layout(d):
    ''' Layout of the switch description '''
    print()
    dp = list(d.keys())[0]
    l = len(str(dp)) + tab
    print(f"Datapath: {dp}\n{'-' * l}")
    for k,v in d[dp].items():
        print(f"{k.replace('_', ' '):<{tab}} :{v}")
    return(0)

url_ = f'http://{host}:{http_port}{url_list[0]}'
list_ = get_rest(url_)

for x in list_:
    url_ = f'http://{host}:{http_port}{url_list[1]}{x}'
    dict_ = get_rest(url_)
    switch_layout(dict_)
print()
```

Figure 78: Ryu RESTful API switch description python snippet

```
cIXP:~$ curl sc1.netlabs.tst:8080/stats/switches;echo
[69133577248, 69131984840]

cIXP:~$ curl sc1.netlabs.tst:8080/stats/desc/69131984840;echo
{"69131984840": {"mfr_desc": "Nicira, Inc.", "hw_desc": "Open vSwitch",
"sw_desc": "2.9.2", "serial_num": "None", "dp_desc": "None"}}

cIXP:~$ curl sc1.netlabs.tst:8080/stats/desc/69133577248;echo
{"69133577248": {"mfr_desc": "Nicira, Inc.", "hw_desc": "Open vSwitch",
"sw_desc": "2.9.2", "serial_num": "None", "dp_desc": "None"}}

cIXP:~$ ./ryu_rest_api.py

[69133577248, 69131984840]

{"69133577248": {"mfr_desc": "Nicira, Inc.", "hw_desc": "Open vSwitch",
"sw_desc": "2.9.2", "serial_num": "None", "dp_desc": "None"}}

Datapath: 69133577248
---------------------
mfr desc   :Nicira, Inc.
hw desc    :Open vSwitch
sw desc    :2.9.2
serial num :None
dp desc    :None

{"69131984840": {"mfr_desc": "Nicira, Inc.", "hw_desc": "Open vSwitch",
"sw_desc": "2.9.2", "serial_num": "None", "dp_desc": "None"}}

Datapath: 69131984840
---------------------
mfr desc   :Nicira, Inc.
hw desc    :Open vSwitch
sw desc    :2.9.2
serial num :None
dp desc    :None
```

Figure 79: Compare cURL and the simple Ryu python snippet output

Table 11: Ryu RESTful API URL examples

RESTful API URL	Function
Retrieve information	
/stats/switches	Get list of switch DPIDs
/stats/desc/<dpid>	Get a description of the switch based on DPID
/stats/portdesc/<dpid>	Get port description as a dictionary
/stats/port/<dpid>	Get dictionary of port statistics, this includes port number
/stats/port/<dpid>/<port>	Get dictionary of statistics on a specific port
/stats/flow/<dpid>	Get dictionary of switch OF flows
/stats/aggregateflow/<dpid>	Get dictionary of aggregate switch OF flow statistics
Modify switch flows	
/stats/flowentry/add	Add a flow
/stats/flowentry/modify	Modify flows
/stats/flowentry/clear/<dpid>	Delete all flows
/stats/flowentry/delete	Delete matching flow entries

Add to http://<host URL>:8080 i.e. http://sc1.netlabs.tst:8080/stats/switches

5.6 Handling peers

The PoC facilitates the addition of new peers to the IXP (Section 11, Appendix A). When a new peer is added the IXP administrator presents the following mandatory information;

- ***Peer name*** : Arbitrary name for the peering member, i.e. Ripplecom.
- ***Peer ASN*** : ASN assigned by Regional Internet Registry (RIR), i.e. AfriNIC.
- ***Peer domain*** : Member domain-name, i.e. ripplecom.net.

The IXP administrator can also allow a peering member to opt out of peering with either or both the *RS* and *BS*. This is in line with the peering inclination of some IXP members. However, members cannot opt out of peering with the *CS* as its data is only gathered for statistical purposes.

- ***Route service*** : Yes/No, default is yes,
- ***AS112 service*** : Yes/No, default is yes.

When this data is given the PoC checks the three pieces of information to ensure that they have not been submitted as part of an earlier entry and that they are of a valid format. Once this step is complete the new peering member information is added to the database. The new peer is added to the first free interface matching the required interface speed, in this example interface 5 on switch 0, the internal OvS OF switch *br100*. The peer is assigned the IP addresses associated with this interface, 199.9.9.2/24 and 2a99:9:9::2/48 and the database table *enabled* flag is set to 2.

Figure 80 shows five peers added, the first four taking the peering interfaces on the server associated with switch 0, the internal OvS OF switch, *br100*. The fifth peer is allocated to the first free interface on the external OF Ethernet switch. *ISP3* has been configured to peer with the *CS* and *RS* only, *ISP4* with the *CS* and *BS* only and *ISP5* with the *CS* only to facilitate the member's peering policy.

If the switching type is *software-defined* then the blocks placed on ports 1005 – 1008 of the internal OvS OF switch are removed and the block on the first interface of the external OvS OF Ethernet switch is also removed as the peers are added. The blocks placed on the remaining ports remains in place until they are configured with their own peers.

After the peers are added to the database the PoC synchronises the configuration of the BIRD daemons. The python code snippet in Figure 81 demonstrates this. It interrogates the BIRD daemons via the *Birdseye* RESTful API. Interrogating the *CS* results in all five peers showing to have ESTABLISHED BGP peering connections. Considering *RS* and *BS*, only peers ISP1, ISP2 and ISP4 have chosen to peer with the *RS*, while only ISP1, ISP2 and ISP3 have chosen to peer with the *BS*. This is reflected in the outputs of the snippet of code demonstrated in Figure 82.

```
cIXP SQL> select * from ipv6_peer where function = 'peer' and enabled = 2;
2a99:9:9::2/48|0|5|ISP1|ISP1|5111|one.com|1G|yes|yes|peer|2|1
2a99:9:9::3/48|0|6|ISP2|ISP2|5222|two.com|1G|yes|yes|peer|2|1
2a99:9:9::4/48|0|7|ISP3|ISP3|5333|three.com|1G|no|yes|peer|2|1
2a99:9:9::5/48|0|8|ISP4|ISP4|5444|four.com|1G|yes|no|peer|2|1
2a99:9:9::6/48|1|3|ISP5|ISP5|5555|five.com|10G|no|no|peer|2|1

 cIXP:~$ sudo ovs-ofctl dump-flows br100 | awk '{print $7 " --> " $8}' |
grep -E 'priority=99'
 [ Note the null response ]

 OvS:~$ sudo ovs-ofctl dump-flows br0 | awk '{print $7 " --> " $8}' | grep -
E 'priority=99'
 priority=99,in_port=4 --> actions=drop
 priority=99,in_port=5 --> actions=drop
 priority=99,in_port=6 --> actions=drop
 priority=99,in_port=7 --> actions=drop
 priority=99,in_port=8 --> actions=drop
```

Figure 80: Peer added to the database

```
cIXP:~$ cat birdseye_rest_api.py
#! /usr/bin/env python3
# -*- coding: utf-8 -*-

import urllib.request
import json
import sys

host = str()
domain = 'netlabs.tst'
host = str()
url_ = '/api/protocols/bgp'
url2_ = '/api/protocol'
ver = list()
dict_ = dict()
tab = 18
list_ = ['protocol', 'state', 'connection',
         'neighbor_address', 'neighbor_as',
         'neighbor_id', 'source_address']

def get_rest(h, u, p = ''):
    if (p == ''):
        url = f'http://{h}{u}'
    else:
        url = f'http://{h}{u}/{p}'
    f = urllib.request.urlopen(url).read()
    s = f.decode('utf-8')
    d = json.loads(s)
    return(d)

def protocol_layout(d):
    d = d['protocol']
    for k in list_:
        if (k in d.keys()):
            v = d[k]
            k = k.replace('_', ' ').title()
            print(f'{k:<{tab}}: {str(v).strip()}')
    return(0)

if (len(sys.argv) > 1):
    host = sys.argv[1]
else:
    print(f'\nERROR: $ {sys.argv[0]} <srv>-<ipv4|ipv6>.<domain>\n')
    sys.exit(1)

protocols = list(get_rest(host, url_)['protocols'].keys())
print()
for x in protocols:
    dict_ = get_rest(host, url2_, x)
    protocol_layout(dict_)
    print()
```

Figure 81: Birdseye RESTful API BGP protocol python snippet

```
cIXP:~$ ./birdseye_rest_api.py rs1-ipv4.netlabs.tst

Protocol          : ISP1
State             : up
Connection        : Established
Neighbor Address  : 199.9.9.2
Neighbor As       : 5111
Neighbor Id       : 200.1.1.1
Source Address    : 199.9.9.1

Protocol          : ISP2
State             : up
Connection        : Established
Neighbor Address  : 199.9.9.3
Neighbor As       : 5222
Neighbor Id       : 200.2.2.2
Source Address    : 199.9.9.1

Protocol          : ISP4
State             : up
Connection        : Established
Neighbor Address  : 199.9.9.5
Neighbor As       : 5444
Neighbor Id       : 200.4.4.4
Source Address    : 199.9.9.1

cIXP:~$ ./birdseye_rest_api.py bs1-ipv6.netlabs.tst

Protocol          : ISP1
State             : up
Connection        : Established
Neighbor Address  : 2a99:9:9::2
Neighbor As       : 5111
Neighbor Id       : 200.1.1.1
Source Address    : 2a99:9:9::234

Protocol          : ISP2
State             : up
Connection        : Established
Neighbor Address  : 2a99:9:9::3
Neighbor As       : 5222
Neighbor Id       : 200.2.2.2
Source Address    : 2a99:9:9::234

Protocol          : ISP3
State             : up
Connection        : Established
Neighbor Address  : 2a99:9:9::4
Neighbor As       : 5333
Neighbor Id       : 200.3.3.3
Source Address    : 2a99:9:9::234
```

Figure 82: BGP status at the RS and BS

5.7 Remote mini IXP

mIXPs are IXPs at remote locations whose day-to-day management is handled from the cIXP. A set of IXPs including one cIXP and one or more mIXPs is termed a dIXP. This can be visualised in Figure 30 on page 100. The build and configuration of each mIXP follows the same path as the cIXP except that the schema *site* table identifies the mIXP by setting the attribute site type: *mini* as demonstrated in Figure 83. The mIXP details are recorded in the *remote* table of the cIXP database. The routing between the two sites is an issue for the IXP management team. Management can decide to connect the two management LANs over the Internet or they may choose to use the services of IXP members to provide Point-to-Point services between cIXP and the mIXPs. The level of traffic is extremely small, simply facilitating the execution of management instructions and the traffic is secured using Rivest, Shamir and Adleman (RSA) public and private key pair comprising of two uniquely related cryptographic keys. As such passing this traffic over the Internet is acceptable.

Record in the mIXP database indicating that it is a mIXP.

```
mIXP SQL> select * from site;
2|mini|software-defined|Uganda|Gulu|1100|netLabs!UG|02 46 40.2314 N|32 17 14.1243 E
```

Record in the cIXP about the mIXP.

```
cIXP SQL> select * from remote;
2|Gulu|Gulu|176.6.6.230|2a76:6:6::230|ubuntu|22
```

Connectivity test between cIXP and mIXP.

```
cIXP:~$ fping 176.6.6.230
176.6.6.230 is alive

cIXP:~$ fping 2a76:6:6::230
2a76:6:6::230 is alive

cIXP:~$ fping 198.8.8.230
198.8.8.230 is alive

cIXP:~$ fping 2a98:8:8::230
2a98:8:8::230 is alive
```

Figure 83: Remote mIXP recorded in the databases and connectivity

The cIXP generates a public/private key pair. The private key is retained by the cIXP but the public key is shared verbatim with the mIXPs as demonstrated in Figure 84 (Section 14, Appendix A). Using these keys with SSH the cIXP can execute commands securely on the remote IXP. The command set is limited as the mIXP is built before its installation at the remote site so only commands to view the remote schema, host, external switches, SDN switches and flows as well as the container functions are necessary. Commands to build, install and configure are not included, except for IXP peer commands where the complete suite is made available to allow the IXP administrator to add, delete and view peers, as well as list and view their status.

Example of public key on cIXP

```
cIXP:~$ cat /srv/keys/public.key; echo
 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC56jJdPht9rhKa/BAdgjw4XkYCllz6Y0oquQ219a7lFpT
piwoKhkv9Mpd2gYBffWonfR8ikfvd/uu0soERN/o7f5EJ0PwhVQGRs0MVpOoxD/
hL+sShNbRXno3XZToATUDqJJmpHGrlbYrFLyMmkLYUus7pqgh5CBp1O80QmI63tdo9kOUhkWb+F
CluKkcr5RsneyDg0vPAtvO5QOgEI0gbXwQJtyQQpgIUxYuvvh5B+813S6vsN5MLIgOMlxVxZeDZ
Ucl+rEDceSp22R6FuZEx2gt+92eGyc1KDJDynhxPWeriJadkVdzqcPX59PXZqn/
Z5CI58Pvek7rTQPhjFZzl
```

cIXP public key copied verbatim to each mIXP

```
mIXP:~$ cat /srv/keys/public.key; echo
 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC56jJdPht9rhKa/BAdgjw4XkYCllz6Y0oquQ219a7lFpT
piwoKhkv9Mpd2gYBffWonfR8ikfvd/uu0soERN/o7f5EJ0PwhVQGRs0MVpOoxD/
hL+sShNbRXno3XZToATUDqJJmpHGrlbYrFLyMmkLYUus7pqgh5CBp1O80QmI63tdo9kOUhkWb+F
CluKkcr5RsneyDg0vPAtvO5QOgEI0gbXwQJtyQQpgIUxYuvvh5B+813S6vsN5MLIgOMlxVxZeDZ
Ucl+rEDceSp22R6FuZEx2gt+92eGyc1KDJDynhxPWeriJadkVdzqcPX59PXZqn/
Z5CI58Pvek7rTQPhjFZzl
```

Figure 84: RSA Public keys matching on cIXP and mIXP

5.7.1 Summary of IXP and SDX models

A summary of all 11 models and sub-models developed is outlined in Table 12. The selection criteria for any particular model is dependent on three factors;

- ***Number of interfaces***: The number of available interfaces on the server hardware maybe a determining factor as to the model chosen.
- ***Switching type***: For interfaces greater than three, the switching type must be chosen, either *traditional* or *software-defined*.
- ***Number of external switches***: The number of external switches as well as the switching type

determines the mix of trunk to peering interfaces. The models are deployable in different scenarios depending on the scale required. Using easily purchased *traditional* managed Ethernet switches further enhances the ease of implementation and affordability of these systems.

Table 12: Summary of IXP models

Model	Switch	Interfaces	Interfaces				
			1	2	3	4	5+
A	Traditional	1	Trunk				
B	Traditional	2	OOB	Trunk			
C	Traditional	> 3	OOB	Management	Peers		
D_1	Traditional	> 3	OOB	Management	Trunk	Peers	
D_2	Traditional	> 3	OOB	Management	Mix of Peer and Trunks		
E_1	Traditional	3	OOB	Management	Trunk		
E_2	Traditional	> 3	OOB	Management	Trunks		
S	OF	> 3	OOB	Management	Peers		
T_1	OF	> 3	OOB	Management	OF 1 CDIP		Peers
T_2	OF	> 3	OOB	Management	Mix of CDIPs and Peers		
U	OF	> 3	OOB	Management	CDIPs		

It is advised that a server with a minimum of four Ethernet interfaces is employed in all cases, thereby eliminating the need for models A, B as well as sub-model E_1. For larger sites, models D or T are recommended depending on the switching type selection for *traditional* Ethernet or OF Ethernet switches. For smaller sites, particularly mIXPs, the use of a server that has enough interfaces for all members plus two additional interfaces for OOB and on-site management is very cost effective as no Ethernet switches are required. For example a site with six members can be catered for using eight interfaces which is a typical COTS server configuration. In this case, models C or S can be employed; however, model S is recommended as the *software-defined* model enables the SDX mode and facilitates additional functionality.

5.8 PoC Demonstration

The PoC software has been built under the name *IXPBuilder*. Having described how the PoC works this section demonstrates how it operates (Appendix A). This takes the system from installation of IXPBuilder on the Ubuntu 10.04 OS, the construction of the IXP schema, the install of LXCs for functions and the installation and configuration of the required software on each

functional container in line with the IXP schema. A number of test peers are added to the IXPs and the operation and results from the various tests are demonstrated.

5.8.1 Single Site IXPBuilder functionality test

The commands in this section are executed on one server to build a basic operational IXP testbed representing a single site with three peers. Installation takes about one hour depending on the choices made. The proximity of the Ubuntu repository as well as the quality of the Internet connection are factors which impact on build time. There are two test within this, a functionality test for *traditional* switching and a test for *software-defined* switching.

5.8.1.1 Install Ubuntu 18.04 LTS and IXPBuilder

Carry out a basic Ubuntu 18.04 LTS server install with OpenSSH as the only additional software and reboot the server (Section 2, Appendix A). The approximate installation time (time of reboot is hardware dependent) is 21 minutes. When the OS reboots install the IXPBuilder software on each server as show in Figure 85 (Section 3, Appendix A). The approximate installation time for this step is 17 minutes.

```
cIXP:~$ tar -xzvf ~/IXPBuilder_v5.0.tgz
cIXP:~$ ls ~
IXPBuilder_v5.0.tgz  ixp

cIXP:~$ cd ~/ixp/tools
cIXP:~/ixp/tools$ sudo ./ixp-install.sh
```

Figure 85: Install the IXPBuilder PoC software

5.8.1.2 Models using traditional Ethernet switches

Building the IXP testbed to support *traditional* Ethernet switches is demonstrated in Figure 86 (Sections 5-9 and 12, Appendix A). For the purpose of the example, model D has been applied. The approximate configuration time for this step is 15 minutes.

```
cIXP:~$ ixp schema default
cIXP:~$ ixp host build
  Configuring the IXP host for model: d
cIXP:~$ ixp switch set –s 1 –si 48
cIXP:~$ ixp server build –y all
cIXP:~$ ixp software install all
cIXP:~$ ixp software configure all
cIXP:~$ ixp peer add –n C1 –a 5111 –d one.com
cIXP:~$ ixp peer add –n C2 –a 5222 –d two.com –rs no
cIXP:~$ ixp peer add –n C3 –a 5333 –d three.com –bs no
```

Figure 86: Build an IXP for traditional Ethernet switches

5.8.1.3 Software-defined Switching models

Building an SDX testbed for *software-defined* switching is demonstrated in Figure 87 (Sections 5 – 9 and 12, Appendix A). The approximate configuration time for this step is 21 minutes.

```
cIXP:~$ ixp schema build –sw soft
cIXP:~$ ixp host build –s 1
cIXP:~$ ixp switch set –s 1 –si 48 –ei all –st ovs
cIXP:~$ ixp server build –y all
cIXP:~$ ixp software install all
cIXP:~$ ixp software configure all
cIXP:~$ ixp peer add –n C1 –a 5111 –d one.com
cIXP:~$ ixp peer add –n C2 –a 5222 –d two.com –rs no
cIXP:~$ ixp peer add –n C3 –a 5333 –d three.com –bs no
```

Figure 87: Build a software-defined switching IXP

A basic test of the configuration on the cIXP involves listing the peers to ensure they were installed correctly as demonstrated in Figure 88. A further check of the peer status as shown in Figure 89 verifies that the peers have established BGP sessions with ISP routers in the *peer tier* (Section 12, Appendix A).

If this is a completely fresh installation, the *ixp server build* process (Section 8, Appendix A) must download a copy of the Ubuntu 18.04 cloud template first. This takes significant time and subsequent installations are faster.

```
cIXP:~$ ixp peer list
  + ------------------------------------------------------------------------------ +
  |                                IPv4 Peering Table                              |
  + ------ + ---- + ------ + ----- + ----- + --------- + --- + ----- + ------------ +
  |  Name  |  ASN | Switch |  Port | Speed | Domain    | RS  | AS112 |  IP Address  |
  + ------ + ---- + ------ + ----- + ----- + --------- + --- + ----- + ------------ +
  |   C1   | 5111 |   0    |   3   |  1G   |  one.com  | yes |  yes  | 199.9.9.2/24 |
  |   C2   | 5222 |   0    |   4   |  1G   |  two.com  | no  |  yes  | 199.9.9.3/24 |
  |   C3   | 5333 |   0    |   5   |  1G   | three.com | yes |  no   | 199.9.9.4/24 |
  + ------ + ---- + ------ + ----- + ----- + --------- + --- + ----- + ------------ +

  + ------------------------------------------------------------------------------ +
  |                                IPv6 Peering Table                              |
  + ------ + ---- + ------ + ----- + ----- + --------- + --- + ----- + ------------ +
  |  Name  |  ASN | Switch |  Port | Speed | Domain    | RS  | AS112 |  IP Address  |
  + ------ + ---- + ------ + ----- + ----- + --------- + --- + ----- + ------------ +
  |   C1   | 5111 |   0    |   3   |  1G   |  one.com  | yes |  yes  | 2a99:9:9::2/48 |
  |   C2   | 5222 |   0    |   4   |  1G   |  two.com  | no  |  yes  | 2a99:9:9::2/48 |
  |   C3   | 5333 |   0    |   5   |  1G   | three.com | yes |  no   | 2a99:9:9::2/48 |
  + ------ + ---- + ------ + ----- + ----- + --------- + --- + ----- + ------------ +
```

Figure 88: IXP peer list

```
cIXP:~$ ixp peer status
  + ---------------------------------------- +
  |            IPv4 BGP State Table          |
  + -------- + ------ + ------- + ------------ +
  |  Server  |  Name  |  State  |     Info     |
  + -------- + ------ + ------- + ------------ +
  |   rs1    | C1     | up      | Established  |
  |          | C2     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
  |   cs1    | C1     | up      | Established  |
  |          | C2     | up      | Established  |
  |          | C3     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
  |   bs1    | C1     | up      | Established  |
  |          | C2     | up      | Established  |
  + -------- + ------ + ------- + ------------ +

  + ---------------------------------------- +
  |            IPv6 BGP State Table          |
  + -------- + ------ + ------- + ------------ +
  |  Server  |  Name  |  State  |     Info     |
  + -------- + ------ + ------- + ------------ +
  |   rs1    | C1     | up      | Established  |
  |          | C3     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
  |   cs1    | C1     | up      | Established  |
  |          | C2     | up      | Established  |
  |          | C3     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
  |   bs1    | C1     | up      | Established  |
  |          | C2     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
```

Figure 89: IXP peer status

5.8.2 Basic multi-site IXPBuilder functionality test

The commands in this section are executed on two servers, one to build a basic operational cIXP and the other to build an mIXP, together forming a dIXP testbed representing two sites. Installation takes about 1 hour depending on the choices made and assuming work on both installs are carried out in parallel. The proximity of the Ubuntu repository as well as the quality of the Internet connection are also factors on the time. This test focuses on a *software-defined switching* model at both sites; however, it is worth noting that a multi-site configuration can employ *traditional*, *software-defined* or even mixed switching types between IXPs within the dIXP set.

5.8.2.1 Install Ubuntu 18.04 LTS and IXPBuilder

Carry out a basic Ubuntu 18.04 LTS server install with OpenSSH as the only additional software on both servers simultaneously. Reboot both servers and install the IXPBuilder software on each server as shown in Figure 90 (Section2, Appendix A).

```
cIXP
cIXP:~$ tar -xzvf ~/IXPBuilder_v5.0.tgz
cIXP:~$ ls ~
IXPBuilder_v5.0.tgz  ixp

cIXP:~$ cd ~/ixp/tools
cIXP:~/ixp/tools$ sudo ./ixp-install.sh



mIXP
mIXP:~$ tar -xzvf ~/IXPBuilder_v5.0.tgz
mIXP:~$ ls ~
IXPBuilder_v5.0.tgz  ixp

mIXP:~$ cd ~/ixp/tools
mIXP:~/ixp/tools$ sudo ./ixp-install.sh
```

Figure 90: Install IXPBuilder on both servers

5.8.2.1.1 cIXP site build

Build the cIXP using switching type: *software-defined* as demonstrated in Figure 91. Test the configuration on this site as shown for the single site using the *ixp peer list* and *ixp peer status* commands (Sections 5, 6, 8, 9 and 12, Appendix A).

```
cIXP:~$ ixp schema build –sw soft
cIXP:~$ ixp host build –s 0
cIXP:~$ ixp server build –y all
cIXP:~$ ixp software install all
cIXP:~$ ixp software configure all
cIXP:~$ ixp peer add –n C1 –a 5111 –d one.com
cIXP:~$ ixp peer add –n C2 –a 5222 –d two.com –rs no
cIXP:~$ ixp peer add –n C3 –a 5333 –d three.com –bs no
```

Figure 91: Build cIXP using software-defined switching type

5.8.2.1.2 mIXP site build

Build an mIXP using switching type: *software-defined* and site type: *mini* as shown in Figure 92. This will generate Model S (Sections 5, 6, 8, 9, Appendix A).

```
mIXP:~$ ixp schema build –st mini –sw soft –sn 2 –p4 177.7.7.0/24 –p6
2a77:7:7::0/48 –m4 176.6.6.0/24 –m6 2a76:6:6::0/48 –as 7999 –t Gulu –e 1100
–la '02 46 40.2314 N' –lo '32 17 14.1243 E'

mIXP:~$ ixp host build –s 0
mIXP:~$ ixp server build –y all
mIXP:~$ ixp software install all
mIXP:~$ ixp software configure all
```

Figure 92: Build mIXP schema and build the system

5.8.2.2 Check routing

Confirm connectivity between the two sites. This may involve adding routes to the two servers. The commands shown in Figure 93 are for demonstration only and in terms of a testbed. Connectivity configuration on production sites will depend on network design.

cIXP

```
cIXP:~$ sudo ip route add 176.6.6.0/24 via 198.8.8.1

cIXP:~$ fping 176.6.6.230
176.6.6.230 is alive
```

mIXP

```
mIXP:~$ sudo ip route add 198.8.8.0/24 via 176.6.6.1

mIXP:~$ fping 198.8.8.230
198.8.8.230 is alive
```

Figure 93: Routing and test connectivity from cIXP to mIXP

5.8.2.3 mIXP site configuration

Generate RSA keys at the cIXP and return the public key to STDOUT on the terminal as demonstrated in Figure 94. The public key must be added, verbatim, to the mIXP which is shown in Figure 95 (Section 14, Appendix A).

```
cIXP:~$ ixp remote key generate
cIXP:~$ ixp remote key show
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDGFLA6tIqgn4PmakWjwyjZ3VzwjcVFmk6ANyTzht9YAaU
7oXEHsVNMft/LL/dWNAt8Dj7T/JkZbmlyB4owR/
3rabRmMHlWIy4CftmusV13JeBFWC2d1YzCGbjs7RwVV1C2Z0UmeN9MtjCYvUEzNo1ymx9cOt9EA
sKFO2Q6AoAUnlnhp9S69Ompi98iQxbDOvJgxeKu8KQqSekz5+jn9qPr+0yA7XQKTzp9Ht8+Kt7a
jEAJV0gWIZa5iiZKNaZJMv9NO7GBzIziwHRu3tx+lUjuPOxvTP7PKlARUOU/
bs29SViGgS7P1f1+BsFXTmxiC/i6OD3HWeWepzId/0llxyWF
```

Figure 94: Generate public/private key pair on cIXP

Add public key displayed by the cIXP to the mIXP to enable secure communications between them (Section 14, Appendix A).

```
mIXP:~$ ixp remote key enter

  Paste the public key test here: ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDGFLA6tIqgn4PmakWjwyjZ3VzwjcVFmk6ANyTzht9YAaU
7oXEHsVNMft/LL/dWNAt8Dj7T/JkZbmlyB4owR/
3rabRmMHlWIy4CftmusV13JeBFWC2d1YzCGbjs7RwVV1C2Z0UmeN9MtjCYvUEzNo1ymx9cOt9EA
sKFO2Q6AoAUnlnhp9S69Ompi98iQxbDOvJgxeKu8KQqSekz5+jn9qPr+0yA7XQKTzp9Ht8+Kt7a
jEAJV0gWIZa5iiZKNaZJMv9NO7GBzIziwHRu3tx+lUjuPOxvTP7PKlARUOU/
bs29SViGgS7P1f1+BsFXTmxiC/i6OD3HWeWepzId/0llxyWF
```

Figure 95: Add public key to the mIXP

Figure 96 demonstrates the remote mIXP site details being added to the cIXP. Configure the peer information in the mIXP from the cIXP as demonstrated in Figure 97 (Section 14, Appendix A).

```
cIXP:~$ ixp remote site add –sn 2 –se Gulu –h4 176.6.6.230 –h6 2a76:6:6::230
```

Figure 96: Add remote mIXP to the cIXP

```
cIXP:~$ ixp remote cmd 2 -- ixp peer add -n M1 -a 7111  -d one.net

cIXP:~$ ixp remote cmd 2 -- ixp peer add -n M2 -a 7222  -d two.net -rs no

cIXP:~$ ixp remote cmd 2 -- ixp peer add -n M3 -a 7333  -d three.net -bs no
```

Figure 97: Add remote IXP peers at the mIXP from the cIXP

Test the configuration on the mIXP from the cIXP as shown in Figure 98.

```
cIXP:~$ ixp remote cmd 2 -- ixp peer list

## Executing on remote mIXP site: 2

  [ mIXP: 2 ]~$ ixp peer list

  + ------------------------------------------------------------------------ +
  |                              IPv4 Peering Table                          |
  + ---- + ------ + ------ + ---- + --------- + ----- + ----- + ------------- +
  | Name |  ASN   | Switch | Port |  Domain   |  RS   | AS112 |   IP Address  |
  + ---- + ------ + ------ + ---- + --------- + ----- + ----- + ------------- +
  |  M1  |  7111  |   0    |  3   |  one.net  |  yes  |  yes  | 177.7.7.2/24  |
  |  M2  |  7222  |   0    |  4   |  two.net  |  no   |  yes  | 177.7.7.3/24  |
  |  M3  |  7333  |   0    |  5   | three.net |  yes  |  no   | 177.7.7.4/24  |
  + ---- + ------ + ------ + ---- + --------- + ----- + ----- + ------------- +
  + ------------------------------------------------------------------------ +
  |                              IPv6 Peering Table                          |
  + ---- + ------ + ------ + ---- + --------- + ----- + ----- + ---------------+
  | Name |  ASN   | Switch | Port |   Domain  |  RS   | AS112 |    IP Address   |
  + ---- + ------ + ------ + ---- + --------- + ----- + ----- + ---------------+
  |  M1  |  7111  |   0    |  3   |  one.net  |  yes  |  yes  | 2a77:7:7::2/48  |
  |  M2  |  7222  |   0    |  4   |  two.net  |  no   |  yes  | 2a77:7:7::3/48  |
  |  M3  |  7333  |   0    |  5   | three.net |  yes  |  no   | 2a77:7:7::4/48  |
  + ---- + ------ + ------ + ---- + --------- + ----- + ----- + ---------------+

cIXP:~$ ixp remote cmd 2 -- ixp peer status

## Executing on remote mIXP site: 2

  [ mIXP: 2 ]~$ ixp peer status

  + ---------------------------------------- +
  |            IPv4 BGP State Table          |
  + -------- + ------ + ------- + ------------ +
  |  Server  |  Name  |  State  |     Info     |
  + -------- + ------ + ------- + ------------ +
  |   rs2    | M1     | up      | Established  |
  |          | M3     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
  |   cs2    | M1     | up      | Established  |
  |          | M2     | up      | Established  |
  |          | M3     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
  |   bs2    | M1     | up      | Established  |
  |          | M2     | up      | Established  |
  + -------- + ------ + ------- + ------------ +

  + ---------------------------------------- +
  |            IPv6 BGP State Table          |
  + -------- + ------ + ------- + ------------ +
  |  Server  |  Name  |  State  |     Info     |
  + -------- + ------ + ------- + ------------ +
  |   rs2    | M1     | up      | Established  |
  |          | M3     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
  |   cs2    | M1     | up      | Established  |
  |          | M2     | up      | Established  |
  |          | M3     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
  |   bs2    | M1     | up      | Established  |
  |          | M2     | up      | Established  |
  + -------- + ------ + ------- + ------------ +
```

Figure 98: IXP peer list and status at the mIXP from the cIXP

5.8.3 General notes

Timings were extracted by prepending the *time* shell command before each *ixp* command. While the testing was carried out in Kampala, Uganda, the Ubuntu repository selected was *ke* not *ug*. This is because the Ugandan repository held at the Research and Education Network for Uganda (RENU) reported a *last update unknown* from the mirror site. The Kenya Education Network (KENET) repository reported *one day behind* and was therefore a preferable selection. The Internet connection was provided by Smile Communications Uganda Limited via a fixed wireless 4G LTE Internet connection in the Nakasero area of Kampala.

5.9 Functionality testing

5.9.1 Continuity testing between hosts

The primary function of an IXP is to provide local connectivity between the end-users of its members to services or end-users on other member networks. For the network in Figure 30 on page 100 a connectivity test was completed between each subscriber host for both IPv4 and IPv6. This has been summarised in Table 13 and detailed results are displayed in Figures 98 - 101. Before testing three peers were established on each IXP, connected to the *CS*, *RS* and *BS*.

Table 13: Host connectivity test

	Host C1 v4, v6	**Host C2 v4, v6**	**Host C3 v4, v6**	**Host M1 v4, v6**	**Host M2 v4, v6**	**Host M3 v4, v6**
Host C1		✓ ✓	✓ ✓	✗ ✗	✗ ✗	✗ ✗
Host C2	✓ ✓		✓ ✓	✗ ✗	✗ ✗	✗ ✗
Host C3	✓ ✓	✓ ✓		✗ ✗	✗ ✗	✗ ✗
Host M1	✗ ✗	✗ ✗	✗ ✗		✓ ✓	✓ ✓
Host M2	✗ ✗	✗ ✗	✗ ✗	✓ ✓		✓ ✓
Host M3	✗ ✗	✗ ✗	✗ ✗	✓ ✓	✓ ✓	

Hosts of ISPs connected to the same IXP have connectivity while hosts connected to ISPs that are connected to different IXPs in the distributed set do not. This is to be expected as peering is confined to members connected to each IXP. Of course hosts would be able to establish connectivity to each other in reality as the ISPs would have upstream transit by which they could connect via a Rendezvous Point (RP). This test demonstrates that such connectivity is not through the IXP.

```
host@C1:~$ mtr --report 199.2.2.100
Start: Sat Apr 20 13:46:36 2019
HOST: C1                          Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 199.1.1.1                 0.0%    10    0.8   0.8   0.7   0.8   0.0
  2.|-- 199.9.9.3                 0.0%    10    1.0   1.0   1.0   1.2   0.0
  3.|-- 199.2.2.100               0.0%    10    0.9   0.9   0.9   1.0   0.0

host@C1:~$ mtr --report 199.3.3.100
Start: Sat Apr 20 13:48:11 2019
HOST: C1                          Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 199.1.1.1                 0.0%    10    0.8   0.8   0.8   0.9   0.0
  2.|-- 199.9.9.4                 0.0%    10    0.8   0.8   0.7   1.1   0.0
  3.|-- 199.3.3.100               0.0%    10    1.1   1.0   1.0   1.1   0.0

host@C1:~$ fping 177.1.1.100
177.1.1.100 is unreachable

host@C1:~$ fping 177.2.2.100
177.2.2.100 is unreachable

host@C1:~$ fping 177.3.3.100
177.3.3.100 is unreachable

host@C2:~$ mtr --report 199.1.1.100
Start: Sat Apr 20 13:50:22 2019
HOST: C2                          Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 199.2.2.1                 0.0%    10    0.8   0.9   0.8   1.7   0.0
  2.|-- 199.9.9.2                 0.0%    10    1.0   1.0   0.9   1.2   0.0
  3.|-- 199.1.1.100               0.0%    10    1.0   0.9   0.9   1.0   0.0

host@C2:~$ mtr --report 199.3.3.100
Start: Fri Dec  8 13:51:51 2019
HOST: C2                          Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 199.2.2.1                 0.0%    10    0.8   0.8   0.8   0.9   0.0
  2.|-- 199.9.9.4                 0.0%    10    0.8   0.8   0.7   1.0   0.0
  3.|-- 199.3.3.100               0.0%    10    1.1   1.0   1.0   1.1   0.0

host@C2:~$ fping 177.1.1.100
177.1.1.100 is unreachable

host@C2:~$ fping 177.2.2.100
177.2.2.100 is unreachable

host@C2:~$ fping 177.3.3.100
177.3.3.100 is unreachable

host@C3:/$ mtr --report 199.1.1.100
Start: Sat Apr 20 13:53:19 2019
HOST: C3                          Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 199.3.3.1                 0.0%    10    0.6   2.6   0.6  20.5   6.3
  2.|-- 199.9.9.2                 0.0%    10    1.2   1.2   1.1   1.3   0.0
  3.|-- 199.1.1.100               0.0%    10    1.1   1.1   1.0   1.2   0.0

host@C3:/$ mtr --report 199.2.2.100
Start: Sat Apr 20 13:54:41 2019
HOST: C3                          Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 199.3.3.1                 0.0%    10    0.6   0.6   0.6   0.7   0.0
  2.|-- 199.9.9.3                 0.0%    10    1.2   1.2   1.1   1.4   0.0
  3.|-- 199.2.2.100               0.0%    10    1.1   1.1   1.0   1.1   0.0

host@C3:~$ fping 177.1.1.100
177.1.1.100 is unreachable

host@C3:~$ fping 177.2.2.100
177.2.2.100 is unreachable

host@C3:~$ fping 177.3.3.100
177.3.3.100 is unreachable
```

Figure 99: IPv4 host connectivity tests on cIXP

```
host@C1:~$ mtr --report 2a99:2:2::100
Start: Sat Apr 20 18:31:24 2019
HOST: C1                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a99:1:1::1             0.0%    10    0.8   0.8   0.7   0.9   0.0
  2.|-- 2a99:9:9::3             0.0%    10    1.4   1.4   1.3   1.7   0.0
  3.|-- 2a99:2:2::100           0.0%    10    1.8   1.8   1.8   1.9   0.0

host@C1:~$ mtr --report 2a99:3:3::100
Start: Sat Apr 20 18:23:43 2019
HOST: C1                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a99:1:1::1             0.0%    10    0.8   0.8   0.8   0.9   0.0
  2.|-- 2a99:9:9::4             0.0%    10    1.3   1.3   1.2   1.4   0.0
  3.|-- 2a99:3:3::100           0.0%    10    1.5   1.6   1.5   1.8   0.0

host@C1:~$ fping 2a77:1:1::100
2a77:1:1::100 is unreachable

host@C1:~$ fping 2a77:2:2::100
2a77:2:2::100 is unreachable

host@C1:~$ fping 2a77:3:3::100
2a77:3:3::100 is unreachable

host@C2:~$ mtr --report 2a99:1:1::100
Start: Sat Apr 20 18:26:48 2019
HOST: C2                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a99:2:2::1             0.0%    10    0.8   0.8   0.7   0.9   0.0
  2.|-- 2a99:9:9::2             0.0%    10    1.5   1.7   1.4   4.3   0.7
  3.|-- 2a99:1:1::100           0.0%    10    1.8   1.8   1.8   1.8   0.00

host@C2:~$ mtr --report 2a99:3:3::100
Start: Sat Apr 20 18:28:51 2019
HOST: C2                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a99:2:2::1             0.0%    10    0.8   0.8   0.8   0.9   0.0
  2.|-- 2a99:9:9::4             0.0%    10    1.3   1.3   1.2   1.5   0.0
  3.|-- 2a99:3:3::100           0.0%    10    1.5   1.6   1.5   1.6   0.0

host@C2:~$ fping 2a77:1:1::100
2a77:1:1::100 is unreachable

host@C2:~$ fping 2a77:2:2::100
2a77:2:2::100 is unreachable

host@C2:~$ fping 2a77:3:3::100
2a77:3:3::100 is unreachable

host@C3:/$ mtr --report 2a99:1:1::100
Start: Sat Apr 20 18:33:32 2019
HOST: C3                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a99:3:3::1             0.0%    10    0.7   0.7   0.6   1.0   0.0
  2.|-- 2a99:9:9::2             0.0%    10    1.1   1.8   1.1   7.9   2.1
  3.|-- 2a99:1:1::100           0.0%    10    1.6   1.7   1.4   3.0   0.3

host@C3:/$ mtr --report 2a99:2:2::100
Start: Sat Apr 20 18:36:21 2019
HOST: C3                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a99:3:3::1             0.0%    10    0.7   0.7   0.6   0.7   0.0
  2.|-- 2a99:9:9::3             0.0%    10    1.1   1.8   1.1   8.2   2.1
  3.|-- 2a99:2:2::100           0.0%    10    1.5   1.6   1.5   2.2   0.0

host@C3:~$ fping 2a77:1:1::100
2a77:1:1::100 is unreachable

host@C3:~$ fping 2a77:2:2::100
2a77:2:2::100 is unreachable

host@C3:~$ fping 2a77:3:3::100
2a77:3:3::100 is unreachable
```

Figure 100: IPv6 host connectivity tests on cIXP

```
host@M1:~$ mtr --report 177.2.2.100
Start: Tue Apr  23 16:20:05 2019
HOST: M1                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 177.1.1.1               0.0%    10    0.8   0.8   0.8   0.9   0.0
  2.|-- 177.7.7.3               0.0%    10    1.0   1.0   0.9   1.0   0.0
  3.|-- 177.2.2.100             0.0%    10    1.0   1.2   0.9   3.3   0.6

host@M1:~$ mtr --report 177.3.3.100
Start: Tue Apr  23 9 16:21:53 2019
HOST: M1                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 177.1.1.1               0.0%    10    0.8   0.8   0.8   0.8   0.0
  2.|-- 177.7.7.4               0.0%    10    0.8   0.9   0.8   1.1   0.0
  3.|-- 177.3.3.100             0.0%    10    1.1   1.1   1.0   1.1   0.0

host@M1:~$ fping 199.1.1.100
199.1.1.100 is unreachable

host@M1:~$ fping 199.2.2.100
199.2.2.100 is unreachable

host@M1:~$ fping 199.3.3.100
199.3.3.100 is unreachable

host@M2:~$ mtr --report 177.1.1.100
Start: Tue Apr  23 16:26:07 2019
HOST: M2                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 177.2.2.1               0.0%    10    0.8   0.8   0.8   0.8   0.0
  2.|-- 177.7.7.2               0.0%    10    1.0   1.0   0.9   1.4   0.0
  3.|-- 177.1.1.100             0.0%    10    0.9   0.9   0.9   1.0   0.0

host@M2:~$ mtr --report 177.3.3.100
Start: Tue Apr  23 16:27:33 2019
HOST: M2                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 177.2.2.1               0.0%    10    0.8   0.8   0.8   0.8   0.0
  2.|-- 177.7.7.4               0.0%    10    0.8   0.8   0.8   1.1   0.0
  3.|-- 177.3.3.100             0.0%    10    1.1   1.1   1.0   1.2   0.0

host@M2:~$ fping 199.1.1.100
199.1.1.100 is unreachable

host@M2:~$ fping 199.2.2.100
199.2.2.100 is unreachable

host@M2:~$ fping 199.3.3.100
199.3.3.100 is unreachable

host@M3:/$ mtr --report 177.1.1.100
Start: Tue Apr  23 16:32:41 2019
HOST: M3                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 177.3.3.1               0.0%    10    0.6   0.6   0.6   0.6   0.0
  2.|-- 177.7.7.2               0.0%    10    1.1   1.2   1.1   1.4   0.0
  3.|-- 177.1.1.100             0.0%    10    1.1   1.1   1.0   1.1   0.0

host@M3:/$ mtr --report 177.2.2.100
Start: Tue Apr  23 16:33:56 2019
HOST: M3                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 177.3.3.1               0.0%    10    0.6   0.6   0.6   0.7   0.0
  2.|-- 177.7.7.3               0.0%    10    1.2   1.2   1.1   1.3   0.0
  3.|-- 177.2.2.100             0.0%    10    1.1   1.1   1.0   1.1   0.0

host@M3:~$ fping 199.1.1.100
199.1.1.100 is unreachable

host@M3:~$ fping 199.2.2.100
199.2.2.100 is unreachable

host@M3:~$ fping 199.3.3.100
199.3.3.100 is unreachable
```

Figure 101: IPv4 host connectivity tests on mIXP

```
host@M1:~$ mtr --report 2a77:2:2::100
Start: Tue Apr  23 16:23:18 2019
HOST: M1                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a77:1:1::1             0.0%    10    0.8   0.9   0.8   1.6   0.0
  2.|-- 2a77:7:7::3             0.0%    10    1.4   1.7   1.4   4.2   0.7
  3.|-- 2a77:2:2::100           0.0%    10    1.8   1.8   1.8   1.9   0.0

host@M1:~$ mtr --report 2a77:3:3::100
Start: Tue Apr  23 16:24:23 2019
HOST: M1                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a77:1:1::1             0.0%    10    0.8   0.8   0.7   0.8   0.0
  2.|-- 2a77:7:7::4             0.0%    10    1.2   1.3   1.2   1.5   0.0
  3.|-- 2a77:3:3::100           0.0%    10    1.6   1.5   1.4   1.6   0.0

host@M1:~$ fping 2a99:1:1::100
2a99:1:1::100 is unreachable

host@M1:~$ fping 2a99:2:2::100
2a99:2:2::100 is unreachable

host@M1:~$ fping 2a99:3:3::100
2a99:3:3::100 is unreachable

host@M2:~$ mtr --report 2a77:1:1::100
Start: Tue Apr  23 16:28:53 2019
HOST: M2                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a77:2:2::1             0.0%    10    0.8   0.8   0.7   0.9   0.0
  2.|-- 2a77:7:7::2             0.0%    10    1.4   1.4   1.3   1.6   0.0
  3.|-- 2a77:1:1::100           0.0%    10    1.8   1.8   1.8   1.9   0.0

host@M2:~$ mtr --report 2a77:3:3::100
Start: Tue Apr  23 16:30:07 2019
HOST: M2                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a77:2:2::1             0.0%    10    0.8   0.8   0.7   0.8   0.0
  2.|-- 2a77:7:7::4             0.0%    10    1.3   1.3   1.2   1.6   0.0
  3.|-- 2a77:3:3::100           0.0%    10    1.6   1.5   1.5   1.6   0.0

host@M2:~$ fping 2a99:1:1::100
2a99:1:1::100 is unreachable

host@M2:~$ fping 2a99:2:2::100
2a99:2:2::100 is unreachable

host@M2:~$ fping 2a99:3:3::100
2a99:3:3::100 is unreachable

host@M3:/$ mtr --report 2a77:1:1::100
Start: Tue Apr  23 16:35:13 2019
HOST: M3                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a77:3:3::1             0.0%    10    0.6   0.7   0.6   0.7   0.0
  2.|-- 2a77:7:7::2             0.0%    10    1.1   1.1   1.1   1.4   0.0
  3.|-- 2a77:1:1::100           0.0%    10    1.5   1.6   1.4   1.6   0.0

host@M3:/$ mtr --report 2a77:2:2::100
Start: Tue Apr  23 16:36:27 2019
HOST: M3                        Loss%   Snt   Last  Avg   Best  Wrst StDev
  1.|-- 2a77:3:3::1             0.0%    10    0.7   0.7   0.6   0.7   0.0
  2.|-- 2a77:7:7::3             0.0%    10    1.1   1.1   1.0   1.2   0.0
  3.|-- 2a77:2:2::100           0.0%    10    1.5   1.5   1.5   1.6   0.0

host@M3:~$ fping 2a99:1:1::100
2a99:1:1::100 is unreachable

host@M3:~$ fping 2a99:2:2::100
2a99:2:2::100 is unreachable

host@M3:~$ fping 2a99:3:3::100
2a99:3:3::100 is unreachable
```

Figure 102: IPv6 host connectivity tests on mIXP

5.10 Basic Usability testing

A key requirement of the PoC is simplicity. From the functional specification in section 5.2, *be simple to build and maintain*. Each IXP should be simple to install and given basic schema data, automate the build process of the required exchange. To test this the PoC software, IXPBuilder, along with the IXPBuilder manual (Appendix A), was given to two separate groups of undergraduate BSc in Telecommunications Engineering students at the College of Engineering, Design, Art and Technology, Makerere University to test the install-ability and the ease of build.

The first group were fourth year undergraduate students who were working on their final year project on ISP Interconnectivity in Uganda. The students wanted to develop an understanding of IXPs so they were ideally placed to undertake the testing function. They were given IXPBuilder software version 2.0 and the associated IXPBuilder manual in November 2017. The group installed the software on a Dell laptop using Oracle VirtualBox and demonstrated the functionality in a tech-talk at netLabs!UG research centre using four MikroTik RB941-2nD hAP lite devices as the *traditional* Ethernet switch and peer routers. Standard laptops were used at each peer to act as clients.

The second group were a mix of undergraduate students who volunteered at netLabs!UG research centre. This group were given IXPBuilder version 5.1 and the corresponding manual (Appendix A) in March 2019 to install and test the functionality. The group installed IXPBuilder on a Dell PowerEdge R730 server with four Ethernet interfaces. They built a testbed in *traditional* mode with a Cisco 3750 48-port 1G switch connected to the third interface on the server. One MikroTik RB941-2nD hAP lite device was configured as a peer router for the network ISP1 and was connected to the final interface on the IXPBuilder server. Two other MikroTik RB941-2nD hAP lite devices were configured as peer routers for ISP2 and ISP3 and were connected to the first two interfaces configured as peer interfaces on the Cisco switch. Three Dell laptops were used as clients connected to each peer router. Initial connectivity tests were performed and the clients on ISP1 and ISP2 could reach each other but both were unable to reach ISP3 but after some troubleshooting the team rectified the problem and all clients passed connectivity tests to eachother.

5.11 Summary

Using the design specification, developed through consultation with experts, the PoC was developed. This PoC established that by leveraging NFV principles, but using containerisation for additional efficiency, it is possible to isolate the essential functions of an IXP into individual LXCs on an LXD. The PoC developed five models that incorporate *traditional* Ethernet switches at the *switching tier* while also developing an SDX via a set of three additional SDN models. These models incorporate an *SC* function hosted on an additional LXC at the *core tier* that controls internal and external OF Ethernet switches at the *switching tier*. The *SC* also exposes a NBI for applications who can manipulate the flow tables of OF switches, over the SBI control-channel, for future functionality that can be employed at the SDX. Additionally the PoC facilitates the build of a dIXP, a set including a cIXP and at least one mIXP, where each IXP remains independent from a peering perspective but the day-to-day management of each mIXP can be centralised at the cIXP. In order to aid deployment the IXPBuilder operations manual was produced.

6. Discussion of Results

6.1 Introduction

Hub towns and cities in developing countries require a fully developed Internet ecosystem that includes an active Internet eXchange Point (IXP). There is significant migration in developing countries from rural to urban areas in particular into capital cities. People migrate in search of employment opportunities and better services. Mitigating this migration and supporting local communities empowers local development. A key element of the solution is to strengthen spatial planning by build IXPs as part of the Internet ecosystem in these hub towns and cities. Developing this solution; however, is exacerbated by a lack of local skills in order to manage and maintain independent IXPs. This leads to a solution where a system is developed that be deployed locally but managed centrally. This research has demonstrated that one system will not suffice and therefore the PoC has 11 possible models and sub-models depending on the scenario.

6.2 Discussion

This research conducted a political economy study and survey of the Internet ecosystem in East Africa since the landing of the submarine fibre-optic cables at Mombasa and Dar-es-Salaam in 2009 (Appendix B). Taken together with the literature review, this provided an overarching analysis of Internet development in East Africa over the last decade. The study also explores the future direction of the Internet ecosystem in East Africa in the context of new technologies such as Software Defined Networking (SDN) and Network Functions Virtualisation (NFV).

The literature review also highlighted the impact of increased levels of Internet penetration on Gross Domestic Product (GDP) (Edquist *et al.*, 2018). A review of the Sustainable Development Goals (SDG) highlighted that Internet penetration is a key tool that can encourage the redistribution of jobs from capital cities through effective spatial strategies (U.N., 2015) and (U.N., 2017). The literature review of systems deployed in other countries demonstrates that they are unsuitable for this context as the combined challenges of simplicity, affordability, adaptability and scalability are not considered together.

Both the literature review and the mixed method study identified IXPs as a key pillar of the Internet ecosystem providing a positive effect on Internet traffic latency, hop count, packet loss and jitter (Chatzis, Smaragdakis, *et al.*, 2013), (Norton, 2014) and (Di Lallo, 2015). Examples have highlighted, demonstrating that by increasing the number of IXPs, further increases in network speed as well as reduced costs can be achieved within the locale of each exchange (Brito *et al.*, 2016).

Examples from Europe in particular highlight a reluctance among IXP organisations to link individual IXPs for the purpose of peering, preferring instead to maintain the independence of each IXP (Gorey, 2016) (LINX, 2017). Such reluctance, *the IXP interconnection hazard*, has its origins in the potential risk that IXP organisations could become competitors in the transit and back-haul space, placing them in competition with their most important customers, Internet Service Providers (ISP). From the mixed methods study the Subject Matter Experts (SME) were clear on the impact that IXPs make on the regional Internet ecosystem. They also considered IXPs to be a component of the solution to spatial development; however, there was some doubt if it could happen in reality due to local skills defects. This research has demonstrated that it is possible and the new models developed can address this issue.

Literature has suggested, demonstrated through some limited examples; (Stringer *et al.*, 2014), (Mambretti, Chen and Yeh, 2014b), (Gupta *et al.*, 2016), (Lapeyrade, Bruyère and Owezarski, 2016), (Bruyere *et al.*, 2018), (Chiesa *et al.*, 2016) and (Antichi *et al.*, 2017) that the incorporation of SDN into IXPs to form Software Defined eXchanges (SDX) offers the potential to redefine the IXP and the Internet ecosystem into the future. SDXs can potentially offer new approaches to inter-domain and multi-domain routing problems, DDoS mitigation problems as well as interlinking SDN islands in a future where SDN has replaced the traditional AS system. The SMEs in the mixed methods study understood the potential for disruptive technologies such as SDN and NFV to increase automation and deliver flexibility to their businesses.

The Proof of Concept (PoC) section of this research was developed based a high level

functional specification that was devised from points drawn from the literature review, the research gap as well as the mixed methods study of the Internet in East Africa. The PoC demonstrates that through the containerisation of functions, with an Open virtual Switch (OvS) and novel IXP models implemented through custom software (called IXPBuilder) it is possible to build a distributed set of IXPs or SDXs that are managed centrally while maintaining the independence of each IXP for the purpose of peering. It also demonstrates that, even in a mini SDX, it is possible to expose the flow tables of the underlying hardware to future applications. The PoC provides a new insight into the relationship between IXP management and IXP peering substrates and demonstrates that it is both mindful of *the IXP interconnection hazard*, while resolving the skills gap identified by the SMEs through centralised management.

The results indicate that the PoC is both usable and functional either stand-alone or as part of a dIXP. Results also demonstrate that each IXP node within the dIXP can operate as an IXP or SDX in small sites without additional switching hardware, as an IXP with *traditional* managed Ethernet switches or as an SDX with OpenFlow (OF) Ethernet switches. Through the selection of appropriate Common Off The Shelf (COTS) hardware to host the PoC as well as through leveraging free of cost open source software, the PoC can also be considered affordable.

6.3 Limitations

There were two main limitations associated with this research. The first relates to the mixed methods study and the inability to access Burundi and South Sudan due to ongoing Irish Department of Foreign Affairs and Trade travel advisories against all travel by Irish citizens. When Internet penetration is considered for example in Figure 8 on page 38 it was clear that both of these countries fell well short of their peers within East Africa. Such access would have enabled the researcher to conclude in richer detail why this is the case.

The second limitation was financial. While the researcher is grateful for the financial support given by the Uganda Communications Commission (UCC) to acquire equipment for the project, more server hardware would have permitted the creation of a larger testbed that would have

facilitated testing of a bigger distributed network of IXPs. Due to this limitation it was necessary to repurpose the hardware to different roles between tests.

In addition to the server limitation, the availability of OF Ethernet switch hardware was also limited to that made available by netLabs!UG research centre at Makerere University. netLabs!UG has Netgear M4300-28G switches that supported the OF protocol at the control plane. Unfortunately this switch did not work well and after raising an issue with Netgear support they eventually responded that Netgear engineering confirmed that the switch only supports a *qualified* OF v1.3 solution when using a specific version of the OpenDaylight (ODL) SDN Controller, namely Helium (0.2.4). This switch does not appear on the Open Networking Foundation (ONF) OF Conformant: Certified Product List ((ONF, no date)). [Netgear Technical Support Case: 40988137]. In order to resolve the problem a second Dell PowerEdge R610 server was employed and converted into an OF Ethernet switch by deploying OvS software on it (Section 15.2.1, Appendix A).

6.4 Summary

Developed countries have enjoyed the privilege of Internet access for decades and their infrastructure is highly developed and robust. However, it has been shown that these systems are of a scale that is unsuitable for deployment in East Africa and therefore new bespoke models have been developed. It was highlighted that there was an empirical link between increased levels of Internet penetration on GDP as well as the a key tool to encourage the redistribution of jobs from capital cities. IXPs were identified as a key pillar of the Internet ecosystem which provides a positive effect on Internet traffic latency, hop count, packet loss and jitter and that by increasing the number of IXPs, further increases in network speed as well as reduced costs can be achieved within the locale of each exchange. The IXP interconnection hazard was described as the potential risk that IXP organisations could become competitors in the transit and back-haul space, placing them in competition with their most important customers. It was considered that while the impact that IXPs make on the regional Internet ecosystem is positive, it is a component of the solution to spatial

development; however, there was some doubt if it could happen in reality due to local skills defects.

Literature has suggested that the incorporation of SDN into IXPs to form SDX has the potential to redefine the IXP with new approaches to inter-domain and multi-domain routing problems as well as interlinking SDN islands in future networks. It was therefore considered necessary to incorporate SDX functionality into the PoC. This functionality has exposed the OF switch flow tables via a RESTful API that demonstrate the feasibility of additional IXP applications.

It was demonstrated that it is possible to build a distributed set of IXPs or SDXs that are managed centrally while maintaining the independence of each IXP for the purpose of peering. This can be achieved through the containerisation of functions, use of OvS as a softswitch and IXP models implemented through the custom IXPBuilder software developed as part of this research. The PoC provides a new insight into the relationship between IXP management and IXP peering substrates and demonstrates that it is both mindful of the IXP interconnection hazard, while resolving the skills gap identified by the SMEs through centralised management.

7. Conclusions, Recommendations and Future work

7.1 Introduction

This final chapter starts by offering some general concluding remarks. It then proceeds to lists the research questions posed in chapter 1 and outlines how this research answers each question.

A number of recommendations are made, both from a political and social perspective to address the challenge of migration to capital cities as well as some considerations to be considered when productising the IXP models in the PoC.

The chapter concludes with views as to the future direction switching and routing will take as well as a suggestion as to further work that can follow from this research.

7.2 Conclusions

In order to strengthen the Internet ecosystem as a key component of spatial planning in East Africa and other developing countries, it is necessary to supplement the national Internet eXchange Point (IXP) with additional IXPs within hub towns and cities. IXPs in combination with other elements such as Internet Data Centres (IDC) create the necessary Internet ecosystem that is required to support the development of local technology hubs.

This research demonstrates that a fully functional and affordable distributed set of IXPs (dIXP) can be achieved by exploiting the most suitable model from the Proof of Concept (PoC) at each site. Mini IXPs (mIXP) at remote sites can be maintained using centralised management from a core IXP (cIXP), thereby addressing the issue of skills deficits. Eleven new models and sub-models have been developed and can be deployed in many different settings. These solutions are location neutral and are transferable to similar contexts other developing countries.

The PoC developed IXP and Software Defined eXchange (SDX) options that are stand-alone, IXP options that are expanded through the use of external *traditional* managed Ethernet switches as well as software-defined options using external OpenFlow (OF) Ethernet switches to create an SDX. Use of the PoC in SDX mode can facilitate further application development in the

future.

7.2.1 The status of the Internet in East Africa

The combination of a literature review as well as the political economy study (Appendix B) triangulated by a survey was used to answer the question, *what are the political, social, economic and technical drivers that have influenced Internet development in East Africa over the last decade?*. The study concluded that over the decade since the landing of the submarine fibre-optic cables on the East African coast the Internet experience and infrastructure has *much improved* and that streaming video services has improved due to the addition of Content Delivery Networks (CDN) by Internet Content Providers (ICP), as well as improvements in the Internet infrastructure.

IXPs have had a positive impact on the Internet since the fibre-optic cables landed and have a future continuing as a point where ISPs, Application Service Providers (ASP) and ICPs peer. The study also proposes that IXPs need to evolve beyond national capitals to support hub towns and cities. The models developed as part of this research can support this process.

Participants from the study are unsure as to whether *software-defined* technologies such as Software Defined Networking (SDN) and Network Functions Virtualisation (NFV) will change the nature of the Internet for end-users and their interaction with it. However, there was census that these new disruptive technologies have the potential to increase automation and add flexibility to ISP networks as well as to deliver improved services to their end-users.

The rapid change in the nature of the Internet will continue to present significant challenges to the national regulator system. Regulators will need to take a macro level view of what is happening globally and regionally in addition to their focus on the national context. There is a convergence of services delivered by a few very large international providers and this presents a potential risk of the emergence of a monopolist or oligopolistic situation in the future. Many participants were of the considered opinion "*that regulators today seem ill equipped to deal with multinational monopolies as they gain dominance over user behaviour and control of content*".

The Internet of the future in East Africa will be driven by Fibre to the Home (FTTH) (in

large urban areas), 4G Long Term Evolution (LTE); and content cashing according to both respondents and participants. Wireless services based on 4G LTE and 5G New Radio (NR) will remain the main access method to the Internet and will continue to do so into the future for the majority of the connected population. Mobile Money (MoMo), FINancial TECHnology (FINTECH), moving more government services to digital platforms and local content creation are also seen as key elements of a healthy future Internet in the region. In additions the development of hotspots in villages to improve access for rural citizens, as well as the improved availability of cheap smartphone devices are seen as critical issues to increase Internet penetration. As new local content and an expanded user base join the Internet to access it, the IXP footprint, based on the models developed in this research, can facilitate keeping the *local traffic local*.

7.2.2 IXP Proof of Concept

The second research question, *how can models of IXP be developed to cater for local IXPs in the context of developing countries?,* has been answered through the development of a PoC called IXPBuilder. Through testbed experiments and software development, system models have been identified and developed that can deliver a key set of IXP nodes in the future Internet ecosystem, within the constraints of a developing nation. The PoC was designed and developed using cost free, open source software which leverages containerisation technology ensuring that the hardware specifications of the required servers are minimised. It can be concluded that the 11 PoC models and sub-models are also affordable using both *traditional* managed Ethernet switches and OF switches. For small sites, IXPBuilder eliminates the requirement for Ethernet switches, rendering the solution very affordable for smaller cities and towns in developing countries.

7.2.3 The centralised management of remote IXPs

Having developed stand-alone IXP models the question; *how can remote IXPs be centrally managed without interfering with their independence from a peering perspective?* was answered with the addition of dIXP functionality. This functionality facilitates the day-to-day management of remote mIXP from a cIXP as part of a dIXP, over a secure connection while preserving the peering

independence of each IXP within the set. It is can be concluded therefore that the IXP administrator can securely monitor and manage the day-to-day operation of remote sites, over the Internet or via a dedicated management link, while peering at each IXP site remains independent.

7.2.4 The development of SDX models

How and what are the potential benefits of the incorporation of SDN into IXP models to create an SDX? was established with the incorporation of SDN functionality into IXPBuilder. This revealed three additional models that facilitate the management of OF based Ethernet switches from an SDN Controller (*SC*) or in the case of a small SDX, the management of the internal OF switch. Adding SDN functionality also exposes a new REpresentational State Transfer (REST) Application Programming Interface (API) that exposed the flow tables of both the internal and external OF Ethernet switches to manipulation. This presents opportunities for new application development that can further the functionally of SDXs in the future. The open source nature of the *SC* means the SDN models are also relatively affordable within the constraints of the pricing of OF Ethernet switches. For small IXP sites, IXPBuilder does not require external OF Ethernet switches and therefore the SDX can match the IXP in terms of affordability for smaller hub towns and cities in developing countries.

7.3 Recommendations

7.3.1 Political

In order to address the challenge of migration to capital cities it is necessary to build a spatial strategy that develops alternative economic hub towns and cities. Technology industries are flexible and can be established quickly once the underlying infrastructure is in place. Adequate and robust electrical power, high-speed fibre-optic connections linking these hubs to the submarine fibre-optic landing stations by the most direct route as well as other services such as water, sanitation, road networks and adequate housing are required.

The region also requires Internet Data Centres (IDC). Currently, in Uganda for example, the first carrier-neutral IDC is in the build phase. Investors should be encouraged to fund the building

of these key infrastructure nodes within the new hubs and provide stimuli through specialised enterprise organisations, at local level, established and measured to attract and provide the environment necessary for technology businesses to succeed. IDCs provide the platform for technology business to host their server hardware. This attracts software businesses to host their platforms on the infrastructure of hardware businesses which in turn serves to attract ISPs looking for transit business. This makes the IDC the ideal location to place the local IXP and together these businesses benefit by the provision of the exchange point to keep *local traffic local* which improves each of their service offerings.

Inter-governmental organisations such as the East African Communications Organisation (EACO) and the East African Science and Technology Commission (EASTECO) have an important regional role and are well placed to advise governments and regulators on the benefits of the harmonisation of Internet infrastructure standards in order to create a common approach to technology hubs which can support economic growth across the region. A regional strategy, implemented well, has the potential to positively develop all the regional economies, where specialisations in one country can complement specialisations in the others making the overall region very attractive to investment.

7.3.2 Productising the IXP models

The PoC system design demonstrates two sets of cost effective, affordable, models for IXPs to cater for many scenarios. These models offer simplicity through automation, affordability through containerisation and astute hardware selection. Adaptability and exchange scalability are characteristics of the models as they can be employed to deliver small remote mIXPs as well as larger centralised cIXP. The models can scale to develop IXPs using either *traditional* managed Ethernet switches or SDXs using OF switches.

Through the PoC builds the software was developed with the aim of evaluating many options and therefore was made highly configurable over the phases of development. Some of these configuration options can be converged into a future production software once a balance is struck

between functionality and simplicity of operation. For example, the host and switch build steps could be converged while the server build as well as software install and configure steps could also be converged into a single step without restraining the functionality that much. Incorporation of these trade-offs will be necessary to the functional specification of the product. The PoC is not country specific and this system can easily be deployed in any similar scenario elsewhere. Having a locally deployed, centrally managed IXP system that is affordable is a desirable solution in any hub town or city.

7.3.2.1 Licensing

During the development of the PoC, care was taken to use open source software to maintain the open nature and cost effectiveness of the software. The overall PoC is licensed under the European Union Public Licence version 1.2 (EUPLv1.2) which is fully compatible with GNU is Not Unix (GNU) General Public License version 3 (GPLv3). Compatibility with other open source licenses was part of the consideration in the design of EUPLv1.2 so it is therefore ideal as an umbrella license to cover works that include multiple open source licenses. The following licenses apply to the major software elements of IXPBuilder;

- ***Ubuntu 18.04***: GPLv3,
- ***BIND9***: Mozilla Public License (MPL 2.0),
- ***BIRD***: GPLv3,
- ***Ryu***: Apache License 2.0,
- ***LXD***: Apache License 2.0,
- ***LXC***: GNU Lesser General Public License (LGPL) v2.1+.

Care should be taken moving IXPBuilder from a PoC to a production solution that the open nature of the work is maintained.

7.3.2.2 Technologies and Models

For any production solution, a decision will be required as to whether it is necessary to

support both *traditional* and *software defined* models. While *software defined* models have the potential to offer new programmable software abstractions through the exposure of a REpresentational State Transfer (REST) Application Programming Interface (API) micro services for the BIRD daemons and the SDN Controller (*SC*), there has been a reluctance by many of the *traditional* hardware vendors to develop OF compatible hardware. Tested compatible OF hardware are listed by the ONF (ONF, no date). Perhaps the promise of the early work from the ONF Stratum project (O'Connor, 2018) will provide affordable white-box switches in the future that can be leveraged to get the most from SDN at an SDX. A major advantage with models using *traditional* Ethernet switches is the readily available access to relatively cheap hardware, particularly in the second hand market, which can be attractive to IXPs in developing countries. A trade-off to be considered is the potential for new services versus overall system affordability.

7.3.2.3 Redundancy

LinuX Containers (LXC) on the LinuX Container hypervisor Daemon (LXD) are used to isolate the IXP functions in this PoC. For larger IXPs, where there are a large number of peering members and server hardware is readily available, a production version of IXPBuilder may consider the incorporation of the LXD clustering feature. With this feature many LXD instances can share the same distributed database and can be managed uniformly using the LXC client or via a RESTful API (Iatrou, 2018), (Canonical, no date).

7.3.2.4 Security

IXPBuilder is built on Ubuntu 18.04 Long Term Support (LTS) and security of the main IXP *user* account is essential as it is used during install and has root access to each LXC via the LXC client from the shell of the core server. For this reason LXC container Operating Systems (OS) are not remote access enabled. On the management LAN there are a number of RESTful APIs exposed as NBIs. There are Birdseye RESTful APIs for monitoring the BIRD daemons (BIRD) on the Route Server (*RS*), Route Collector (*CS*) and AS112 Blackhole Server (*BS*) in both IXP and SDX modes as well as a RESTful API for managing the *SC* on SDX modes. Access to the RESTful API on the

SC could facilitate unwanted access to disrupt the SDX and for this reason the *SC* was not given an interface on the peering LAN. In any future production software it is recommended that further security of these micro-services should be considered using Transport Layer Security (TLS) version 1.2 or better, over Hyper Text Transfer Protocol Secure (HTTPS). For small sites in particular the inbuilt GNU/Linux Netfilter, firewall, NAT and packet mangler for Linux could be employed (Netfilter, no date) and for larger sites the implementation of a pFsense firewall would be in keeping with the open source nature of the PoC (pFsense, no date).

7.3.2.5 Privacy

The PoC in its current form does not store or share Personally Identifiable Information (PII). Data stored relates to the IXP itself and to the peering information of the member organisations. It is likely that in any future production software, contact information for individuals or other PII could be stored to make the job of IXP administrator easier. In this case it is essential that careful consideration is given as to what information is actually required for the efficient running of the IXP as well as policies which should be developed to support the removal of unnecessary records after a set time period, in order to keep the IXP in compliance with data privacy laws. An example from Uganda is the Data Protection and Privacy Act, 2019 (DPP Act, 2019), (Greenleaf, 2019).

7.3.2.6 User interface

While the Command Line Interface (CLI) method of access to the functionality is perfectly adequate in the PoC, a visual front end, perhaps browser driven or in the form of a mobile application could also be considered. This would facilitate management of a production version from an external computer, tablet or smartphone and would remove the requirement for shell access to the core server. This could be used to further reduce the skill-set requirement by the IXP engineer and administrator while also providing an opportunity to improve the overall security of the solution.

7.4 Future work

This research built the *core tier* of an IXP, containerising the functions, managing the

members, interfacing with *traditional* Ethernet switches, as well as, managing OF switches with an *SC* through the control plane API offered by the OF protocol, when in SDX mode. The next evolution of SDN will explore the programmability of the data plane.

Over the last three years there has been a push to extend SDN beyond the control plane and look at ways to program the switch hardware itself. Switches, even OF ones, use rigid switching Application Specific Integrated Circuit (ASIC) based chip hardware and research is ongoing to develop new Protocol Independent Switch Architecture (PISA) chips, such that, it becomes possible to program the data plane processing directly and remove the reliance on vendor ASICs (Cascone, 2018). Currently OF permits limited flexibility over the control plane as the switch vendors define which headers they support on their ASIC. OF actually gives the *SC* a means to populate the ASIC's fixed tables with flows based on these fixed header types. Future PISA based hardware will permit the direct programming of the switch using languages such as Programming Protocol-independent Packet Processors (P4) (P4, 2018). Networking languages such as P4 is the future direction for SDN as it offers a new level of flexibility through data plane programmability.

Future research in this area can further explore how future PISA based switches can be incorporated into an SDX utilising P4 and the P4Runtime API to control data plane elements within the SDX *switching tier*. Software projects such as the Programmable, Protocol-Independent Software Switch (PISCES), a P4 based switch derived from Open virtual Switch (OvS) offers an entry point for such future research (Shahbaz *et al.*, 2016). The ONF are working on an implementation of P4 called Stratum (O'Connor, 2018) which is currently in the incubation phase, source code and documentation are currently only available to Stratum members (as of September 2019). The aim of Stratum is to avoid the vendor lock-in that exists today on switching hardware via proprietary ASIC interfaces and closed software APIs. Instead it is hoped that Stratum will deliver PISA based white-box switch solutions as the next step in programmable networks and offer an avenue for future research.

7.5 Summary

This research concludes with a reassertion that in order to strengthen the Internet ecosystem as a key component of spatial planning in East Africa and other developing countries, it is necessary to supplement the national Internet eXchange Point (IXP) with additional IXPs within hub towns and cities. From the research question, *what are the political, social, economic and technical drivers that have influenced Internet development in East Africa over the last decade?* it was determined that the landing of the submarine fibre-optic cables on the East African coast was the key trigger driver to development. CDNs located locally at IXPs and ISPs are also key drivers and IXPs need to evolve beyond national capitals to support hub towns and cities. A spatial strategy will need to incorporate this development in order to maintain rural communities. New technologies as well as the potential risk of the emergence of a monopolies create an uncertain future, particularly if multinational monopolies gain dominance over user behaviour and control of content. The IXP footprint can play a major role in the success of local content development and distribution, the lack of local content is seen a major barrier to Internet penetration.

Models of IXP can be developed to cater for local IXPs in the context of developing countries through the development of the PoC in this research. The PoC was developed through testbed experiments and software development to support a number of models leaveraging open source software and containerisation technology to ensure solutions are affordable. The PoC has 11 new models and sub-models which demonstrate scalability and incorporation of existing switching technology.

IXPs can be centrally managed without interfering with their independence from a peering perspective through the inclusion of dIXP functionality which facilitates the day-to-day management of remote mIXP from a cIXP as part of a dIXP while preserving the peering independence of each IXP within the set.

SDX models in the PoC demonstrate that it is possible to incorporate *incorporation of SDN into IXP models to create an SDX* and that *the potential benefits* are demonstrated through an exposed RESTful API that allows the flow tables of OF switches to be manipulated.

A number of recommendations were made. From a political and social perspective to address the challenge of migration to capital cities it is necessary to build a spatial strategy that develops alternative economic hub towns and cities. From an Internet ecosystem perspective, apart from fibre connectivity, IDCs and IXPs are key foundation elements for the establishment of technology industries. Inter-governmental organisations such as the EACO and the EASTECO have an important regional role and are well placed to advise governments and regulators on a common approach to technology hubs which can support economic growth across the region.

Some considerations when productising the IXP models in the PoC are the maintenance of the open source licensing structure, whether there is a need for both IXP and SDX models, the incorporation of redundancy for containers, security and privacy issues as well as user interface options to access the IXP library module classes.

The work being carried out in the Stratum project under the auspices of the ONF offers a glimpse into the future of networking development when the hardware becomes programmable and protocol independent. Future work on IXP development can potentially leverage PISA hardware and network programming languages like P4 to create new functionality and flexibility. Today an variant of OvS called PISCES offers a platform for experimentation.

8. Bibliography

A4AI (2018) *A4AI, 2018 Affordability Report*. Alliance for Affordable Internet (A4AI). Available at: https://a4ai.org/just-released-2018-affordability-report/ (Accessed: 14 May 2019).

Abley, J. and Sotomayor, W. (2015) 'RFC7534. AS112 Nameserver Operations'. Internet Engineering Task Force. Available at: https://tools.ietf.org/pdf/rfc7534.pdf (Accessed: 8 March 2018).

Ahmad, M. Z. and Guha, R. (2012) 'A tale of nine internet exchange points: Studying path latencies through major regional ixps', in *37th Annual IEEE Conference on Local Computer Networks*. IEEE, pp. 618–625.

Alimi, R. *et al.* (2014) *RFC7285. Application-layer traffic optimization (ALTO) protocol*. Available at: https://tools.ietf.org/pdf/rfc7285.pdf (Accessed: 11 November 2018).

AMS-IX (2013) *AMS-IX Starts New Regional Internet Exchange Hub in Mombasa, Kenya, in Collaboration with KIXP, AMS-IX*. Available at: https://ams-ix.net/newsitems/101 (Accessed: 29 December 2018).

ANACOM (no date) 'The Submarine Cable in a Sea of Connectivity'. Autoridade Nacional de Comunicações (ANACOM).

Antichi, G. *et al.* (2017) 'ENDEAVOUR: A scalable SDN architecture for real-world IXPs', *IEEE Journal on Selected Areas in Communications*, 35(11), pp. 2553–2562.

Asiimwe, F. (2017) *Corporate Governance and Performance of SMEs in Uganda*. ISSN 2518-8623. International Journal of Technology and Management.

Asteroid (no date) *Mombasa Campaign*. Available at: https://www.asteroidhq.com/campaigns/2 (Accessed: 5 July 2019).

AWS (no date) *AWS Global Infrastructure, Amazon Web Services*. Available at: https://aws.amazon.com/about-aws/global-infrastructure/ (Accessed: 22 May 2019).

Bakker, N. *et al.* (2016) *RFC7947. Internet Exchange BGP Route Server*. Available at: https://tools.ietf.org/pdf/rfc7947.pdf (Accessed: 16 January 2017).

BIRD (no date) *The BIRD Internet Routing Daemon Project*. Available at: http://bird.network.cz/ (Accessed: 17 January 2017).

Bosshart, P. *et al.* (2014) 'P4: Programming protocol-independent packet processors', *ACM SIGCOMM Computer Communication Review*, 44(3), pp. 87–95.

Boyd, C. O. (1993) 'Combining qualitative and quantitative approaches.', *NLN publications*, (19–2535), pp. 454–475.

Brady, T. (ed.) (2017) 'RFC7159. The JavaScript Object Notation (JSON) Data Interchange Format'. Internet Engineering Task Force. Available at: https://tools.ietf.org/pdf/rfc7159.pdf (Accessed: 23 April 2019).

Brito, S. H. B. *et al.* (2016) 'An Analysis of the Largest National Ecosystem of Public Internet eXchange Points: The Case of Brazil', *Journal of Communication and Information Systems*, 31(1).

Bruyere, M. *et al.* (2018) 'Rethinking IXPs' Architecture in the Age of SDN', *IEEE Journal on Selected Areas in Communications*, 36(12), pp. 2667–2674.

Bulega, T. *et al.* (2011) 'Uganda's National Transmission Backbone Infrastructure Project: Technical Challenges

and the Way Forward', *Fiber and Integrated Optics*, 30(5), pp. 282–295.

Canonical (2015) *LXD crushes KVM in density and speed*. Available at: https://insights.ubuntu.com/2015/05/18/lxd-crushes-kvm-in-density-and-speed/ (Accessed: 29 November 3017).

Canonical (no date) *LXD Clustering*, *LXD Read the docs*. Available at: https://lxd.readthedocs.io/en/latest/clustering/ (Accessed: 5 May 2019).

Cascone, C. (2018) 'P4 and P4Runtime Technical Introduction and Use Cases for Service Providers'. *Open Networking Summit 2018*, 27 September. Available at: https://events.linuxfoundation.org/wp-content/uploads/2017/12/Tutorial-P4-and-P4Runtime-Technical-Introduction-and-Use-Cases-for-Service-Providers-Carmelo-Cascone-Open-Networking-Foundation.pdf (Accessed: 6 July 2019).

Chatzis, N., Smaragdakis, G., *et al.* (2013) 'On the benefits of using a large IXP as an Internet vantage point', in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, pp. 333–346.

Chatzis, N., Smaradgakis, G., *et al.* (2013) 'There is More to IXPs than Meets the Eye.', in *Computer Communications Review*. ACM SIGCOMM Computer Communication Review. doi: 10.1145/2541468.2541473.

Chatzis, N. *et al.* (2015) 'Quo vadis Open-IX?', *ACM SIGCOMM Computer Communication Review*, 45(1), pp. 12–18.

Chavula, J. *et al.* (2014) 'Quantifying the effects of circuitous routes on the latency of intra-Africa internet traffic: a study of research and education networks', in *International Conference on e-Infrastructure and e-Services for Developing Countries*. Springer, pp. 64–73.

Chiesa, M. *et al.* (2016) 'Inter-domain networking innovation on steroids: empowering ixps with SDN capabilities', *IEEE Communications Magazine*, 54(10), pp. 102–108.

Cisco (2019) 'Cisco Visual Networking Index: Forecast and Trends, 2017–2022'. Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html (Accessed: 2 March 2019).

Cloudflare (2015) *Mombasa, Kenya: CloudFlare's 43rd data center*, *Cloudflare blog*. Available at: https://blog.cloudflare.com/mombasa-kenya-cloudflares-43rd-data-center/ (Accessed: 30 October 2018).

Corning (2014) 'Corning SMF-28 Ultra Optical Fibre'. Corning. Available at: https://www.corning.com/media/worldwide/coc/documents/Fiber/SMF-28%20Ultra.pdf (Accessed: 19 January 2019).

Daily Monitor (2019) 'Press Statement on the UCC directing it to suspend the online newspaper', 7 February.

DE-CIX (no date) *GlobePEER - DE-CIX*, *DE-CIX Website*. Available at: https://de-cix.net/en/services/globepeer (Accessed: 11 May 2017).

DeNardis, L. (2012) 'Governance at the Internet's Core: The Geopolitics of Interconnection and IXPs in Emerging Markets', in. *40th Research Conference on Communication, Information and Internet Policy, 2012*, Telecommunications Policy Research Conference (TPRC).

Di Lallo, R. (2015) *Is It Really Worth Peering at IXPs? A Comparative Study*. Regional Internet Registry for Europe (RIPE) Labs. Available at: https://labs.ripe.net/Members/roberto_di_lallo/is-it-really-worth-peering-at-ixps (Accessed: 25 February 2016).

Dibley, L. (2011) 'Analysing narrative data using McCormack's Lenses.', *Nurse Researcher*, 18(3).

DPP Act (2019) *Data Protection and Privacy Act, 2019*. Available at: https://ulii.org/ug/legislation/act/2019/1 (Accessed: 2 April 2019).

EAC (no date) *East African Community (EAC)*, *East African Community (EAC)*. Available at: https://www.eac.int (Accessed: 24 June 2019).

EACO (2014) 'Report of 3rd meeting of Working Group 5 (WG5) - (IP Networks, Standards and Cybersecurity)'. EACO. Available at: http://www.eaco.int/admin/docs/reports/3rd_Meeting_WG5_%20IP_Networks_Standards_and_Cybersecurity_Report_October_2014.pdf (Accessed: 20 December 2018).

EACO (2017) *Peering and Interconnectivity in East Africa*. East Africa Communications Organisation.

EADC (no date) *The East Africa Data Centre (EADC)*, *The East Africa Data Centre (EADC)*. Available at: https://eastafricadatacentre.com/ (Accessed: 12 March 2018).

Eccles, B. (2017) 'Total Societal Impact Is the Key To Improving Total Shareholder Return', *Forbes*, 25 October. Available at: https://www.forbes.com/sites/bobeccles/2017/10/25/total-societal-impact-is-the-key-to-improving-total-shareholder-return/#726829f42113 (Accessed: 18 December 2018).

Ecobank (2018) *The high cost of mobile data in Sub-Saharan Africa*. Research. Ecobank Research. Available at: https://www.ecobank.com/upload/publication/20180910054643018QJEBKEVZKD/20180910054635730h.pdf (Accessed: 20 October 2018).

Edquist, H. *et al.* (2018) 'How Important are Mobile Broadband Networks for Global Economic Development?', in. *International Association for Research in Income and Wealth*, Copenhagen, Denmark. Available at: http://www.iariw.org/copenhagen/edquist.pdf (Accessed: 23 December 2018).

Edwards, R. and Holland, J. (2013) *What is Qualitative Interviewing?* Bloomsbury Publishing. Available at: https://books.google.co.ug/books?id=GdCOAQAAQBAJ.

Esselaar, S. and Adam, L. (2014) 'Understanding what is happening in ICT in Tanzania - A supply- and demand-side analysis of the ICT sector'. Research ICT Africa. Available at: https://www.researchictafrica.net/publications/Evidence_for_ICT_Policy_Action/Policy_Paper_11_-_Understanding_what_is_happening_in_ICT_in_Tanzania.pdf (Accessed: 12 May 2017).

ETSI (no date) *Network Functions Virtualisation*. Available at: http://www.etsi.org/technologies-clusters/technologies/nfv (Accessed: 1 July 2019).

EU (2018) *Directive establishing the European Electronic Communications Code*. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018L1972&from=en (Accessed: 16 December 2018).

Fanou, R. *et al.* (2017) 'Reshaping the African Internet: From scattered islands to a connected continent', *Computer Communications*, 113, pp. 25–42.

Fanou, R., Francois, P. and Aben, E. (2015) 'On The Diversity of Interdomain Routing in Africa', in. *International Conference on Passive and Active Network Measurement*, IMDEA Networks Institute, pp. 41–54. doi: 10.1007/978-3-319-15509-8_4.

Fielding, R. T. (2000) *Architectural Styles and the Design of Network-based Software Architectures: Chapter 5: Representational state transfer (rest)*. University of California. Available at: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (Accessed: 10 December 2018).

Filip, O. *et al.* (no date) 'BIRD Programmer's Documentation'. Available at: http://bird.network.cz/?get_doc&f=prog.html (Accessed: 9 October 2017).

Flick, U. (2014) *An Introduction to Qualitative Research*. SAGE Publications. Available at: https://books.google.co.ug/books?id=HB-VAgAAQBAJ.

Formoso, A. *et al*. (2018) 'Deep Diving into Africa's Inter-Country Latencies', in. *IEEE International Conference on Computer Communications (INFOCOM)*, Honolulu, Hawaii: IEEE.

Freedom House (2016) *Freedom on the Net 2016 - Uganda*. Internet Freedom report. Freedom House. Available at: https://freedomhouse.org/report/freedom-net/2016/uganda (Accessed: 2 February 2016).

Freedom House (no date) *Freedom on the Net 2018 - Uganda*. Internet Freedom report. Freedom House. Available at: https://freedomhouse.org/report/freedom-net/2018/uganda (Accessed: 2 February 2016).

de Freytas-Tamura, K. (2017) 'Kenya Supreme Court Nullifies Presidential Election', *The New York Times*, 1 September. Available at: https://www.nytimes.com/2017/09/01/world/africa/kenya-election-kenyatta-odinga.html (Accessed: 6 February 2019).

Fusch, P. I. and Ness, L. R. (2015) 'Are we there yet? Data saturation in qualitative research', *The qualitative report*, 20(9), pp. 1408–1416.

Gill, P. *et al*. (2008) 'The flattening internet topology: Natural evolution, unsightly barnacles or contrived collapse?', in *International Conference on Passive and Active Network Measurement*. Springer, pp. 1–10.

Google (no date) *Google infrastructure, Peering Google*. Available at: https://peering.google.com/#/infrastructure (Accessed: 30 October 2018).

Gorey, C. (2016) *New INEX internet exchange to open in Cork to boost region, Silicon Republic*. Available at: https://www.siliconrepublic.com/comms/inex-cork-holyhill (Accessed: 12 July 2017).

Government of Rwanda (2013) 'National Broadband Policy for Rwanda'. Government of Rwanda. Available at: http://197.243.19.11/mitec/fileadmin/Documents/Policies_and_Rugulations/ICT_Polices/National_Broadband_Policy.pdf.

Government of Tanzania (2003) 'National Information and Communications Technologies Policy'. Tanzania Ministry of Communications and Transport. Available at: http://www.tzonline.org/pdf/ictpolicy2003.pdf (Accessed: 2 December 2017).

Government of Tanzania (2011) *The Electronic and Postal Communications Act (Access, Co-location and Infrastructure sharing) Regulations, United Republic of Tanzania*. Available at: https://www.tcra.go.tz/images/documents/regulations/accessCo-locationInfrsSharing.pdf (Accessed: 11 May 2017).

Government of Uganda (2015) *National Development Plan (NDPII) 2015/16 – 2019/20*. National Development Plan II. Kampala, Uganda: The Republic of Uganda, pp. 124–125. Available at: http://library.health.go.ug/download/file/fid/580764 (Accessed: 12 July 2017).

Government of Uganda (2018) 'Uganda National Broadband Policy'. Government of Uganda.

Grant Gross (2018) *Internet Freedom Declines Again, with 'Polarized Echo Chambers' Aiding Censorship Efforts*. Blog. Internet Society. Available at: https://www.internetsociety.org/blog/2018/11/internet-freedom-declines-again-with-polarized-echo-chambers-aiding-censorship-efforts/?gclid=EAIaIQobChMIorDog42t4AIV0OF3Ch1VhgdKEAAYASAAEgIMl_D_BwE (Accessed: 8 February 2019).

Greenleaf, G. (2019) *Global Tables of Data Privacy Laws and Bills*. 6th Ed January 2019. SSRN. Available at: https://ssrn.com/abstract=3380794.

GSMA (2018a) *Delivering the Digital Revolution: Will mobile infrastructure keep up with rising demand?* London: GSMA. Available at: https://www.gsma.com/publicpolicy/wp-content/uploads/2018/02/GSMA_DigitalTransformation_Delivering-the-Digital-Revolution.pdf (Accessed: 20 November 2018).

GSMA (2018b) *The mobile economy, Sub-Saharan Africa, 2018*. London: GSMA. Available at: https://www.gsmaintelligence.com/research/?file=809c442550e5487f3b1d025fdc70e23b&download (Accessed: 23 November 2018).

Gupta, A. *et al.* (2013) 'Peering at the Internet's Frontier: A First Look at ISP Interconnectivity in Africa', in. *International Conference on Passive and Active Network Measurement (PAM 2014)*, Los Angeles, CA, USA: Springer, pp. 204–213. doi: 10.1007/978-3-319-04918-2.

Gupta, A. *et al.* (2014) 'SDX: A Software Defined Internet Exchange', in *Special Interest Group on Data Communication (SIGCOMM) 2014*. *SIGCOMM-2014*, Chicago, Illinois, USA: Association for Computing Machinery (ACM), pp. 551–562. doi: 10.1145/2619239.2626300.

Gupta, A. *et al.* (2016) 'An Industrial-Scale Software Defined Internet Exchange Point.', in *NSDI. 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16)*, Santa Clara, CA, USA: USENIX.

Haleplidis, E. *et al.* (2015) 'RFC7426. Software-defined networking (SDN): Layers and architecture terminology'. Available at: https://tools.ietf.org/pdf/rfc7426.pdf (Accessed: 22 February 2015).

Hares, S., Rekhter, Y. and Li, T. (2006) 'RFC4271. A Border Gateway Protocol 4 (BGP-4)'. Available at: https://tools.ietf.org/pdf/rfc4271.pdf (Accessed: 9 October 2017).

Houghton, C. *et al.* (2013) 'Rigour in qualitative case-study research.', *Nurse researcher*, 20(4).

Hoyle, R. H. (2018) *Improving Health Research on Small Populations: Proceedings of a Workshop*. National Academies Press.

Huston, G. (2015) 'Happy Eyeballs for the DNS'. *DNS Operations, Analysis and Research Centre, APNIC Labs, fall workshop*, 6 October. Available at: https://labs.apnic.net/presentations/store/2015-10-04-dns-dual-stack.pdf (Accessed: 12 March 2016).

Huurdeman, A. A. (2003) *The Worldwide History of Telecommunications*. Wiley (A Wiley-interscience publication). Available at: https://books.google.co.ug/books?id=SnjGRDVIUL4C.

Iatrou, M. (2018) 'LXD Clusters: A Primer'. Canonical Limited. Available at: https://ubuntu.com/blog/lxd-clusters-a-primer (Accessed: 20 June 2019).

icolo.io (no date) *icolo.io, Mombasa One (MBA1), icolo.io*. Available at: https://www.icolo.io (Accessed: 12 March 2018).

ITU (2017) 'ICT Development Index (IDI) 2017'. International Telecommunication Union. Available at: https://www.itu.int/net4/ITU-D/idi/2017/index.html (Accessed: 14 May 2019).

ITU-T (2018) *Measuring the Information Society Report - Volume 1. 2018*. ICT Development Index 2017 (IDI). Geneva, Switzerland: Internetional Telecommunications Union (ITU). Available at: https://www.itu.int/net4/ITU-D/idi/2017/index.html (Accessed: 10 May 2019).

IXP Manager (no date). Available at: https://www.ixpmanager.org/ (Accessed: 30 November 2017).

Jagun, A. (2008) 'The Case for "Open Access" Communications Infrastructure in Africa: The SAT-3/WASC cable– A briefing', in. Association for Progressive Communications. Available at:

https://www.apc.org/sites/default/files/APC_SAT3Briefing_20080515.pdf.

Jasinska, E. *et al.* (2017) 'RFC7947. Internet Exchange BGP Route Server'. Internet Engineering Task Force. Available at: https://tools.ietf.org/pdf/rfc7947.pdf (Accessed: 17 March 2017).

Jumbe, I. (2016) 'Mombasa tipped to be region's internet hub after IXP launch', *Saturday Standard (Kenya)*, 23 June. Available at: https://www.standardmedia.co.ke/article/2000206339/mombasa-tipped-to-be-region-s-internet-hub-after-ixp-launch (Accessed: 29 December 2018).

Kafeero, S. (2018) 'Why the flip-flop on Mobile Money tax', *Daily Monitor*, 28 October. Available at: https://www.monitor.co.ug/SpecialReports/Mobile-Money-tax--MTN-Uganda-Bank-of-Uganda-Parliament/688342-4825414-s3rgje/index.html (Accessed: 30 January 2019).

Kasemire, C. (2019) 'OTT volumes fall to Shs4b in December URA data indicates', *Daily Monitor*, 31 January. Available at: https://www.monitor.co.ug/Business/Markets/OTT-volumes-fall-Shs4b-December-URA-data-UCC-/688606-4958766-8jyqcz/index.html (Accessed: 31 January 2019).

Kende, M. and Rose, K. (2015) 'Promoting Local Content Hosting to Develop the Internet Ecosystem'. Internet Society.

Kini, K. *et al.* (2014) 'Increasing Internet Connectivity through the Development of Local Networks'. International Telecommunications Union.

Krejcie, R. V. and Morgan, D. W. (1970) 'Determining sample size for research activities', *Educational and psychological measurement*, 30(3), pp. 607–610.

Lapeyrade, R., Bruyère, M. and Owezarski, P. (2016) 'OpenFlow-based Migration and Management of the TouIX IXP', in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, pp. 1131–1136.

Leininger, M. (1994) 'Evaluation criteria and critique of qualitative research studies', *Critical issues in qualitative research methods*, 95, p. 115.

Lien, S.-Y. *et al.* (2017) '5G new radio: Waveform, frame structure, multiple access, and initial access', *IEEE communications magazine*, 55(6), pp. 64–71. doi: 10.1109/MCOM.2017.1601107.

LINX (2014) *LINX NoVA Goes Live*, *LINX*. Available at: https://www.linx.net/communications/press-releases/linx-nova-goes-live (Accessed: 11 May 2017).

LINX (2017) 'IXScotland expansion'. Available at: https://www.linx.net/ixscotland-expansion/ (Accessed: 2 June 2019).

LINX (no date) *LINX Network Topology*, *LINX*. Available at: https://www.linx.net/tech-info-help/network-topology (Accessed: 11 May 2017).

Liquid Telecom (2019) 'Liquid Telecom map'. East Africa: Liquid Telecom. Available at: https://www.liquidtelecom.com/about-us/network-map.html (Accessed: 5 July 2019).

Lodhi, A. *et al.* (2014) 'Using PeeringDB to Understand the Peering Ecosystem', in. Available at: http://www.sigcomm.org/sites/default/files/ccr/papers/2014/April/0000000-0000002.pdf (Accessed: 8 November 2018).

LRG (2019) '31% of Adults Watch Video via a Connected TV Device Daily'. Leichtman Research Group. Available at: https://www.leichtmanresearch.com/wp-content/uploads/2019/05/LRG-Press-Release-05-31-2019.pdf (Accessed: 2 July 2019).

Mambretti, J., Chen, J. and Yeh, F. (2014a) 'Software-Defined Network Exchanges (SDXs) and Infrastructure (SDI): Emerging innovations in SDN and SDI interdomain multi-layer services and capabilities', in *2014 International Science and Technology Conference (Modern Networking Technologies)(MoNeTeC)*. IEEE, pp. 1–6.

Mambretti, J., Chen, J. and Yeh, F. (2014b) 'Software-defined network exchanges (sdxs): Architecture, services, capabilities, and foundation technologies', in *2014 26th International Teletraffic Congress (ITC)*. IEEE, pp. 1–6.

Mang'unyi, E. (2015) *Kenya's second Internet Exchange Point (IXP) Closed*, *Tech Savvy*. Available at: http://techsavvy.or.ke/kenyas-second-internet-exchange-point-ixp-closed/#sthash.UOKwOT0J.dpbs (Accessed: 29 December 2018).

McCann, M. (2018) 'Teraco: An African story. The journey of a vendor neutral data centre'. *AfPIF 2018*, Capetown, South Africa, 2 August. Available at: https://www.afpif.org/wp-content/uploads/2018/08/02-Teraco_Overview_2018-MichelleMcann-Teraco.pdf (Accessed: 2 August 2018).

McKeown, N. *et al.* (2008) 'OpenFlow: enabling innovation in campus networks', *SIGCOMM Comput. Commun. Rev.*, 38(2), pp. 69–74.

Mendelsohn, N. (2016) 'Metcalfe's Law : Why is the Web so Big?' Tufts University, 26 January. Available at: https://www.cs.tufts.edu/comp/150IDS/slides/Metcalfe.pdf (Accessed: 4 May 2019).

Mockapetris, P. (1987) 'RFC1034. Domain Names - Concepts and Facilities'. Internet Engineering Task Force. Available at: https://tools.ietf.org/pdf/rfc1034.pdf (Accessed: 17 March 2017).

Mugabe, D. (2009) 'Uganda gets fibre optic cable', *New Vision*, 25 July. Available at: https://www.newvision.co.ug/new_vision/news/1239778/uganda-fibre-optic-cable (Accessed: 20 May 2017).

Muller, R. (2018) 'EASSy enters commercial service', *My Broadband*, 8 May. Available at: https://mybroadband.co.za/news/telecoms/14278-EASSy-enters-commercial-service.html (Accessed: 18 October 2018).

Nanfuka, J. (2019) *Social Media Tax Cuts Ugandan Internet Users by Five Million, Penetration Down From 47% to 35%*. Collaboration on International ICT Policy in East and Southern Africa (CIPESA). Available at: https://cipesa.org/2019/01/%EF%BB%BFsocial-media-tax-cuts-ugandan-internet-users-by-five-million-penetration-down-from-47-to-35/ (Accessed: 2 February 2019).

NAPAfrica (no date) *NAPAfrica*. Available at: https://www.napafrica.net/ (Accessed: 2 July 2019).

NCIP (2016) *Northern corridor integration projects (NCIP): Regional Broadband Strategy*. Regional Broadband Strategy. East African Community. Available at: http://www.ict.go.ug/wp-content/uploads/2018/06/Regional-Broadband-Strategy_Signed.pdf (Accessed: 13 April 2017).

Netfilter (no date) *Netfilter, firewall, NAT and packet mangling for Linux*, *Netfilter*. Available at: https://netfilter.org/ (Accessed: 12 March 2016).

netnod (no date) *Background | Netnod*, *netnod*. Available at: https://www.netnod.se/about-netnod/netnod-history (Accessed: 10 September 2018).

NOFBI (no date) *National Optic Fibre Backbone (NOFBI)*, *National Optic Fibre Backbone (NOFBI)*. Available at: http://icta.go.ke/national-optic-fibre-backbone-nofbi/ (Accessed: 18 October 2018).

Norton, W. B. (2014) *The Internet Peering Playbook: Connecting to the Core of the Internet*. DrPeering Press. Available at: https://www.amazon.com/gp/product/1937451119/ref=dbs_a_def_rwt_hsch_vamf_taft_p1_i0.

Nunamaker Jr, J. F., Chen, M. and Purdin, T. D. (1990) 'Systems development in information systems research',

Journal of management information systems, 7(3), pp. 89–106.

Nyabola, N. (2018) *Digital Democracy, Analogue Politics: How the Internet Era Is Transforming Kenya*. Zed Books. Available at: https://books.google.co.ug/books?id=Zd5xswEACAAJ.

Ó Briain, D. (2018) 'The Access Tier ISP, considering upstream connectivity'. *MilroTik User Meeting (MUM) - East Africa*, Nairobi, Kenya, 30 January. Available at: http://www.obriain.com/mikrotik/20180130-MikroTik-ISP-IXP_Lecture.pdf (Accessed: 30 January 2018).

O'Briain, D. *et al.* (2017) 'Rebuilding the Internet eXchange Point in Uganda', in. *28th Irish Signals and Systems Conference*, Killarney, Ireland: IEEEXplore.

O'Connor, B. (2018) 'Stratum: An Overview'. Open Networking Foundation (ONF). Available at: https://www.opennetworking.org/wp-content/uploads/2018/12/Stratum_-An-Overview.pdf (Accessed: 6 July 2019).

OEDC (2012) 'Innovation for Development'. Organisation for Economic Co-operation and Development (OECD). Available at: https://www.oecd.org/innovation/inno/50586251.pdf (Accessed: 31 October 2018).

Offerman, A. (2016) *BIRD manages routing at world's largest Internet Exchanges (BIRD)*. Open Source Observatory (OSOR). Available at: https://joinup.ec.europa.eu/document/bird-manages-routing-worlds-largest-internet-exchanges-bird (Accessed: 30 April 2016).

ONF (no date) *ONF OpenFlow Conformant: Certified Product List*, *Open Networking Foundation (ONF)*. Available at: https://www.opennetworking.org/product-registry/ (Accessed: 12 March 2016).

Open vSwitch (no date) *OvS Open vSwitch*. Available at: http://openvswitch.org/ (Accessed: 14 June 2015).

OpenStack Foundation (2017) *OpenStack Ocata Strengthens Core Infrastructure Services and Container Integration with 15th Release of Cloud Computing Software*, *OpenStack*. Available at: https://www.openstack.org/news/view/302/openstack-ocata-strengthens-core-infrastructure-services-and-container-integration-with-15th-release-of-cloud-computing-software (Accessed: 27 February 2017).

P4 (2018) 'P4 Language Specification'. The P4 Language Consortium. Available at: https://p4.org/p4-spec/docs/P4-16-v1.1.0-spec.pdf (Accessed: 5 July 2019).

Pazi, S. and Chatwin, C. (2013) 'Assessing the Economic Benefits and Challenges of Tanzania's National ICT Broadband Backbone (NICTBB)', *International Journal of Information and Computer Science 2 Issue 7, November 2013*, 2(7), pp. 117–126. Available at: http://www.seipub.org/ijics/Download.aspx?ID=10501 (Accessed: 11 May 2017).

PeeringDB (no date) *PeeringDB*. Available at: https://www.peeringdb.com (Accessed: 12 August 2018).

Peffers, K. *et al.* (2007) 'A design science research methodology for information systems research', *Journal of management information systems*, 24(3), pp. 45–77.

Petersen, K. *et al.* (no date) 'Action Research as a Model for Industry-Academia Collaboration in the Software Engineering Context'.

pFsense (no date) *pFsense, open source security, pFsense*. Available at: https://www.pfsense.org/ (Accessed: 12 March 2016).

Pitt, D. (2016) *What Is SDNFV & Why Should You Use It?*, *Light Reading*. Available at: http://www.lightreading.com/nfv/nfv-mano/what-is-sdnfv-and-why-should-you-use-it/a/d-id/724090 (Accessed: 10 July 2016).

PWC (2018) *Tax Watch*. PricewaterhouseCoopers. Available at: https://www.pwc.com/ug/en/assets/pdf/ug-tax-watch2018.pdf (Accessed: 1 February 2019).

R Project (no date) *The R Project for Statistical Computing*, *The R Project for Statistical Computing*. Available at: https://www.r-project.org/ (Accessed: 26 July 2017).

Ralph, P. (2014) 'Lab-based action design research', in *Companion Proceedings of the 36th International Conference on Software Engineering*. ACM, pp. 528–531.

Rao, N. (2016) *Bandwidth Costs Around the World*, *CloudFlare blog*. Available at: https://blog.cloudflare.com/bandwidth-costs-around-the-world/ (Accessed: 28 October 2018).

Raxio (2018) 'Uganda gets first in-country state-of-the-art Data Centre at Namanve', *Raxio*, 23 July. Available at: https://raxio.co.ug/wp-content/uploads/2018/07/Raxio-Press-Release-20180723.pdf (Accessed: 25 October 2018).

Rekhter, Y. and Lougheed, K. (1989) 'Border Gateway Protocol (BGP)'. Available at: https://tools.ietf.org/html/rfc1105 (Accessed: 15 July 2016).

RICTA (2017) 'Rwanda Peering Day Meeting, meeting report'. Available at: https://ricta.org.rw/sites/default/files/resources/rwanda_peering_day-report-2_0.pdf (Accessed: 20 December 2018).

Roberts, B. (2018) 'Cape to Cairo and Other African Journeys'. *African Peering & Interconnection Forum (AfPIF)*, Cape Town, South Africa, 21 August. Available at: 01-Cape-to-Cairo-and-Other-African-Journeys-BenRoberts-LiquidTelecom.pdf (Accessed: 21 August 2018).

RURA (2009) 'Guidelines for Rwanda Internet eXchange Point (RINEX) Management'. Rwanda Utilities Regulatory Authority. Available at: https://www.researchictafrica.net/countries/rwanda/Guidelines_for_Rwanda_Internet_Exchange_Point_(RINEX)_Management_2009.pdf (Accessed: 28 December 2018).

RYU (2018) *RYU SDN Framework — Ryu documentation Release 4.30*, *Ryu documentation Release 4.30*. Available at: https://media.readthedocs.org/pdf/ryu/latest/ryu.pdf (Accessed: 29 November 2018).

Santos, P. S. M. dos and Travassos, G. H. (2009) 'Action research use in software engineering: An initial survey', in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, pp. 414–417.

SEACOM (2010) *SEACOM: Continuing its Commitment to East Africa Development*. SubOptic, international undersea communications industry Association. Available at: https://www.suboptic.org/wp-content/uploads/2014/10/246_Poster_PD_01.pdf (Accessed: 20 October 2018).

Sedoyeka, E. and Sicilima, J. (2016) 'Tanzania National Fibre Broadband Backbone: Challenges and Opportunities', *International Journal of Computing and ICT Research*, 10(1), pp. 61–92. Available at: http://ijcir.mak.ac.ug/volume10-issue1/article6.pdf (Accessed: 29 December 2018).

Sein, M. K. *et al.* (2011) 'Action design research', *MIS quarterly*, pp. 37–56.

Shahbaz, A. (2018) *Freedom on the Net 2018: The Rise of Digital Authoritarianism. Fake news, data collection, and the challenge to democracy.* Freedom House. Available at: https://freedomhouse.org/sites/default/files/FOTN_2018_Final%20Booklet_11_1_2018.pdf (Accessed: 8 February 2019).

Shahbaz, M. *et al.* (2016) 'Pisces: A programmable, protocol-independent software switch', in *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, pp. 525–538.

Smolaks, M. (2017) *Tanzania's largest data center is struggling to find customers*, *Data Center Dynamics (DCD)*. Available at: https://www.datacenterdynamics.com/news/tanzanias-largest-data-center-is-struggling-to-find-customers/ (Accessed: 20 December 2018).

Souter, D. and Kerretts-Makau, M. (2012) 'Internet Governance in Kenya - An Assessment for the Internet Society'. Internet Society. Available at: https://www.researchictafrica.net/multistake/Souter_Kerretts-Makau_2012_-_Internet_governance_in_Kenya_-_an_assessment_for_the_Internet_Society.pdf (Accessed: 29 December 2018).

SportsPesa (no date) *SportsPesa*. Available at: https://www.sportpesa.com (Accessed: 20 November 2017).

Ssempebwa, J. and Lubuulwa, M. (2011) 'E-government for development: Implementation challenges of Uganda's national backbone infrastructure project and key lessons', in *IST-Africa Conference Proceedings, 2011*. IEEE, pp. 1–9.

Stallings, W. (2015) *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Pearson Education. Available at: https://books.google.co.ug/books?id=nL_QCgAAQBAJ.

Stringer, J. *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.

TEAMS (no date) *The East African Marine System (TEAMS)*, *The East African Marine System Limited*. Available at: https://www.teams.co.ke/ (Accessed: 22 September 2017).

techjaja (2017) 'The truth about Uganda's National Backbone Infrastructure unravelled: UTL, NITA and MDAs', *techjaja*, 27 September. Available at: https://www.techjaja.com/the-truth-about-ugandas-national-backbone-infrastructure-unraveled-utl-nita-and-mdas/ (Accessed: 30 September 2017).

TeleGeography (2019) 'Submarine cable map'. TeleGeography. Available at: https://www.submarinecablemap.com/ (Accessed: 10 July 2019).

The Open Internet (no date) *A guide to the Open Internet*, *The Open Internet*. Available at: http://www.theopeninter.net/ (Accessed: 14 May 2019).

TIA (2017) 'TIA-942-B: Telecommunications Infrastructure Standard for Data Centers'. Telecommunications Industry Association (TIA).

UBIST (2009) *Uganda Broadband Infrastructure Strategy Team (UBIST) report*. Available at: http://www.ict.go.ug/sites/default/files/Resource/DRAFT%20UBIST%20FINAL%20REPORT%20MARCH%202009.pdf (Accessed: 12 February 2016).

UCC (2019) 'The Uganda Communications Commission Guidelines on Internet Exchange Points'. Uganda Communications Commission (UCC). Available at: https://uixp.co.ug/sites/default/files/documentation/UIXP-UCC-Proposed_IXP_License_Framework-Formal_Feedback-Final.pdf (Accessed: 25 May 2019).

UCC (no date) *Uganda Communications Commission: Market & industry Quarterly Reports*. Industry. Uganda Communications Commission. Available at: https://www.ucc.co.ug/market-industry-quarterly-reports/ (Accessed: 28 January 2018).

U.N. (2015) *Transforming our world: the 2030 Agenda for Sustainable Development*. Available at: http://www.un.org/ga/search/view_doc.asp?symbol=A/RES/70/1 (Accessed: 3 September 2017).

U.N. (2017) *Global indicator framework for the Sustainable Development Goals and Targets of the 2030 Agenda for Sustainable Development*, *A/RES/71/313*. Available at: https://unstats.un.org/sdgs/indicators/indicators-list/ (Accessed: 14 May 2019).

Verbrugge, S. *et al.* (2011) 'Research approach towards the profitability of future FTTH business models', in *Future Network & Mobile Summit (FutureNetw), 2011*. IEEE, pp. 1–10.

Viusasa (no date) *Viusasa*. Available at: https://viusasa.com (Accessed: 20 November 2017).

Waites, W. *et al.* (2016) 'Remix: A distributed internet exchange for remote and rural networks', in *Proceedings of the 2016 workshop on Global Access to the Internet for All*. ACM, pp. 25–30.

Walubiri, M. (2018) 'Uganda's internet growth second in EA', *New Vision*. Vol.33 No.222, 6 November, p. 5.

WBG (2017) *World Indicator Database, World Economic Outlook Database*. Available at: https://data.worldbank.org/indicator/NY.GDP.PCAP.CD (Accessed: 7 November 2018).

WEF (2016) *Internet for All: A Framework for Accelerating Internet Access and Adoption*. White Paper. World Economic Forum. Available at: http://www3.weforum.org/docs/WEF_Internet_for_All_Framework_Accelerating_Internet_Access_Adoption_report_2016.pdf (Accessed: 3 November 2018).

WEF (2017) *Internet for All: An Investment Framework for Digital Adoption*. White Paper. World Economic Forum. Available at: http://www3.weforum.org/docs/White_Paper_Internet_for_All_Investment_Framework_Digital_Adoption_2017.pdf (Accessed: 3 November 2018).

WEF (2018) *Internet for All: Financing a Forward-Looking Internet for All*. White Paper. World Economic Forum. Available at: http://www3.weforum.org/docs/WP_Financing_Forward-Looking_Internet_for_All_report_2018.pdf (Accessed: 3 November 2018).

Weldon, M. K. (2015) *The Future X Network: A Bell Labs Perspective*. CRC PressINC. Available at: https://books.google.ie/books?id=DZEZjgEACAAJ.

Williamson, K. (2002) *Research Methods for Students, Academics and Professionals: Information Management and Systems*. Elsevier Science. Available at: https://books.google.co.ug/books?id=4veiAgAAQBAJ.

WTO (2016) *World Trade Report 2016: Levelling the trading field for SMEs*. World Trade organisation (WTO). Available at: https://www.wto.org/english/res_e/booksp_e/world_trade_report16_e.pdf (Accessed: 31 October 2018).

9. Appendices

Appendix A: **IXPBuilder Manual**

Appendix B: **political economy study**
- B1: **Interview Guide and Participant Information Leaflet**
- B2: **Survey Form**

INSTITUTE of TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

Enabling models of Internet eXchange Points to support spatial planning: the case for East Africa

Appendix A

IXPBuilder manual

Institute of Technology, Carlow

IXP Builder

Internet eXchange Point Builder

Installation and Operations Manual

gamecore
engaging people with technology

GameCORE and netLabs!UG Research Centres
Institute of Technology, Carlow and Makerere University
September 2019
Version 5.2.2

Copyright 2019 C²S Consulting

Support

Support queries can be made to: *ixpbuilder@netlabsug.org*

GameCORE *is a research centre of the Institute of Technology, Carlow, Ireland*

netLabs!UG *is a research centre of Makerere University, Kampala, Uganda*

This project was carried out under the Memorandum of Understanding between Makerere University and the Institution of Technology, Carlow, 16th October 2015.

Table of Contents

List of Illustrations

List of Tables

Table of Abbreviations

AfriNIC	African Regional Network Information Centre
API	Application Programming Interface
ASP	Application Service Providers
ASN	Autonomous System Number
AS	Autonomous System
BIND	Berkeley Internet Name Domain
BIND9	Berkeley Internet Name Domain version 9
BIRD	Bird Internet Routing Daemon
BGP	Border Gateway Protocol
BS	AS112 Blackhole Server on IXPBuilder
CDIP	Control/Data Interface Pair
cIXP	core Internet eXchange Point
CLI	Command Line Interface
CS	Route Collector Server on IXPBuilder
DNS	Domain Name Service
DHCP	Dynamic Host Configuration Protocol
dIXP	Distributed IXP, an ecosystem consisting of a cIXP and mIXPs
eBGP	external Border Gateway Protocol
HDD	Hard-drive
IXP	Internet eXchange Point
IP	Internet Protocol
ISP	Internet Service Provider
IPv4	IP version 4
IPv6	IP version 6
KVM	Kernel Virtual Machine
LXC	LinuX Containers
LXD	Linux Hypervisor Daemon
LAN	Local Area Network
mIXP	mini Internet eXchange Point
NS	DNS Server on IXPBuilder
NIC	Network Interface Card
OvS	Open virtual Switch – Open vSwitch
OS	Operating System
OOB	Out of Band Management
PoC	Proof of Concept
RAID	Redundant Array of Independent Disks
REN	Regional Education Networks
RS	Route Server on IXPBuilder
RSA	Rivest, Shamir, Adleman
SATA	Serial Advanced Technology Attachment
SC	SDN Controller on IXPBuilder
SLAAC	Stateless Address Autoconfiguration
SQL	Structured Query Language
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
YAML	YAML Ain’t Markup Language

1. Introduction

1.1. Internet eXchange Point Builder (IXPBuilder)

IXPBuilder is a software Proof of Concept (PoC) application developed in Python version 3 (python3) to simplify the process of building Internet eXchange Points (IXP). It provides a mechanism to build and manage the IXP *core tier* which is a containerised core incorporating the functionality of a Route Collector (CS), a Route Server (RS), a Domain Name Service (NS) and a AS112 Blackhole service (BS) plus an SDN Controller (SC) in models where one is required. It consists of a python class and function library module `ixp.py` accessed via a Command Line Interface (CLI) program `ixp` and a set of template files used as part of the host and container build and configure processes. The *core tier* provides services to members in the *peer tier* via an internal switch, external switches or a combination of these at the *switching tier*.

IXPBuilder is represented in Illustration 1 as a block diagram. It is built upon the Ubuntu 18.04 Server Long Term Support (LTS) (Bionic Beaver) Operating System (OS) which is supported by Canonical Limited until April 2023.

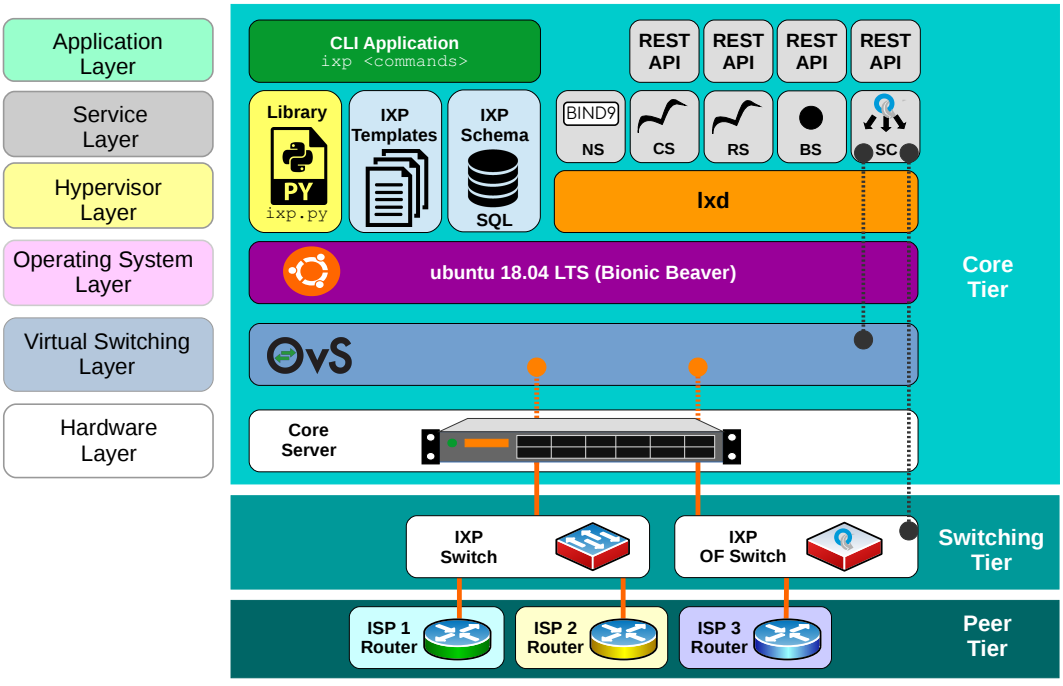


Illustration 1: IXPBuilder Phase 3.0 block diagram

The system is presented as three tiers, a *peer tier* representing the hardware of the IXP members, a *switching tier* representing both the internal Open virtual Switch (OvS) peering bridge and any external switching hardware of the IXP and a *core tier* that is a containerisation of the IXP functions. This tier is further described as six software layers on the IXP server. For the purpose of testing, a Dell PowerEdge R610 with 2 Intel Xeon X5670 processors, an inbuilt PowerEdge Redundant Array of Independent Disks (RAID)

Controller (PERC) H700 RAID Controller configuration with two 1TB 7.2K 2.5" Serial Advanced Technology Attachment (SATA) hard-drives (HDD) was employed, though the software can be used on any comparable hardware. For networking, the server has a Broadcom NetXtreme II BCM5709 Quad Ethernet Network Interface Card (NIC) to complement the embedded pair of dual port Broadcom NetXtreme II 5709c Gigabit Ethernet NICs giving eight physical Gigabit Ethernet interfaces. These components together form the *hardware layer*.

Ubuntu 18.04 Server LTS performs the function of the *OS Layer* and OvS provides a *vSwitching Layer*.

The *hypervisor layer* consists of an LinuX hypervisor (LXD) which hosts IXP functions in LinuX Containers (LXC) at the *service layer*.

The IXPBuilder program (`ixp`) is a python3 application that uses the IXP python class and function library module (`ixp.py`). The module stores configuration settings in an SQL database and uses IXP templates to build the various IXP services.

1.2. IXP Build steps

For each site the initial build of the IXP consists of the following steps. The difference between the cIXP and mIXPs is identified by the selection of the appropriate `--site-type` of *core* or *mini* in the IXP schema. The remaining configuration is carried out as follows;

1. ***Create an IXP Schema***
   The *ixp schema* process creates a domain and numbering framework upon which the IXP is built. This includes Domain, Autonomous System (AS) Number (ASN), Peering Local Area Network (LAN) and Management LAN Internet Protocol (IP) addressing. It also stores site information used by the *NS* as well as the IXP type.

2. ***Setup the IXP host***
   The Ubuntu 18.04 LTS is installed on the host server to provide the *OS layer*. IXPBuilder lists the network interfaces available at the *hardware layer* and presents model options to the IXP build engineer as part of the *ixp host* process.

3. ***Setup the IXP switch parameters***
   If the IXP Host is configured with `--switch` set to a figure greater than 0 then an external switch template will already exist in the database from the *ixp host* process. Some additional parameters can be adjusted as required with the *ixp switch* process.

4. ***Configure LXD on the host and install the containers***
   The *service layer* is built using LXCs, each to provide a specific service; *NS*, *RS*, *CS* and *BS* plus *SC* if the host `--switching-type` is set to *software-defined*. Each are connected to the management bridge in the internal OvS and all except the *SC* are connected to the peering bridge in the same OvS.

5. ***Install and configure the IXP Server software***
   There are three sub-steps in the *ixp server* process, firstly the LXCs are upgraded and updated from online repositories. Then the function specific software is installed in each LXC. These software functions are configured in line with the IXP schema to provide the services at the *service layer*.

6. ***Review SDN switches and flows***
   The *software-defined* functionality of IXPBuilder operates behind the scenes. With *ixp sdn* process the IXP engineer can view switches from the *SC*'s perspective as well as the OpenFlow (OF) flows injected into the switch.

7. ***Configure IXP peers***
   IXP peers can be added to the three functions provided with the Bird Internet Routing Daemon (BIRD), *RS*, *CS* and *BS*, via the *ixp peer* process. This allows peering members to establish Border Gateway Protocol (BGP) peer relationships with the IXP. Closed peering inclinations are catered for by permitting opt out from peering with both the *RS* and the *BS*; however, connection to the *CS* server is mandatory.

8. ***Configure the IXP Switch***
   The *switching tier* consists of an IXP OvS and optional IXP *traditional* Ethernet switch or an OF based Ethernet switch. Depending on the model selected by the IXP build engineer the *traditional* Ethernet switches are connected using a Virtual LAN (VLAN) trunk which connects both the management and peering LANs to configured *traditional* Ethernet switches. Interfaces on each *traditional* Ethernet switch are then assigned to VLANs depending on their function. Alternatively, in a *software-defined* configuration, OF switches can be connected to the OvS and are managed by the *SC* via the management LAN.

2. Install Ubuntu 18.04 LTS

Before installing the IXPBuilder software, the server must have the Ubuntu 18.04 LTS OS installed. This OS was chosen as Canonical Limited, the developers of Ubuntu, also developed LXD/LXC and as such it makes sense to use the base OS too. It is also the base OS for a number of projects relating to SDN. The following steps should be followed:

a) Create a USB Disk or use a CD/DVD with the *ubuntu-18.04-server-amd64.iso* image on it.
b) Boot the server from the CD/DVD or ISO image and install a minimal Ubuntu Server to include the *OpenSSH* package. When the installer is running make the following selections:

- Language: **English**
- **Install Ubuntu Server**
- Select a language: **<local preference**, i.e. English**>**
  - Select your location: **Other**
  - Select your Continent or Region: **<local preference**, i.e. Africa**>**
  - Select your location: **<local preference**, i.e. Uganda**>**
  - Configure locales: **<local preference**, i.e. United Kingdom – en_GB.UTF.8**>**
- Configure the keyboard
  - Detect keyboard layout: **<No>**
  - Origin for the keyboard: **<local preference**, i.e. English (UK)**>**
  - Keyboard layout: **<local preference**, i.e. English (UK)>
- Configure the network
  - Primary network interface: **<local preference, select first interface,** i.e. eno1**>**
  - IP address will autoconfigure
  - Name server addresses: **8.8.8.8 8.8.8.4**
  - Hostname: **cIXP**
- Choose a mirror of the Ubuntu archive: **<local preference**, i.e. Uganda**>**
  - Ubuntu archive mirror: **<local preference**, i.e. ug.archive.ubuntu.com >
  - HTTP proxy information: **(blank for none)**
- Setup user and password
  - Full name of new user: **ubuntu user**
  - Username for your account: **<local preference**, i.e. ubuntu >
  - Password for your account: **<local preference**, i.e. ubuntu >
  - Use weak password? **<Yes>**
- Encrypt home directory: **<No>**
- Configure the clock: **<local preference**, i.e. Africa/Kampala>
- Partitioning method: **Guided - use entire disk**
  - Select disk to partition: **SCSI1 (0,0,0) (sda)**
  - Write the changes to disks? **<Yes>**
- Configuring tasksel: **No automatic updates**

- Software selection
  - Choose software to install: [*] **OpenSSH server**
- Install the GRUB boot loader on a hard disk: **<Yes>**
  - Device for boot loader installation: **<typically /dev/sda>**

c) Check the install.

```
ubuntu@lxd1:~$ lsb_release -cidr
Distributor ID:  Ubuntu
Description:     Ubuntu 18.04
Release:         18.04
Codename:        bionic
```

3. Installation of IXPBuilder

With the Ubuntu 18.04 server installed, copy the *ixp_builder-v5.2.1.tgz* file to the home directory of the created user and extract it.

```
cIXP:~$ tar -xzvf ~/IXPBuilder_v5.2.1.tgz
cIXP:~$ ls ~
IXPBuilder_v5.2.1.tgz  ixp
```

Switch to the `/ixp/tools/` directory of the extracted application and run the installer script with root privileges via the *sudo* command.

```
cIXP:~$ cd ~/ixp/tools
cIXP:~/ixp/tools$ sudo ./ixp-install.sh
```

After the install, logout of the shell and log back in again so the changes to the users group membership made by the install can be enabled. This can be chacked as follows;

```
cIXP:~$ cat /etc/group | grep ixp
ixp:x:1001:ubuntu
```

The IXPBuilder installation script creates a new group called *ixp*. It creates a pre-seed YAML file for LXD and carries out the LXD initialisation process. It also creates a dual NIC profile that will be used as a template by LXD when new LXC are being launched. Files are created and placed in the directories as demonstrated in Illustration 2.

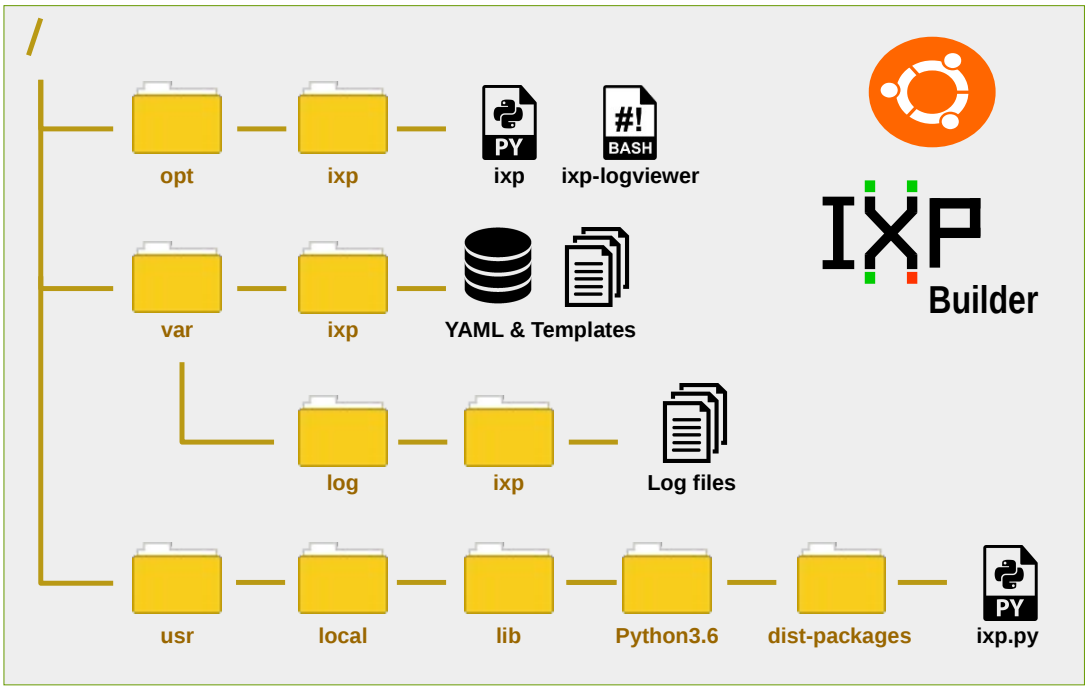


Illustration 2: IXPBuilder file layout

The `/opt/ixp/` directory houses the main IXPBuilder application (`ixp`) which is accessed via the softlink `/usr/local/bin/ixp`. The install script also creates and loads the `/var/ixp/` directory with the initial template files. This directory is also used by IXPBuilder to store the IXP database as well as configured files generated by IXPBuilder later during operation. A `/var/log/ixp/` log directory is created and within it, daily log files are generated, each new log file is triggered by the first `ixp` command executed on any particular day.

At the heart of IXPBuilder is the IXP class and function library module, '`ixp.py`'. This is a python3 module consisting of 30 functions as well as 11 classes which include 153 methods.

These library classes facilitate;

- **IxpHelp –** IXP Help class, provides methods to respond to help commands.
- **_IxpLxc –** IXP LXC class, a private class to provide methods for LXC container management.
- **IxpSchema –** IXP Schema Class, provides methods to develop the IXP site schema.
- **IxpHost –** IXP Host Class, contains methods to prepare the core server for its role as a host to IXP containerised services.
- **IxpSwitch –** IXP Switch Class, contains methods to configure and review the IXP switches.
- **IxpOpenflow –** IXP Openflow Class, contains methods to review and to update information on OF switches in the IXP Schema.
- **IxpServer –** IXP Server Class, creates the LXC defined in the IXP Schema.
- **IxpSoftware –** The IXP Software Class, contains methods that configure the LXC daemons to provide the IXP functions.
- **IxpSdn –** The IXP SDN Class, contains methods that allow the operator to interrogate the *SC*. It is also a skeleton class for new *software-defined* features to be developed to extend the IXP functionality.
- **IxpPeer –** The IXP Peer Class, facilitates the management of IXP peers and the monitoring of peers and routes.
- **IxpRemote –** The IXP Remote Class, facilitates the day-to-day management and the monitoring of remote mIXP sites from the cIXP.

The `ixp` CLI application accepts commands from the host shell. These commands pass information to the appropriately instantiated library classes.

4. IXPBuilder help and troubleshooting

4.1. The IXPBuilder version

The IXPBuilder version can be viewed from the application as follows;

```
cIXP:~$ ixp version
IXPBuilder: Phase 3.0, Version 5.2
            Copyright 2019, C²S Consulting
            European Union Public Licence v1.2
```

4.2. The IXPBuilder help

Each command level in IXPBuilder offers a help option. Here is an example of the top level help command;

```
cIXP:~$ ixp help
Usage: ixp { help | ? } [OPTION] ... ...

        help, ?    - This help message.

        OPTION [list]:
        version   - Returns the version of the IXPBuilder module.
        schema    - Builds an IP Schema for the testbed.
        host      - Builds configuration for the host.
        switch    - Configuration and monitoring of IXP switches.
        server    - Operations on the IXP container servers.
        software  - Installation and configuration of software on
                     the IXP container servers.
        peer      - Management of IXP peers.
        route     - Review IXP container server routes.
        remote    - Access mIXP from the cIXP command shell.
```

4.3. The IXPBuilder logviewer

After the IXPBuilder has ran at least one command, logs are generated to the `/var/log/ixp/` directory. It is possible to monitor a running log in a second shell using the `ixp-logviewer` command as follows:

```
cIXP:~$ ixp-logviewer

IXP Logger enabled
------------------

20190215-232433: 'ixp help' called.
20190215-232433: INFO[_read_db]: Extracting column names from schema
  PRAGMA table_info(schema)
20190215-232433: INFO[_read_db]: Extracting data from schema
  PRAGMA table_info(schema)
20190215-232433: INFO[_read_db]: Importing configuration from database
```

5. The IXP Schema (`ixp schema`)

The IXP schema is presented to the IXPBuilder application as tables within a Structured Query Language (SQL) database. Data is stored across nine tables:

- host
- ipv4_peer
- ipv6_peer
- site
- remote
- ipv4_man
- ipv6_man
- schema
- switch

The IXP schema maintains the following list of values for IXP container functions *NS*, *RS*, *CS*, *BS* and *SC* in the schema table within the database;

- IP version 4 (IPv4) address on the peering LAN,
- IP version 6 (IPv6) address on the Peering LAN,
- IPv4 address on the Management LAN,
- IPv6 address on the Management LAN,
- The IXP Domain name,
- The IXP assigned ASN.

Additional site information data is also maintained;

- The country in which the IXP is located,
- The town or city the IXP is located,
- The IXP site elevation (m) above sea level,
- The latitude and longitude of the site.

The hostnames for each LXC adopt the site number. For the purpose of this document examples assume the cIXP site is number *1* and mIXP sites are numbered from *2*.

This IXP schema and site tables are built with the `ixp schema build` command. IP network addressing, domain and schema information is given using appropriate command switches;

- `-sn, --site-number` - Site Number (i.e. **1**-100)
- `-st, --site-type` - Core IXP or Mini IXP (i.e. **core**|mini)
- `-sw --switching-type` - Switching (i.e. **traditional**|software-defined)
- `-o, --organisation` - Organisation (i.e. **netLabs!UG**)
- `-p4, --peer-ipv4` - IPv4 Peering LAN (i.e. **199.9.9.0/24**)
- `-p6, --peer-ipv6` - IPv6 Peering LAN (i.e. **2a99:9:9::/48**)
- `-m4, --man-ipv4` - IPv4 Management LAN (i.e. **198.8.8.0/24**)
- `-m6, --man-ipv6` - IPv6 Management LAN (i.e. **2a98:8:8::/48**)
- `-d, --domain` - IXP Domain name (i.e. **netlabs.tst**)
- `-as, --as-number` - IXP Autonomous System number (i.e. **5999**)
- `-c, --country` - Country IXP is located in (i.e. **Uganda**)
- `-t, --town` - Town/city IXP is located in (i.e. **Kampala**)
- `-e, --elevation` - IXP elevation (m) above sea level (i.e. **1027**)
- `-la, --latitude` - Deg, min, sec, N | S (i.e. '**00 20 51.3456 N**')
- `-lo, --longitude` - Deg, min, sec, E | W (i.e. '**32 34 57.0720 E**').

The default IXP schema can be implemented using the following command. The default settings are indicated in bold above.

```
cIXP:~$ ixp schema default all
  ------------------------------------------------------
  WARNING: This command erases the database configuration
  ------------------------------------------------------

  Are you certain you want to proceed with this step? (y/n): y
  Building IXP schema ...

IXP Schema inserted in the schema, IPv4 and IPv6 database tables

IXP Site information inserted in the site database table
```

6. The IXP Host (`ixp host`)

6.1. Traditional Switching

If the site `--switching-type` is set to *traditional* in the IXP schema then this process builds the host as well as the on-board OvS based on five models as shown in Table 1. IXPBuilder detects the number of available Ethernet interfaces and the IXP build engineer specifies the number of Ethernet switches required, IXPBuilder then selects the model to apply, or where a choice is possible, presents options to the IXP build engineer.

Table 1: Traditional models

Model	Interfaces	Interfaces			
		1	2	3	4+
A	1	Trunk			
B	2	OOB	Trunk interface		
C	> 3	OOB	Management	Peer interfaces	
D_1	> 3	OOB	Management	Trunk interface	Peer interfaces
D_2	> 3	OOB	Management	Mix of Peer and Trunk interfaces	
E_1	3	OOB	Management	Trunk interface	NIL
E_2	> 3	OOB	Management	Trunk interfaces	

6.1.1) Selecting the IXP host traditional model

In the case of a single or two Ethernet interface the program will automatically select from model A or model B.

If the number of Ethernet interfaces is three or greater and no switches are specified, then it is assumed that there are no external switches to be configured. Should the number of selected switches be less than the number of available Ethernet interfaces (interfaces – 2) then the IXP build engineer is given the option to choose between model C as demonstrated in Illustration 3, model D as demonstrated in Illustration 4 or model E as demonstrated in Illustration 5. In the case of each of these models the first interface is configured for Out of Band (OOB) management while the second is configured for connection to the management LAN.

In the case of model C the remaining interfaces are established for peering. In model D the third interface is configured as a VLAN trunk and the remaining interfaces for peering. Model E has the remaining interfaces established as VLAN trunks which are connected to VLAN configured *traditional* Ethernet switches.

Note: For the purpose of testing Model A or Model B in a multi-port switch environment two command switches are included in the *traditional* switching model that will simulate models' A or B, by masking the number of actual Ethernet interfaces from the IXP host build method:

```
        -sa, --sim-model-a - Simulate model A
        -sb, --sim-model-b - Simulate model B
```

```
cIXP:~$ ixp host build

Configuring the IXP host for traditional switching model: c

  -------------------------------------
  INSTALLER NOTE
  "Traditional switching"
  Model: C is ideal as an IXP, switches
  do not require VLANs to be configured
  -------------------------------------

  IXP Host interfaces - Model 'C'
 + ------------------------------ +
 |        Host interfaces         |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.202/24
 | eth2          | Management    |
 | eth3          | Local peer    | 199.9.9.2/24, 2a99:9:9::2/48
 | eth4          | Local peer    | 199.9.9.3/24, 2a99:9:9::3/48
 | eth5          | Local peer    | 199.9.9.4/24, 2a99:9:9::4/48
 | eth6          | Local peer    | 199.9.9.5/24, 2a99:9:9::5/48
 | eth7          | Local peer    | 199.9.9.6/24, 2a99:9:9::6/48
 | eth8          | Local peer    | 199.9.9.7/24, 2a99:9:9::7/48
 + ------------- + ------------- +

Completed configuration of IXP host
```

Illustration 3: IXP Host build - model C

Note: As no command switch options are given IXPBuilder assumes that no external Ethernet switches are being configured.

```
cIXP:~$ ixp host build --switch 3

Traditional Switching Models
----------------------------

Model C: OOB, Management, and remaining 'Peer' interfaces
Model D: OOB, Management, one 'Trunk' interface and remaining 'Peer' interfaces
Model E: OOB, Management, and remaining 'Trunk' interfaces

  Choice of model, 'C', 'D' or 'E': d

Configuring the IXP host for traditional switching model: D

  -------------------------------------
  INSTALLER NOTE
  "Traditional switching"
  Model: D is ideal as an IXP, switches
  require VLANs 100 and 900 configured
  for peering and management
  -------------------------------------

  IXP Host interfaces - Model 'D'
 + ----------------------------- +
 |        Host interfaces        |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.202/24
 | eth2          | Management    |
 | eth3          | Trunk         |
 | eth4          | Trunk         |
 | eth5          | Trunk         |
 | eth6          | Local peer    | 199.9.9.2/24, 2a99:9:9::2/48
 | eth7          | Local peer    | 199.9.9.3/24, 2a99:9:9::3/48
 | eth8          | Local peer    | 199.9.9.4/24, 2a99:9:9::4/48
 + ------------- + ------------- +

Completed configuration of IXP host
```

Illustration 4: IXP Host build - model D

```
cIXP:~$  ixp host build --switch 6

Configuring the IXP host for traditional switching model: E

  ------------------------------------------
  INSTALLER NOTE
  "Traditional switching"
  Model: E is ideal as an IXP, all interfaces
  are trunked except the first two. Switches
  require VLANs 100 and 900 configured for
  peering and management
  ------------------------------------------
  IXP Host interfaces - Model 'E'
 + ------------------------------ +
 |        Host interfaces         |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.202/24
 | eth2          | Management    |
 | eth3          | Trunk         |
 | eth4          | Trunk         |
 | eth5          | Trunk         |
 | eth6          | Trunk         |
 | eth7          | Trunk         |
 | eth8          | Trunk         |
 + ------------- + ------------- +
Completed configuration of IXP host
```

Illustration 5: IXP Host build - model E

6.1.2) Selecting the IXP host software-defined model

If the site `--switching-type` is set to *software-defined* in the IXP schema then this process builds the host as well as the on-board OvS based on three additional models as outlined in Table 2.

Table 2: Software-defined models

Model	Interfaces	Interfaces			
		1	2	3/4	5+
S	> 3	OOB	Management	Peer interfaces	
T_1	> 3	OOB	Management	OF 1 CDIP	Peer interfaces
T_2	> 3	OOB	Management	Mix of CDIPs and Peer interfaces	
U	> 3	OOB	Management	CDIPs	

For *software-defined* models the server hardware requires at least 3 Ethernet interfaces; however, with just 3 interfaces Model S with a single *peer* interface is the only available model for the hardware configuration. For all *software-defined* models the first and second interfaces are reserved for OOB and management LAN functions. Model S has all remaining interfaces configured as *peer* interfaces as shown in Illustration 6.

```
cIXP:~$ ixp host build

Configuring the IXP host for software-defined model: S

  ----------------------------------
  INSTALLER NOTE
  "Software-defined switching"
  Model: S is ideal for mIXP and can
  support 6 unmanaged traditional
  Ethernet switches
  ----------------------------------

  IXP Host interfaces - Model 'S'
 + ---------------------------- +
 |        Host interfaces       |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.202/24
 | eth2          | Management    |
 | eth3          | Local peer    | 199.9.9.2/24, 2a99:9:9::2/48
 | eth4          | Local peer    | 199.9.9.3/24, 2a99:9:9::3/48
 | eth5          | Local peer    | 199.9.9.4/24, 2a99:9:9::4/48
 | eth6          | Local peer    | 199.9.9.5/24, 2a99:9:9::5/48
 | eth7          | Local peer    | 199.9.9.6/24, 2a99:9:9::6/48
 | eth8          | Local peer    | 199.9.9.7/24, 2a99:9:9::7/48
 + ------------- + ------------- +
Completed configuration of IXP host
```

Illustration 6: IXP Host build - model S

Connection to OF switches requires two ports, one for the OF management interface which is often via the OOB interface on the physical switch and a second as a data plane interface. This is called the Control/Data Interface Pair (CDIP)*. Model T contains a mix of CDIPs to connect to OF switches as well as some remaining interfaces used for *peers* (as shown in Illustration 7). Finally model U, where there must be an even number of Ethernet interfaces and all interfaces from 3 are used for CDIPs (as can be seen in Illustration 8).

*** *Note:*** While OF does allow for the control-channel to exist in the same network as the data, it is not advisable in the context of the IXP where there is a clear demarcation between the peering and management LAN functions. For this reason the *SC* does not have an interface in the peering LAN.

```
cIXP:~$ ixp host build --switch 1

Configuring the IXP host for software-defined model: T

  -------------------------------------
  INSTALLER NOTE
  "Software-defined switching"
  Model: T supports a mix of 1 OpenFlow
  switches as well as 4 peers
  -------------------------------------

  IXP Host interfaces - Model 'T'
 + ----------------------------- +
 |        Host interfaces        |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.202/24
 | eth2          | Management    |
 | eth3          | Ctrl (SDN)    | Switch 1 OOB
 | eth4          | Peer (Data)   | Switch 1 Port 1
 | eth5          | Local peer    | 199.9.9.2/24, 2a99:9:9::2/48
 | eth6          | Local peer    | 199.9.9.3/24, 2a99:9:9::3/48
 | eth7          | Local peer    | 199.9.9.4/24, 2a99:9:9::4/48
 | eth8          | Local peer    | 199.9.9.5/24, 2a99:9:9::5/48
 + ------------- + ------------- +
Completed configuration of IXP host
```

Illustration 7: IXP Host build - model T

```
cIXP:~$ ixp host build --switch 3

Configuring the IXP host for software-defined model: U

  -------------------------------------
  INSTALLER NOTE
  "Software-defined switching"
  Model: U supports 3 OpenFlow switches
  -------------------------------------

  IXP Host interfaces - Model 'U'
 + ----------------------------- +
 |        Host interfaces        |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.202/24
 | eth2          | Management    |
 | eth3          | Ctrl (SDN)    | Switch 1 OOB
 | eth4          | Peer (Data)   | Switch 1 Port 1
 | eth5          | Ctrl (SDN)    | Switch 2 OOB
 | eth6          | Peer (Data)   | Switch 2 Port 1
 | eth7          | Ctrl (SDN)    | Switch 3 OOB
 | eth8          | Peer (Data)   | Switch 3 Port 1
 + ------------- + ------------- +
Completed configuration of IXP host
```

Illustration 8: IXP Host build - model U

At any time the operator can review the configured model on the host. This is demonstrated in Illustration 9.

```
cIXP:~$ ixp host show
  IXP Host interfaces - Model 'D'
 + ---------------------------- +
 |        Host interfaces       |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.201/24
 | eth2          | Management    |
 | eth3          | Trunk         |
 | eth4          | Local peer    | 199.9.9.2/24, 2a99:9:9::2/48
 | eth5          | Local peer    | 199.9.9.3/24, 2a99:9:9::3/48
 | eth6          | Local peer    | 199.9.9.4/24, 2a99:9:9::4/48
 | eth7          | Local peer    | 199.9.9.5/24, 2a99:9:9::5/48
 | eth8          | Local peer    | 199.9.9.6/24, 2a99:9:9::6/48
 + ------------- + ------------- +


cIXP:~$ ixp host show
  IXP Host interfaces - Model 'T'
 + ---------------------------- +
 |        Host interfaces       |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.202/24
 | eth2          | Management    |
 | eth3          | Ctrl (SDN)    | Switch 1 OOB
 | eth4          | Peer (Data)   | Switch 1 Port 1
 | eth5          | Ctrl (SDN)    | Switch 2 OOB
 | eth6          | Peer (Data)   | Switch 2 Port 1
 | eth7          | Local peer    | 199.9.9.2/24, 2a99:9:9::2/48
 | eth8          | Local peer    | 199.9.9.3/24, 2a99:9:9::3/48
 + ------------- + ------------- +
```

Illustration 9: IXP host show

7. The IXP Switches (`ixp switch`)

The number of switches was defined during the *ixp host* build process. The *ixp switch* process allows the IXP build engineer to interface with both the internal switch and any additional external switches. The *ixp switch set* and *ixp switch set speed* commands have the following command switch options;

- `-s, --switch-number` - Switch Number (i.e. 2 but not 0).
- `-sn, --switch-name` - Switch Name (i.e. core_sw_1).
- `-st, --switch-type` - Switch Type (i.e. M4300-48G).
- `-m, --manufacturer` - Manufacturer (i.e. Netgear).
- `-si, --switch-interfaces` - Switch Interfaces, number 1 - 99 (i.e. 2).
- `-ei, --enabled-interfaces` - Enabled Interfaces, list (i.e. 2,4,6-9,11,13-20 or 'all').
- `-is, --interface-speed` - Interface speed from 100M, 1G, 10G or 40G, default 1G).

In the case of external switches this command defines the number of interfaces available. Additionally in SDN models the command also identifies the specific interfaces to be enabled. These interfaces are then assigned an IP address in the database.

The build engineer can also specify the speed of each interface.

```
cIXP:~$ ixp switch set --switch-number 1 --switch-interfaces 24

cIXP:~$ ixp switch set speed -s 0 --enabled-interfaces 7-8 \
 --interface-speed 10G

Switch 0 configuration complete

cIXP:~$ ixp switch set speed -s 1 -ei 21-24 -is 10G

Switch 1 configuration complete
```

The process to confirm the switch configuration on the switches is outlined below.

Note: that both the control and data interfaces are assigned to the *eth3* interface because this is a *traditional* Ethernet switch connected to a *traditional* model core.

```
cIXP:~$ ixp switch show

Switch No. 0
------------

Switch Name         : int
Switch Type         : OvS
Manufacturer        : LF
Control Interface   : int
data_interface      : int
Switch Interfaces   : 6
                      1G   3-6
                      10G  7-8
enabled_interfaces  : 3-8
OOB IPv4 address    : N/A
OOB IPv6 address    : N/A

switch no. 1
------------

Switch Name         : sw_1
Switch Type         : Traditional Ethernet
Manufacturer        : Traditional Ethernet
Control Interface   : eth3
Data Interface      : eth3
Switch Interfaces   : 24
                      1G   3-20
                      10G  21-24
Enabled Interfaces  : 3-24
OOB IPv4 address    : 198.8.8.241/24
OOB IPv6 address    : 2a98:8:8::241/48
```

Considering an example for an OvS external OF switch connected to a *software-defined* core. In this case it is necessary to define the `--switch-type` so it is clear to IXPBuilder if the first interface is used in lieu of an OOB interface on the external OF switch. For example an OvS based external OF switch does not have a separate OOB port whereas the Netgear switch does.

```
ubuntu@cixp:~$ ixp switch set -s 1 -si 8 -ei all --switch-type ovs

Switch 1 configuration complete

cIXP:~$ ixp switch set speed -s 0 -si 7-8 -is 10G

cIXP:~$ ixp switch set speed -s 1 -si 7-8 -is 10G
```

Confirm the switch configuration on the switches as shown below. The control and and data interfaces of the external OF switch are connected to interfaces *eth3* and *eth4* respectfully, this forms the CDIP. As this is an OvS external OF switch, the first interface is connected to *eth3* and the second interface is connected to *eth4*.

Note: that if the external OF switch was a Netgear M4300-28G then the OOB interface is connected to *eth3* and the first interface is connected to *eth4*.

```
ubuntu@cixp:~$ ixp switch show

Switch No. 0
------------

Switch Name         : int
Switch Type         : OvS
Manufacturer        : LF
Control Interface   : int
Data Interface      : int
Switch Interfaces   : 6
                      1G   3-6
                      10G  7-8
enabled_interfaces : 3-8
OOB IPv4 address    : N/A
OOB IPv6 address    : N/A

switch no. 1
------------

Switch Name         : sw_1
Switch Type         : ovs
Manufacturer        : Bare-metal
Control Interface   : eth3
Data Interface      : eth4
Switch Interfaces   : 8
                      1G   3-6
                      10G  7-8
Enabled Interfaces : 3-8
OOB IPv4 address    : 198.8.8.241/24
OOB IPv6 address    : 2a98:8:8::241/48
```

8. The IXP Servers (`ixp server`)

The *ixp server* build process confirms there is a template Ubuntu 18.04 LXC. If there not, then it retrieves one from the Internet and installs it.

It then assigns a dual NIC profile to the template which is used to generate VLANs for the peering and management LANs on each LXC once built. Each of the LXC container functions, *NS*, *RS*, *CS*, *BS* and *SC* are built and their networks are configured in accordance with the structure laid out in the IXP schema. The site number is appended to the LXC name.

```
cIXP:~$ ixp server build ?
Usage: ixp server build { help | ? } [ARGUMENT] [OBJECT(s)]

        ?, help  -  This help message.

        ARGUMENT
        -y       -  Skip the yes/no questions.

        OBJECT(s) [list]:
        all      -  Build all IXP Servers.
        ns1      -  Build Name Server.
        cs1      -  Build CS.
        rs1      -  Build RS.
        bs1      -  Build AS112 BS.
        sc1      -  Build SC Server.
```

The following command builds the core function containers *NS*, *RS*, *CS*, *BS* and *SC*. It configures each LXC IP configuration according to the IXP schema. The command then returns the state of each as confirmation.

```
cIXP:~$ ixp server build -y all

Container state Summary
-----------------------
ns1:     Running and has good Internet connectivity
cs1:     Running and has good Internet connectivity
rs1:     Running and has good Internet connectivity
bs1:     Running and has good Internet connectivity
sc1:     Running and has good Internet connectivity
```

LXC details can be obtained via the `ixp server show` command.

```
cIXP:~$ ixp server show rs1 --connectivity-test
+------+--------------------+--------------------------------------+---------+
| NAME |        IPV4        |                 IPV6                 |  STATE  |
+------+--------------------+--------------------------------------+---------+
| rs1  | 199.9.9.1 (ens3)   | 2a99:9:9::1 (ens3)                   | RUNNING |
|      | 198.8.8.232 (ens4) | 2a98:8:8::232 (ens4)                 |         |
|      |                    | 2a98:8:8:0:216:3eff:fe8e:e21a (ens4) |         |
+------+--------------------+--------------------------------------+---------+

  rs1 has good Internet connectivity
```

9. The IXP Software (`ixp software`)

9.1. IXP Software Install steps

The *ixp software install* process constructs the *netplan* for each container, *NS*, *RS*, *CS*, *BS* and *SC* if necessary. *Netplan* is a new Ubuntu utility for configuring networking based on YAML Ain't Markup Language (YAML) descriptor files. This replaces the `/etc/network/interfaces`, *ifupdown*, networking system. *Netplan* generates the networking information from YAML description files located in the `/etc/netplan/` directory for each LXC.

```
cIXP:~$ ixp software install all
```

The *ixp software install* process changes the Domain Name Service (DNS) server for each LXC to the public server defined in the `ixp.py` class and function library module to ensure access to the Ubuntu repositories during software installation process. It then proceeds to update each LXC software repository list and the OS distribution is upgraded to the latest versions. Finally the software packages required of each server are installed;

- ***ns1***: bind9, bind9utils
- ***cs1***, ***rs1***: bird, birdseye
- ***bs1***: bird, birdseye, bind9, bind9utils
- ***sc1***: python3-ryu

9.2. IXP Software Configure steps

The *ixp software configure* process prepares the installed software on each LXC, *NS*, *RS*, *CS*, *BS* and *SC* for operation, all in accordance with the IXP schema structure.

```
cIXP:~$ ixp software configure all
```

For the *NS*, the process builds the Berkeley Internet Name Domain (BIND) version 9 (BIND9) daemon configuration files *named.conf.options*, the *named.conf.local* and the query log. It then creates the zone database files for the IPv4, IPv6 and domain as described within the IXP schema. After restarting the *NS* in order to bring the DNS Server online, IXPBuilder replaces the DNS server defined in each server *netplan* file with the IP address of the *NS* and reactivates *netplan* on each. DNS queries from each server are now processed by the *NS* and forwarding is handled by the upstream ISP DNS server. By default these are Google DNS servers 8.8.8.8 and 8.8.8.4.

The following example demonstrates this. The DNS on *RS* is shown to have changed to the IP address of *NS*, 198.8.8.231. A resolution request for *www.ubuntu.com* server resolves to the IP address 91.189.90.59.

```
root@rs1:~# systemd-resolve --status --no-pager | grep 'DNS Servers'
         DNS Servers: 198.8.8.231

root@rs1:~# dig www.ubuntu.com | grep -1 'ANSWER SECTION'

;; ANSWER SECTION:
www.ubuntu.com.         142    IN     A      91.189.90.59
```

With regard to *RS* and *CS* the BIRD configuration files */etc/bird/bird.conf* and */etc/bird/bird6.conf* are backed up and replaced with files configured as described in the IXP schema.

The application also generate filters to block *martians* or *bogons*. These are IP blocks that are reserved for special use. Examples are the RFC 1918 - *Address Allocation for Private Internets* and loopback networks. These special purpose blocks are documented in RFC 6890 - *Special-Purpose IP Address Registries* and updated in RFC 8190 - *Updates to the Special-Purpose IP Address Registries*. LXCs delivering the *RS* and *CS* functions filter routes received for any of these *martian* or *bogon* networks.

The *bird* and *bird6* daemons are restarted to activate.

The AS112 *BS* has both BIND9 and BIRD daemons installed and configured to handle reverse lookups for the following prefixes;

- 192.175.48.0/24,
- 192.31.196.0/24,
- 2620:4f:8000::/48,
- 2001:4:112::/48.

The BIRD and BIND9 configuration files */etc/bird/bird.conf* and */etc/bird/bird6.conf* are backed up and replaced with files configured as described in the IXP schema. This combination responds to reverse-lookups for Pointer records (PTR) associated with the subnets listed above.

The BIND9 Server on *BS* can be tested once full configuration of the IXP is completed as follows. The following networks 192.175.48.0/24, 192.31.196.0/24 are routed to the ISPs who choose to peer with the AS112 *BS* service.

```
cIXP:~$ ixp peer route -ip ipv4
  + ------------------------------------------------------------------------------------ +
  |                                The IPv4 Route Table                                  |
  + -------- + ---------------- + ------------ + ----------- + ----------- + -------- + ------ +
  |  Server  |      Address     |   Gateway    |  Interface  |   Member    |  LPref   |  ASN   |
  + -------- + ---------------- + ------------ + ----------- + ----------- + -------- + ------ +
  |   rs4    | 199.1.1.0/24     | 199.9.9.11   | ens3        | ISP one     |   100    |  5111  |
  |          | 199.3.3.0/24     | 199.9.9.33   | ens3        | ISP three   |   100    |  5333  |
  |          | 199.2.2.0/24     | 199.9.9.22   | ens3        | ISP two     |   100    |  5222  |
  + -------- + ---------------- + ------------ + ----------- + ----------- + -------- + ------ +
  |   cs4    | No routes        |              |             |             |          |        |
  + -------- + ---------------- + ------------ + ----------- + ----------- + -------- + ------ +
  |   bs4    | 192.175.48.0/24  | Local        | as112_br1   |             |          |        |
  |          | 192.31.196.0/24  | Local        | as112_br1   |             |          |        |
  + -------- + ---------------- + ------------ + ----------- + ----------- + -------- + ------ +
```

Considering the IP route table on a peering router, in this case from ISP 1. For IPv4 the route to the AS112 black-hole networks 192.175.48.0/24 and 192.31.196.0/24 can be seen in the IXP route table as well as on the ISP 1 router.

```
ISP1_RTR#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    199.9.9.0/24 is directly connected, FastEthernet0/0
B    192.31.196.0/24 [20/0] via 199.9.9.234, 00:56:29
     200.1.1.0/32 is subnetted, 1 subnets
C       200.1.1.1 is directly connected, Loopback0
B    199.3.3.0/24 [20/0] via 199.9.9.33, 00:55:53
B    199.2.2.0/24 [20/0] via 199.9.9.22, 00:55:21
C    199.1.1.0/24 is directly connected, FastEthernet0/1
B    192.175.48.0/24 [20/0] via 199.9.9.234, 00:56:29
```

In a similar way it can be seen that the networks 2620:4f:8000::/48 and 2001:4:112::/48 are routed to the ISPs who peer with the AS112 *BS* service.

```
cIXP:~$ ixp peer route -ip ipv6
  + ------------------------------------------------------------------------------------------ +
  |                                    The IPv6 Route Table                                    |
  + -------- + ------------------ + ------------- + ------------ + ---------- + -------- + ------ +
  |  Server  |       Address      |    Gateway    |  Interface   |   Member   |  LPref   |  ASN   |
  + -------- + ------------------ + ------------- + ------------ + ---------- + -------- + ------ +
  |   rs1    | 2a99:3:3::/48      | 2a99:9:9::33  | ens3         | ISP three  |   100    |  5333  |
  |          | 2a99:2:2::/48      | 2a99:9:9::22  | ens3         | ISP two    |   100    |  5222  |
  |          | 2a99:9:9::/48      | 2a99:9:9::11  | ens3         | ISP one    |   100    |  5111  |
  + -------- + ------------------ + ------------- + ------------ + ---------- + -------- + ------ +
  |   cs1    | No routes          |               |              |                       |        |
  + -------- + ------------------ + ------------- + ------------ + ---------- + -------- + ------ +
  |   bs1    | 2001:4:112::/48    | Local         | as112_br1    |            |          |        |
  |          | 2620:4f:8000::/48  | Local         | as112_br1    |            |          |        |
  + -------- + ------------------ + ------------- + ------------ + ---------- + -------- + ------ +
```

Reviewing the IPv6 route table on ISP 1's peering router. For IPv6 the AS112 black-hole prefixes 2620:4f:8000::/48 and 2001:4:112::/48 can be seen to have corresponding routes in the routing tables.

```
ISP1_RTR# show ipv6 route
IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
B   2001:4:112::/48 [20/0]
     via FE80::216:3EFF:FEFF:184F, FastEthernet0/0
B   2620:4F:8000::/48 [20/0]
     via FE80::216:3EFF:FEFF:184F, FastEthernet0/0
C   2A99:1:1::/48 [0/0]
     via ::, FastEthernet0/1
L   2A99:1:1::1/128 [0/0]
     via ::, FastEthernet0/1
B   2A99:2:2::/48 [20/0]
     via FE80::21E:BEFF:FE17:EB9A, FastEthernet0/0
B   2A99:3:3::/48 [20/0]
     via FE80::C671:FEFF:FE10:FE00, FastEthernet0/0
C   2A99:9:9::/48 [0/0]
     via ::, FastEthernet0/0
L   2A99:9:9::11/128 [0/0]
     via ::, FastEthernet0/0
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
```

The DNS Server on *BS* can be checked as follows:

```
root@bs1:~# dig @prisoner.iana.org hostname.as112.arpa txt +short +norec
"netLabs" "Kampala, Uganda"
"Unique IP: 199.9.9.234" "Unique IPv6: 2a99:9:9::234"
"See http://www.as112.net/ for more information."

root@bs1:~# dig @prisoner.iana.org hostname.as112.net txt +short +norec
"netLabs" "Kampala, Uganda"
"See http://www.as112.net/ for more information."
"Unique IP: 199.9.9.234" "Unique IPv6: 2a99:9:9::234"
```

The final function is the *SC* upon which the *python3-ryu* application is installed. Ryu is an Apache v2.0 licensed *SC* framework from the Nippon Telegraph and Telephone Corporation that can be developed for specific SDN tasks. This implementation of Ryu contains, the project developed, *ixp_switch_13* subclass. This handles the switching requirements of the IXP. *ixp_switch_13* is a subclass of the Ryu *app_manager.RyuApp*.

10. IXP SDN (`ixp sdn`)

The *software-defined* model gives a different, more logical view of the IXP. For example, while the OvS in the host is a *switch*, it is only the OvS peering bridge that SDN can visualise. This is because the *SC* is programmed specifically to access it via OF. Consider here the list of switches the *SC* can view. The first is the OvS peering LAN on the host and the second is the external OvS OF switch.

```
cIXP:~$ ixp sdn list switch

Switch Number: 0 (internal OvS)
-------------------------------
Switch ID     : 69131984840
Manufacturer  : Nicira, Inc.
Hardware Desc : Open vSwitch
Software Desc : 2.9.2
Serial number : None
DP Description: None
IP Address    : 198.8.8.230

Switch Number: 1 (External)
---------------------------
Switch ID     : 69133577248
Manufacturer  : Nicira, Inc.
Hardware Desc : Open vSwitch
Software Desc : 2.9.2
Serial number : None
DP Description: None
IP Address    : 198.8.8.241
```

Flows can be viewed for each OF switch, either individually or for all switches.

```
cIXP:~$ ixp sdn list flows --switch-number 0

Switch Number: 0 (internal OvS)
-------------------------------
P: 99 HT: 0 IT: 0 M: {'in_port': 1005}  A:
P: 99 HT: 0 IT: 0 M: {'in_port': 1006}  A:
P: 99 HT: 0 IT: 0 M: {'in_port': 1007}  A:
P: 99 HT: 0 IT: 0 M: {'in_port': 1008}  A:
P: 1 HT: 0 IT: 0 M: {'in_port': 2001, 'dl_src': '00:16:3e:dc:37:59',
'dl_dst': '00:16:3e:b8:70:9c'}  A: OUTPUT:2003
P: 1 HT: 0 IT: 0 M: {'in_port': 2003, 'dl_src': '00:16:3e:b8:70:9c',
'dl_dst': '00:16:3e:dc:37:59'}  A: OUTPUT:2001
P: 0 HT: 0 IT: 0 M: {}  A: OUTPUT:CONTROLLER

P - Priority, H - Hard timeout, I - Idle timeout, M - Match, A - Actions
```

A number of non command related *software-defined* features happen in the background. For example, interfaces on switches that are not configured by the `ixp peer` command are automatically configured by the *SC* to drop packets at the point of ingress until such time as the interface is configured. This feature offers an glimpse of the potential for additional functionality that a programmable *SC* can offer over *traditional* switching.

11. IXP SDN Controller RESTful API

IXPBuilder offers the Ryu RESTful Application Programming Interface (API). Data is presented in JavaScript Object Notation (JSON) format. The first step is to generate a Uniform Resource Locator (URL) as follows to formulate a request to the RESTful API and access the JSON output.

```
http://<SC>.<domain>:8080/stats/switches
http://sc1.netlabs.tst:8080/stats/switches
```

The initial step acquires a list of switches that the *SC* can view via OF is demonstrated in Illustration 10.

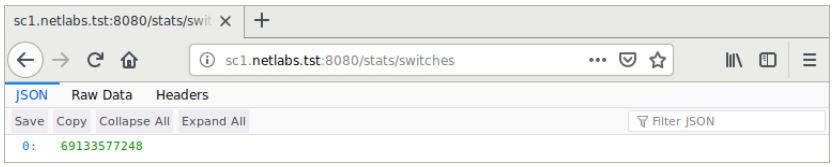


Illustration 10: Ryu get all switches

Armed with this switch list more information can be obtained about each individual switch in the list as shown in Illustration 11, this is similar to the output from the `ixp sdn show switches` command.

Illustration 11: Ryu get switch description

A flow table can be generated for the OF switch as can be seen in Illustration 12, this is similar to the output from the `ixp sdn show flows` command.

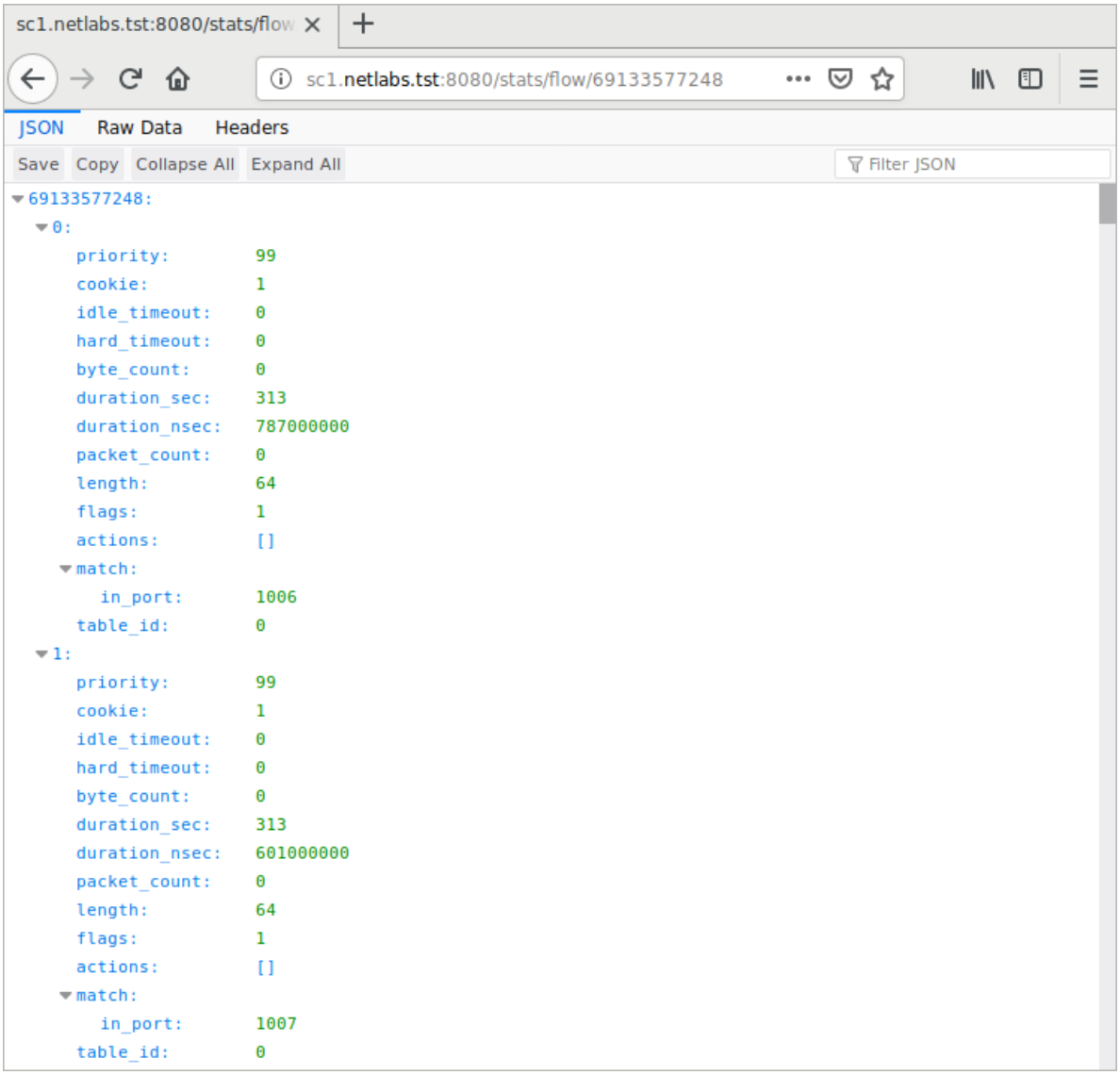


Illustration 12: Ryu switch flow table

The RESTful API exposes a North Bound Interface (NBI) that can be used for automation and monitoring. It returns data in simple JSON format as shown and this can be used to develop further tools that can enhance the functionality of the *software-defined* IXP. Here is an example accessing the RESTful API using the GNU/Linux cURL, transfer a URL, command.

```
ubuntu@mIXP:~$ curl http://sc1.netlabs.tst:8080/stats/desc/69133577248; echo
{"69133577248": {"mfr_desc": "Nicira, Inc.", "hw_desc": "Open vSwitch",
"sw_desc": "2.9.2", "serial_num": "None", "dp_desc": "None"}}
```

12. The IXP Peers (`ixp peer`)

IXP memberships have traditionally been the preserve of Internet Service Providers (ISP); however, in recent years this has changed where membership now includes specialist Internet Content Providers (ICP), Application Service Providers (ASP), banks, Regional Education Networks (REN), government entities to name but a few. What is common to all the peering members is they must have an ASN assigned by the Regional Internet Registry (RIR), AfriNIC in the case of Africa for example.

These members can be added to the BIRD routing tables in the *RS*, *CS* and *BS* functions. Connection to the *CS* function is compulsory whereas members can decide if they wish to propagate routes with peers at the *RS* and use the AS112 service on the *BS*, depending on their peering inclination and desire to have the IXP provide an AS112 service.

```
cIXP:~$ ixp peer add ?
Usage: ixp peer add { help | ? } [ARGUMENT(s)] [VALUE]

        ?, help                 -  This help message.

        ARGUMENT(s) [list]:
        -n, --name          (M)  - Remote Autonomous System Name.
        -a, --as-number     (M)  - Remote Autonomous System Number.
        -d, --domain        (M)  - Peering member domain name.
        -rs, --route-server      - Include RS for this peer.
        -bs, --blackhole-server  - Include BS for this peer.
        -is, --interface-speed   - Interface speed required for peer.

        (M) mandatory
```

Consider the example of ISP 2 being added. In this case the ISP was physically connected to a peer interface on the IXP server itself and is being switched by the internal OvS peering LAN bridge.

```
cIXP:~$ ixp peer add -n 'ISP 2' -a 5222 -d two.com

'ISP 2' assigned ipv4 peer address 199.9.9.2/24 on switch 0, port 4

'ISP 2' assigned ipv6 peer address 2a99:9:9::2/48 on switch 0, port 4
```

Compare this to the example of ISP 12 being added to the system. This ISP has been added and assigned interface 10 on the first external switch. In this case the ISP has a peering inclination with a predisposition against peering but wants to connect to the AS112 service. Additionally this peer is connected to a 10G interface.

```
cIXP:~$ ixp peer add -n 'ISP 12' -a 51212 -d twelve.com -rs no -bs yes -is 10G

'ISP 12' assigned ipv4 peer address 199.9.9.13/24 on switch 1, port 10

'ISP 12' assigned ipv6 peer address 2a99:9:9::13/48 on switch 1, port 10
```

The peer neighbour lists and routing tables of the *RS*, *CS* and *BS* functions can be reviewed as demonstrated below. This demonstrates that all five ISPs have BGP *Established* connections with the *CS* but have connections with the *RS* or AS112 *BS* only if configured to do so. If a BGP state of *Connect*, *Active* or *Idle* are shown, these are different phases of

the BGP state machine and are waiting for the connection over the Transmission Control Protocol (TCP) port 179 connection to complete with the ISP router.

```
cIXP:~$ ixp peer show --ip-type ipv6
```

IPv6 Peering Table

Name	ASN	Switch	Port	Speed	Domain	RS	AS112	IP Address
as5111	5111	0	5	1G	one.com	yes	yes	2a99:9:9::2/48
as5222	5222	0	6	1G	two.com	yes	yes	2a99:9:9::3/48
as5333	5333	0	7	10G	three.com	no	no	2a99:9:9::4/48
as5444	5444	0	8	100M	four.com	no	yes	2a99:9:9::5/48
as5555	5555	1	3	40G	five.com	yes	no	2a99:9:9::6/48

```
cIXP:~$ ixp peer status --ip-type ipv4
```

IPv4 BGP State Table

Server	Name	State	Info
rs1	as5111	up	Established
	as5222	up	Established
	as5555	up	Established
cs1	as5111	up	Established
	as5222	up	Established
	as5333	up	Established
	as5444	up	Established
	as5555	up	Established
bs1	as5111	up	Established
	as5222	up	Established
	as5444	up	Established

The routes table on the *RS*, *CS* and *BS* functions can also be retrieved from the BIRD daemon.

```
cIXP:~$ ixp peer route -ip ipv4
```

IPv4 Route Table

Server	Address	Gateway	Interface	Member	LPref	ASN
rs1	192.175.48.0/24	199.9.9.234	ens3	AS5111	99.9.9.2	
	192.31.196.0/24	199.9.9.234	ens3	AS5111	99.9.9.2	
	199.1.1.0/24	199.9.9.2	ens3	AS5111	100	5111
	199.2.2.0/24	199.9.9.3	ens3	AS5222	100	5222
	199.5.5.0/24	199.9.9.6	ens3	AS5555	100	5555
cs1	192.175.48.0/24	199.9.9.234	ens3	AS5111	99.9.9.2	
	192.31.196.0/24	199.9.9.234	ens3	AS5111	99.9.9.2	
	199.1.1.0/24	199.9.9.2	ens3	AS5111	100	5111
	199.2.2.0/24	199.9.9.3	ens3	AS5222	100	5222
	199.3.3.0/24	199.9.9.4	ens3	AS5333	100	5333
	199.4.4.0/24	199.9.9.5	ens3	AS5444	100	5444
	199.5.5.0/24	199.9.9.6	ens3	AS5555	100	5555
bs1	192.175.48.0/24	Local	as112_br1			
	192.31.196.0/24	Local	as112_br1			
	199.1.1.0/24	199.9.9.2	ens3	AS5111	100	5111
	199.2.2.0/24	199.9.9.3	ens3	AS5222	100	5222
	199.4.4.0/24	199.9.9.5	ens3	AS5444	100	5444

13. IXP Router RESTful API

IXPBuilder also implements the *Birdseye* RESTful API for the BIRD routing daemon on the *RS*, *CS* and *BS* functions. The *Birdseye* API provides access to some BIRD protocol routing information via a Hypertext Transfer Protocol (HTTP) API, formatted as JSON, via a *lighttpd* webserver. This service is complimentary to the BIRD routing daemon.

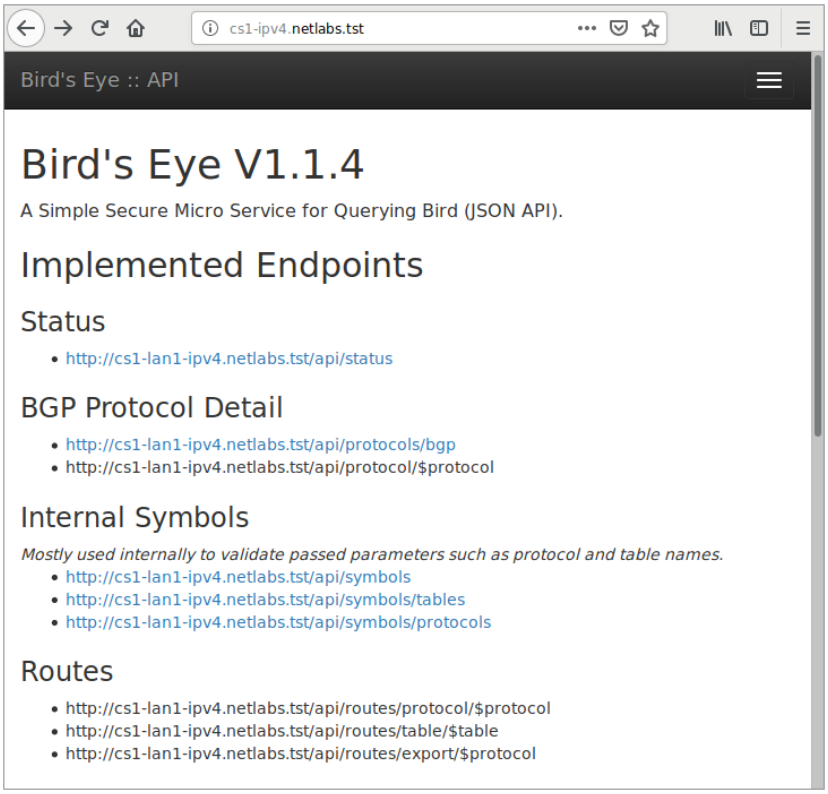


Illustration 13: Birdseye control panel

Birdseye gives an information panel as shown in Illustration 13 that has some links giving URLs to information pages. It is accessed using the URL:

```
http://<container RS, CS or BS>-<ipv4|ipv6>.<domain>
http://cs1-ipv6.netlabs.tst
```

The simplest of these as demonstrated in Illustration 14 returns the status for the specific BIRD daemon that was called. There are a number of other links like that showing the status of the BGP protocol as in Error: Reference source not found. This report is similar to the information received from the `ixp peer status` command. Finally there are a number of URLs in black text, these have a variable like `$protocol` in the URL which the user is expected to transpose with a replacement. For example in Illustration 16 `$protocol` is replaced by `ISP2` (which was one of the protocols shown in the previous illustration). This page gives similar output as received by the `ixp peer route` command.

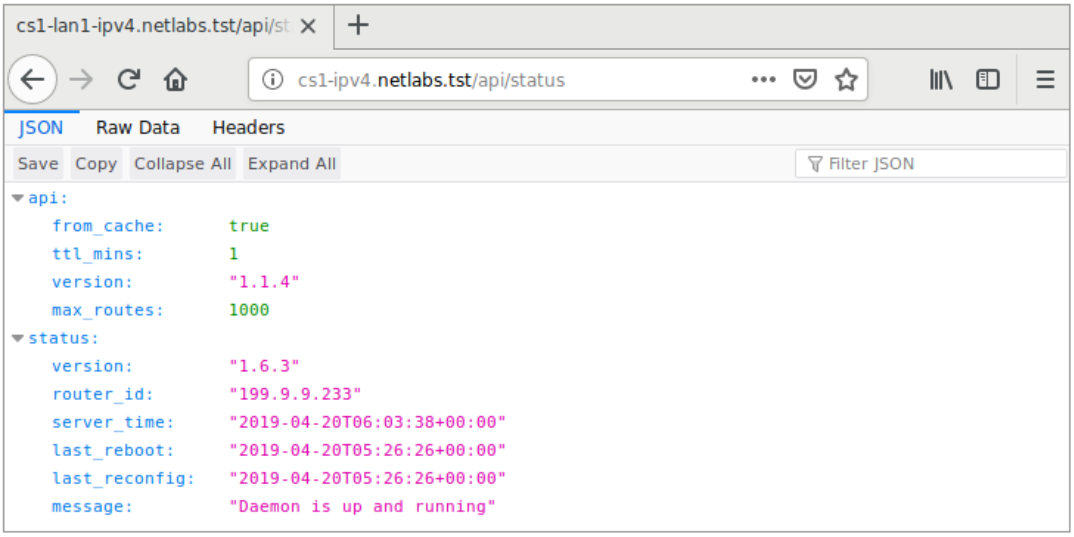


Illustration 14: Birdseye reporting BIRD status

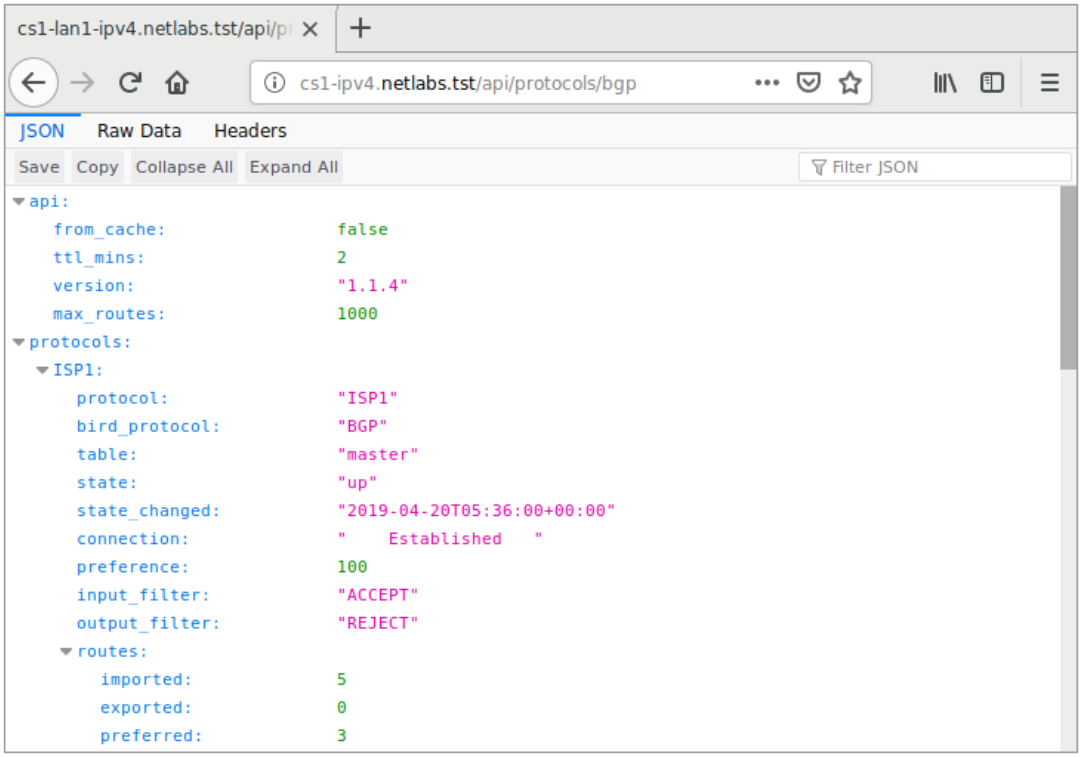


Illustration 15: Birdseye reporting the status of the BIRD BGP protocol

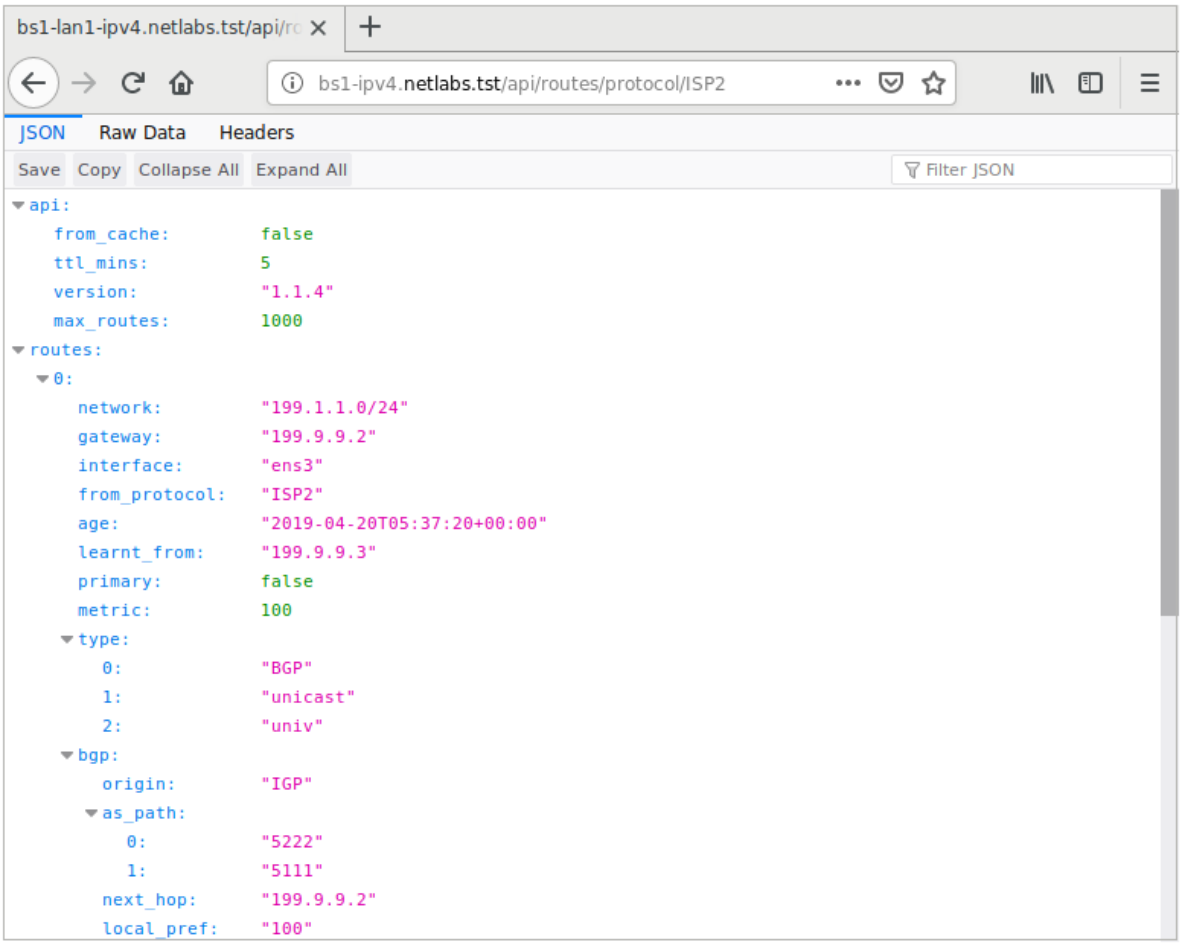


Illustration 16: Birdseye reporting the BIRD routes for a specific protocol

The *Birdseye* RESTful API is a useful tool for automation and monitoring. It returns data in simple JSON format as shown and this can be used to develop further tools that can enhance the IXP.

```
ubuntu@mIXP:~$ curl http://cs1-ipv4.netlabs.tst/api/status; echo
{"api":
{"from_cache":true,"ttl_mins":1,"version":"1.1.4","max_routes":1000},"status":
{"version":"1.6.3","router_id":"199.9.9.233","server_time":"2019-04-
20T07:00:22+00:00","last_reboot":"2019-04-
20T05:26:26+00:00","last_reconfig":"2019-04-20T05:26:26+00:00","message":"Daemon
is up and running"}}
```

14. IXP Remote (`ixp remote`)

14.1. Introduction

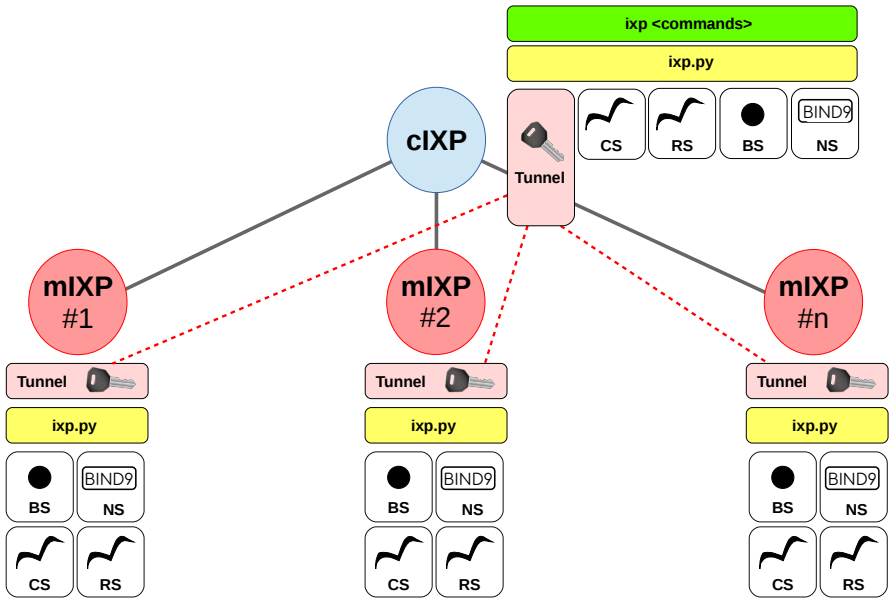


Illustration 17: dIXP diagram

The *ixp remote* process allows for the day-to-day management of mIXP's from the cIXP. Consider Illustration 17, the dIXP consists of the cIXP and three mIXPs. A Rivest, Shamir, Adleman (RSA) key pair is generated with the private key remaining at the cIXP and the public key shared with the mIXPs. This allows IXPBuilder to establish tunnels over the Secure Shell (SSH) port for the management of the remote mIXPs.

14.2. Setup a mIXP

Initial configuration of an mIXP is similar to that of a cIXP except the mIXP `--site-type` will be specified as *mini* as demonstrated in Illustration 18.

In order to suit the particular site the `--switching-type` and the model chosen at the mIXP can be different from the cIXP.

To establish an mIXP, complete the initial installation steps, the schema, host, switch, server and software processes as described earlier in this document.

```
mIXP:~$ ixp schema build                \
  --site-number 2                       \
  --site-type mini                      \
  --switching-type software-defined     \
  --ipv4-peer 177.7.7.0/24              \
  --ipv6-peer 2a77:7:7::/48             \
  --ipv4-man 176.6.6.0/24               \
  --ipv6-man 2a76:6:6::/48              \
  --as-number 7999                      \
  --country Uganda                      \
  --town Gulu                           \
  --elevation 1100                      \
  --latitude '02 46 40.2314 N'          \
  --longitude '32 17 14.1243 E'
```

Illustration 18: mIXP schema

14.3. Generate tunnel key pair

On the cIXP a private/public key pair is generated to protect management traffic passing from the cIXP to the mIXPs. The following command generates the pair and once created the public key can be accessed.

```
ubuntu@cIXP:~$ ixp remote key generate
Generated encryption keys for core IXP (cIXP)

ubuntu@cIXP:~$ ixp remote key show
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCh6cka7cGu0UzgnG/aHJHmgqK5mKFDF/RFoHQlaT0GpAqqZU3pF
pIumCFQKojmfcOKFYHuB0qxG3JpRDxY/
BOGTB9tX0cZmZWm74jfb6a+qLFjcG3X3utZgDmTb17Oj+o2Hg5fTqFarsAzb7LWI/+/
vjx6xTkxrWqbb0vs+lUTpfdaQUie9yc4tQuL/
UrmEFDS+sboQD1nrs1unSM2hsnyVptLaLRpsAaH99fvEpjkaLbzECRSK2+nHCkCIEOMOTElI8/
rupPIBqHgUruP7VVH/yMbFyo6hhk26AwGiN3p/r68uTUcEUD47QanU97i8J6/WId4RSQzo5WYILmky3GT
```

Copy the key as presented on the cIXP verbatim and enter it on the mIXP. The next two commands demonstrate entering the new key on an mIXP and confirming it with the show command.

```
ubuntu@mIXP:~$ ixp remote key enter

  Paste the public key test here: ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCh6cka7cGu0UzgnG/aHJHmgqK5mKFDF/RFoHQlaT0GpAqqZU3pF
pIumCFQKojmfcOKFYHuB0qxG3JpRDxY/
BOGTB9tX0cZmZWm74jfb6a+qLFjcG3X3utZgDmTb17Oj+o2Hg5fTqFarsAzb7LWI/+/
vjx6xTkxrWqbb0vs+lUTpfdaQUie9yc4tQuL/
UrmEFDS+sboQD1nrs1unSM2hsnyVptLaLRpsAaH99fvEpjkaLbzECRSK2+nHCkCIEOMOTElI8/
rupPIBqHgUruP7VVH/yMbFyo6hhk26AwGiN3p/r68uTUcEUD47QanU97i8J6/WId4RSQzo5WYILmky3GT
Added public key to mini IXP (mIXP)

ubuntu@mIXP:~$ ixp remote key show
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCh6cka7cGu0UzgnG/aHJHmgqK5mKFDF/RFoHQlaT0GpAqqZU3pF
pIumCFQKojmfcOKFYHuB0qxG3JpRDxY/
BOGTB9tX0cZmZWm74jfb6a+qLFjcG3X3utZgDmTb17Oj+o2Hg5fTqFarsAzb7LWI/+/
vjx6xTkxrWqbb0vs+lUTpfdaQUie9yc4tQuL/
UrmEFDS+sboQD1nrs1unSM2hsnyVptLaLRpsAaH99fvEpjkaLbzECRSK2+nHCkCIEOMOTElI8/
rupPIBqHgUruP7VVH/yMbFyo6hhk26AwGiN3p/r68uTUcEUD47QanU97i8J6/WId4RSQzo5WYILmky3GT
```

14.4. Confirm connectivity

The next step is to have the mIXP physically installed at the remote site and confirm connectivity to the management LAN port on it from the cIXP. It is also advisable to establish an alternative connection such as mobile 4G Long Term Evolution (LTE) or 5G New Radio (NR) to the OOB interface in order to ensure connectivity in case of an outage over the management LAN.

Even if there is a connectivity failure to the management LAN, it is not necessary that there is also disruption to the peering function as this service is local in nature.

Note: This step may involve adding default routes or employing a routing protocol depending on the specifics of the Internet connectivity between management LANs. To create persistent routes append them to the *ovs-config-start.sh* script on the server.

cIXP

```
ubuntu@cIXP:~$ echo 'ip route add 176.6.6.0/24 via 198.8.8.1' >>
/srv/ovs/ovs-config_start.sh

ubuntu@cIXP:~$ systemctl restart ovs-config.service

ubuntu@cIXP:~$ fping 176.6.6.230
176.6.6.230 is alive
```

mIXP

```
ubuntu@mIXP:~$ echo 'ip route add 198.8.8.0/24 via 176.6.6.1' >>
/srv/ovs/ovs-config_start.sh

ubuntu@mIXP:~$ systemctl restart ovs-config.service

ubuntu@mIXP:~$ fping 198.8.8.230
198.8.8.230 is alive
```

The remote mIXP site is added to the cIXP. By default the local user on the cIXP is assumed to be the same as the local user on the mIXP; however, if not, this can be changed with the `--user` option.

```
ubuntu@cIXP:~$ ixp remote site add --site-number 2                \
                                   --site-name Gulu               \
                                   --location 'Main Street, Gulu' \
                                   --host-v4 176.6.6.230          \
                                   --host-v6 2a76:6:6::230

Loaded site 2 data to the database


ubuntu@cIXP:~$ ixp remote site show
  + ------------------------------------------------------------------------------------------------ +
  |                                        Remote mIXP sites                                         |
  + -------- + ----------- + ------------------- + ------------- + ------------- + ---------- + ------ +
  | Site No. |  Site Name  |      Location       |     IPv4      |     IPv6      |    User    |  Port  |
  + -------- + ----------- + ------------------- + ------------- + ------------- + ---------- + ------ +
  |    2     |    Gulu     |  Main Street, Gulu  |  176.6.6.230  | 2a76:6:6::230 |   ubuntu   |   22   |
  + -------- + ----------- + ------------------- + ------------- + ------------- + ---------- + ------ +
```

14.5. Accessing the mIXP from the cIXP for day to day management

It is necessary that remote access is available from the cIXP to the *show* and *list* commands, for all classes, of each mIXP to allow the IXP administrator ascertain their configuration and carry out remote troubleshooting activities. Full access to the *ixp peer* process is also available to facilitate integrated day-to-day management of the mIXP from the cIXP location. Here are a number of example activities that are carried out at the cIXP but applied the the mIXP, 2 in this case.

```
ubuntu@cIXP:~$ ixp remote cmd 2 -- ixp host show

## Executing on remote mIXP site: 2

  [ mIXP: 2 ]~$ ixp host show

  IXP Host interfaces - Model 'S'
 + ----------------------------- +
 |        Host interfaces        |
 + ------------- + ------------- +
 |   Interface   |   Function    |
 + ------------- + ------------- +
 | eth1          | OOB           | 192.168.234.202/24, 2a98:8:8:0:221:9bff:fea1:f5fe/64
 | eth2          | Management    |
 | eth3          | Local peer    | 177.7.7.2/24, 2a77:7:7::2/48
 | eth4          | Local peer    | 177.7.7.3/24, 2a77:7:7::3/48
 | eth5          | Local peer    | 177.7.7.4/24, 2a77:7:7::4/48
 | eth6          | Local peer    | 177.7.7.5/24, 2a77:7:7::5/48
 | eth7          | Local peer    | 177.7.7.6/24, 2a77:7:7::6/48
 | eth8          | Local peer    | 177.7.7.7/24, 2a77:7:7::7/48
 + ------------- + ------------- +

ubuntu@cIXP:~$ ixp remote cmd 2 -- ixp peer show

## Executing on remote mIXP site: 2

  [ mIXP: 2 ]~$ ixp peer show

  + ------------------------------------------------------------------------------------- +
  |                                    IPv4 Peering Table                                 |
  + ------ + ------ + -------- + ------ + ----- + --------- + ----- + ----- + ------------ +
  |  Name  |  ASN   |  Switch  |  Port  | speed |   Domain  |  RS   | AS112 |  IP Address  |
  + ------ + ------ + -------- + ------ + ----- + --------- + ----- + ----- + ------------ +
  | as7111 |  7111  |    0     |   3    |  1G   |  one.net  |  yes  |  yes  | 177.7.7.2/24 |
  | as7222 |  7222  |    0     |   4    |  1G   |  two.net  |  yes  |  yes  | 177.7.7.3/24 |
  + ------ + ------ + -------- + ------ + ----- + --------- + ----- + ----- + ------------ +

  + ------------------------------------------------------------------------------------- +
  |                                    IPv6 Peering Table                                 |
  + ------ + ------ + -------- + ------ + ----- + --------- + ----- + ----- + ------------- +
  |  Name  |  ASN   |  Switch  |  Port  | speed |   Domain  |  RS   | AS112 |  IP Address   |
  + ------ + ------ + -------- + ------ + ----- + --------- + ----- + ----- + ------------- +
  | as7111 |  7111  |    0     |   3    |  1G   |  one.net  |  yes  |  yes  | 2a77:7:7::2/48 |
  | as7222 |  7222  |    0     |   4    |  1G   |  two.net  |  yes  |  yes  | 2a77:7:7::3/48 |
  + ------ + ------ + -------- + ------ + ----- + --------- + ----- + ----- + ------------- +
```

This is an example addition of a new peer at the remote mIXP created by the IXP administrator at the cIXP location.

```
ubuntu@cIXP:~$ ixp remote cmd 2 -- ixp peer add -n as7333 -a 7333 -d three.net

## Executing on remote mIXP site: 2

  [ mIXP: 2 ]~$ ixp peer add --name as7333 --as-number 7333 --domain three.net

'as7333' assigned ipv4 peer address 177.7.7.4/24 on switch 0, port 5

'as7333' assigned ipv6 peer address 2a77:7:7::4/48 on switch 0, port 5
```

```
ubuntu@cIXP:~$ ixp remote cmd 2 -- ixp peer show

## Executing on remote mIXP site: 2

  [ mIXP: 2 ]~$ ixp peer show

  + ------------------------------------------------------------------------------------ +
  |                                    IPv4 Peering Table                                |
  + ------ + ------ + -------- + ------ + ----- + --------- + --- + ----- + ------------ +
  |  Name  |  ASN   |  Switch  |  Port  | speed |  Domain   | RS  | AS112 |  IP Address  |
  + ------ + ------ + -------- + ------ + ----- + --------- + --- + ----- + ------------ +
  | as7111 |  7111  |    0     |   3    |  1G   | one.net   | yes |  yes  | 177.7.7.2/24 |
  | as7222 |  7222  |    0     |   4    |  1G   | two.net   | yes |  yes  | 177.7.7.3/24 |
  | as7222 |  7222  |    0     |   5    |  1G   | three.net | yes |  yes  | 177.7.7.3/24 |
  + ------ + ------ + -------- + ------ + ----- + --------- + --- + ----- + ------------ +

  + ------------------------------------------------------------------------------------ +
  |                                    IPv6 Peering Table                                |
  + ------ + ------ + -------- + ------ + ----- + --------- + --- + ----- + ------------ +
  |  Name  |  ASN   |  Switch  |  Port  | speed |  Domain   | RS  | AS112 |  IP Address  |
  + ------ + ------ + -------- + ------ + ----- + --------- + --- + ----- + ------------ +
  | as7111 |  7111  |    0     |   3    |  1G   | one.net   | yes |  yes  | 2a77:7:7::2/48 |
  | as7222 |  7222  |    0     |   4    |  1G   | two.net   | yes |  yes  | 2a77:7:7::3/48 |
  | as7333 |  7333  |    0     |   5    |  1G   | three.net | yes |  yes  | 177.7.7.3/24   |
  + ------ + ------ + -------- + ------ + ----- + --------- + --- + ----- + ------------ +

ubuntu@cIXP:~$ ixp remote cmd 2 -- ixp peer status --ip-type ipv6

## Executing on remote mIXP site: 2

  [ mIXP: 2 ]~$ ixp peer status -ip ipv6

  + ------------------------------------------- +
  |            IPv6 BGP State Table             |
  + -------- + -------- + ------- + ------------ +
  |  Server  |   Name   |  State  |     Info     |
  + -------- + -------- + ------- + ------------ +
  |   rs2    | as7111   | up      | Established  |
  |          | as7222   | up      | Established  |
  |          | as7333   | up      | Established  |
  + -------- + -------- + ------- + ------------ +
  |   cs2    | as7111   | up      | Established  |
  |          | as7222   | up      | Established  |
  |          | as7333   | up      | Established  |
  + -------- + -------- + ------- + ------------ +
  |   bs2    | as7111   | up      | Established  |
  |          | as7222   | up      | Established  |
  |          | as7333   | up      | Established  |
  + -------- + -------- + ------- + ------------ +
```

```
ubuntu@cIXP:~$ ixp remote cmd 2 -- ixp peer route -ip ipv4

## Executing on remote mIXP site: 2

  [ mIXP: 2 ]~$ ixp peer route --ip-type ipv4
```

IPv4 Route Table

Server	Address	Gateway	Interface	Member	LPref	ASN
rs2	192.175.48.0/24	177.7.7.234	ens3	as7111	77.7.7.2	
	192.31.196.0/24	177.7.7.234	ens3	as7111	77.7.7.2	
	177.1.1.0/24	177.7.7.2	ens3	as7111	100	7111
	177.2.2.0/24	177.7.7.3	ens3	as7222	100	7222
	177.3.3.0/24	177.7.7.4	ens3	as7333	100	7333
cs2	192.175.48.0/24	177.7.7.234	ens3	as7111	77.7.7.2	
	192.31.196.0/24	177.7.7.234	ens3	as7111	77.7.7.2	
	177.1.1.0/24	177.7.7.2	ens3	as7111	100	7111
	177.2.2.0/24	177.7.7.3	ens3	as7222	100	7222
	177.3.3.0/24	177.7.7.4	ens3	as7333	100	7333
bs2	192.175.48.0/24	Local	as112_br1			
	192.31.196.0/24	Local	as112_br1			
	177.1.1.0/24	177.7.7.2	ens3	as7111	100	7111
	177.2.2.0/24	177.7.7.3	ens3	as7222	100	7222
	177.3.3.0/24	177.7.7.4	ens3	as7333	100	7333

15. The External IXP Switch

This section of the document considers the configuration of external switches which may be connected to the IXPBuilder server.

For *traditional* IXP models, an IXP switch is demonstrated in Illustration 19 which has 24 interfaces. The first interface is configured as a VLAN trunk connected to an Ethernet interface on the core server which is a corresponding VLAN trunk. The second interface is an untagged interface connected to the management LAN while all the other interfaces on a switch are VLAN access interfaces to the peering LAN. The remainder of this section gives example configurations for Cisco, Netgear, Juniper and MikroTik switches that demonstrate the steps to prepare an external Ethernet switch to connect to IXPBuilder server.

15.1. Traditional switching configuration

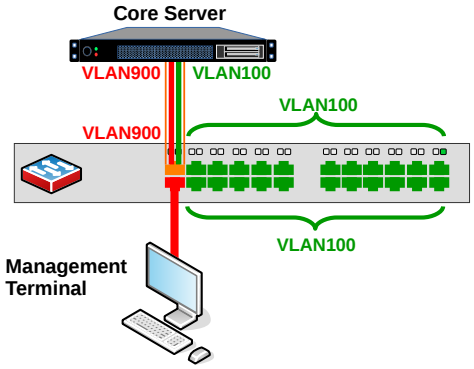


Illustration 19: The traditional IXP Switch

15.1.1) Cisco Catalyst Switch

Default Username: <*no username*> Password: <*no password*>

Configure the switch hostname.

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# hostname IXP_SW
```

Configure the peering VLANs 100 and 900.

```
IXP_SW(config)# vlan 100
IXP_SW(config-vlan)# name VLAN100-Peering

IXP_SW(config-vlan)# vlan 900
IXP_SW(config-vlan)# name VLAN900-Management
IXP_SW(config-vlan)# exit
```

Add the first interface as a VLAN trunk, this interface faces the IXP Server.

```
IXP_SW(config)# interface GigabitEthernet 1/0/1
IXP_SW(config-if)# description "VLAN Trunk interface to lxd1"
IXP_SW(config-if)# switchport trunk encapsulation dot1q
IXP_SW(config-if)# switchport mode trunk
IXP_SW(config-if)# switchport trunk allowed vlan add 100,900
IXP_SW(config-if)# no shutdown
IXP_SW(config-if)# exit
```

Configure the management VLAN 900 and add an IP address to the VLAN for the purpose of management access. Place the second interface in VLAN 900 as an untagged interface.

```
IXP_SW(config)# interface vlan 900
IXP_SW(config-if)# ip address 198.8.8.241 255.255.255.0
IXP_SW(config-if)# no shutdown
IXP_SW(config-if)# exit
IXP_SW(config)# ip default-gateway 198.8.8.1

IXP_SW(config)# interface GigabitEthernet 1/0/2
IXP_SW(config-if)# description "Management interface in VLAN 900"
IXP_SW(config-if)# switchport mode access
IXP_SW(config-if)# switchport access vlan 900
IXP_SW(config-if)# no shutdown
IXP_SW(config-if)# exit
```

Add the interfaces from 3 to 24 untagged to VLAN100. Only interfaces 3 and 24 are demonstrated here.

```
IXP_SW(config)# interface GigabitEthernet 1/0/3
IXP_SW(config-if)# description "<<ISP 1>> VLAN 100"
IXP_SW(config-if)# switchport mode access
IXP_SW(config-if)# switchport access vlan 100
IXP_SW(config-if)# spanning-tree portfast
IXP_SW(config-if)# no shutdown
IXP_SW(config-if)# exit
...

IXP_SW(config)# interface GigabitEthernet 1/0/48
IXP_SW(config-if)# description "<<ISP 46>> VLAN 100"
IXP_SW(config-if)# switchport mode access
IXP_SW(config-if)# switchport access vlan 100
IXP_SW(config-if)# spanning-tree portfast
IXP_SW(config-if)# no shutdown
IXP_SW(config-if)# exit
```

Save the configuration.

```
IXP_SW(config-if)# exit
IXP_SW(config)# exit
IXP_SW# copy running-config startup-config
```

15.1.2) Netgear ProSAFE Switch

Default User: *admin* Password: *<no password>*

Configure the console to 9,600 baud (to match the other devices).

```
(M4300-28G)> enable
(M4300-28G)> configure
(M4300-28G)(Config)# line console
(M4300-28G)(Config-line)# serial baudrate 9600
```

Configure the switch hostname.

```
(M4300-28G)# hostname IXP_SW
```

Configure the peering VLANs 100 and 900.

```
(IXP_SW)# vlan database
(IXP_SW)(Vlan)# vlan 100
(IXP_SW)(Vlan)# vlan name 100 VLAN100-Peering
(IXP_SW)(Vlan)# vlan 900
(IXP_SW)(Vlan)# vlan name 900 VLAN900-Management
```

Add the first interface as a VLAN trunk, this interface faces the IXP Server.

```
(IXP_SW)# configure
(IXP_SW)(Config)# interface 1/0/1
(IXP_SW)(Interface 1/0/1)# description "VLAN Trunk interface to lxd1"
(IXP_SW)(Interface 1/0/1)# switchport mode trunk
(IXP_SW)(Interface 1/0/1)# no shutdown
(IXP_SW)(Interface 1/0/1)# exit
```

Configure the management VLAN 900 and add an IP address to the VLAN for the purpose of management access. Place the second interface in VLAN 900 as an untagged interface.

```
(IXP_SW)(Config)# ip management vlan 900 198.8.8.241 255.255.255.0
(IXP_SW)(Config)# ip default-gateway 198.8.8.1

(IXP_SW)(Config)# interface 1/0/2
(IXP_SW)(Interface 1/0/2)# description "Management interface in VLAN 900"
(IXP_SW)(Interface 1/0/2)# switchport mode access
(IXP_SW)(Interface 1/0/2)# switchport access vlan 900
(IXP_SW)(Interface 1/0/2)# no shutdown
(IXP_SW)(Interface 1/0/2)# exit
```

Add the interfaces from 3 to 24 untagged to VLAN 100. Only interfaces 3 and 24 are demonstrated here.

```
(IXP_SW)(Config)# interface 1/0/3
(IXP_SW)(Interface 1/0/3)# description "<<ISP 1>> VLAN 100"
(IXP_SW)(Interface 1/0/3)# switchport mode access
(IXP_SW)(Interface 1/0/3)# switchport access vlan 100
(IXP_SW)(Interface 1/0/3)# spanning-tree edgeport
(IXP_SW)(Interface 1/0/3)# no shutdown
(IXP_SW)(Interface 1/0/3)# exit
...
```

```
(IXP_SW)(Config)# interface 1/0/24
(IXP_SW)(Interface 1/0/24)# description "<<ISP 22>> VLAN 100"
(IXP_SW)(Interface 1/0/24)# switchport mode access
(IXP_SW)(Interface 1/0/24)# switchport access vlan 100
(IXP_SW)(Interface 1/0/24)# spanning-tree edgeport
(IXP_SW)(Interface 1/0/24)# no shutdown
(IXP_SW)(Interface 1/0/24)# exit
```

Save the configuration.

```
(IXP_SW)(Config)# exit
(IXP_SW)# save
This operation may take a few minutes.
Management interfaces will not be available during this time.
Are you sure you want to save? (y/n) y
Config file 'startup-config' created successfully.
Configuration Saved!
```

15.1.3) Juniper EX series Switch

Default Login: *root* Password: *<no password>*

Configure the switch hostname.

```
root@:RE:0% cli
root> configure
root# set system host-name IXP_SW
```

Configure the peering VLANs 100 and 900.

```
root@IXP_SW# set vlans vlan-100 vlan-id 100 description "Peering VLAN"
root@IXP_SW# set vlans vlan-900 vlan-id 900 description "Management VLAN"
```

Add the first interface (ge-0/0/0) as a VLAN trunk, this interface faces the IXP Server.

```
root@IXP_SW# set interfaces ge-0/0/0 unit 0 family ethernet-switching
interface-mode trunk
```

Add the second interface (ge-0/0/1) to VLAN 900.

```
root@IXP_SW# delete interfaces ge-0/0/1 unit 0 family ethernet-switching
vlan members default
root@IXP_SW# set interfaces ge-0/0/1 unit 0 family ethernet-switching
vlan members vlan-900
```

Configure the management VLAN 900 and add an IP address to the VLAN for the purpose of management access. Place the second interface (ge-0/0/1) in VLAN 900 as an untagged interface.

```
root@IXP_SW# set interfaces irb.900 family inet address 198.8.8.241/24
root@IXP_SW# set vlans vlan-900 l3-interface irb.900
root@IXP_SW# set routing-options static route 0.0.0.0/0 next-hop
198.8.8.1
```

Add the interfaces from interface 3 (ge-0/0/2) to 24 (ge-0/0/23) untagged to VLAN 100. Only interfaces 3 (ge-0/0/2) and 24 (ge-0/0/23) are demonstrated here.

```
root@IXP_SW# delete interfaces ge-0/0/2 unit 0 family ethernet-switching
vlan members default
root@A01# set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan
members vlan-100

...

root@IXP_SW# delete interfaces ge-0/0/23 unit 0 family ethernet-switching
vlan members default
root@A01# set interfaces ge-0/0/23 unit 0 family ethernet-switching vlan
members vlan-100
```

Commit the configuration.

```
root@IXP_SW# commit
```

15.1.4) MikroTik Switch

Default Username: *admin* Password: <*no password*>

Configure the switch hostname.

```
/system identity set name=IXP_SW
```

Configure the peering VLANs 100 and 900. Add the first interface as a trunk interface connected to the IXP server.

```
/interface vlan
add name=ether1-vlan900 vlan-id=900 interface=ether1
add name=ether1-vlan100 vlan-id=100 interface=ether1
```

Create two bridges, one for each VLAN.

```
/interface bridge
add name=bridge-vlan900
add name=bridge-vlan100
```

Add the VLAN 900 to the bridge *bridge-vlan900* and the second interface (which will be an untagged management interface).

```
/interface bridge port
add bridge=bridge-vlan900 interface=ether1-vlan900
add bridge=bridge-vlan900 interface=ether2
```

Add an IP address to the VLAN for the purpose of management access.

```
/ip address add address=198.8.8.241/24 interface=bridge-vlan900
/ip route add gateway=198.8.8.1
```

Add the VLAN 100 to the bridge *bridge-vlan100* and the remaining interfaces which will be the untagged peering interfaces connecting peer ISPs.

```
/interface bridge port
add bridge=bridge-vlan100 interface=ether1-vlan100

add bridge=bridge-vlan100 interface=ether3 comment="<<ISP 1>> VLAN 100"

...

add bridge=bridge-vlan100 interface=ether24 comment="<<ISP 22>> VLAN 100"
```

15.2. Software-defined configuration

The number of available OF capable Ethernet switches is limited at this stage. This project was able to access a Netgear ProSAFE M4800-28G switch. However, this switch proved to have a limited and poorly compliant OF implementation. To effectively demonstrate the concept, an external OvS based OF switch was developed on Dell PowerEdge R610 hardware, as demonstrated in the left pane of Illustration 20.

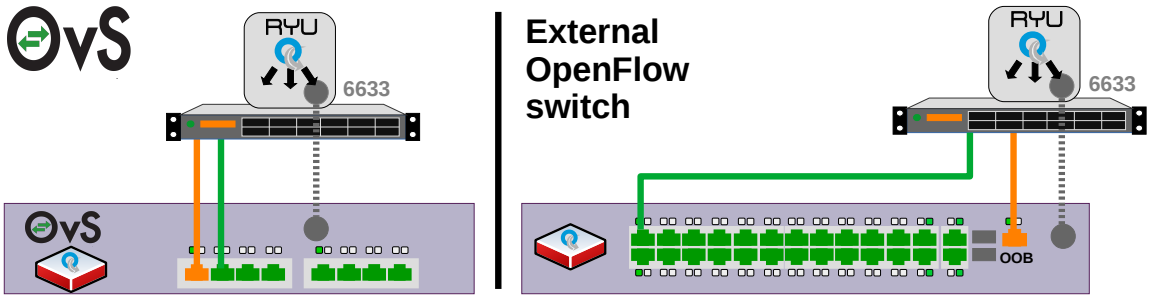


Illustration 20: Software-defined external switches

15.2.1) Open vSwitch

This configuration is for generic hardware with Ubuntu 18.04 LTS OS hosting an OvS. Install Ubuntu as outlined in chapter 2 except for the network section. For networking follow these instructions during that section of the installation process.

- Configure the network
  - Primary network interface: **eth1**
  - IP address will attempt to autoconfigure
    - Network autoconfiguration failed; <**Continue**>
    - Network configuration method: **Configure network manually**
  - IP address: <**local preference** from man LAN, i.e. 198.8.8.241/24>
  - Gateway: <**local preference** from man LAN, i.e. 198.8.8.1>
  - Name server : <**local preference** from man LAN, i.e. 198.8.8.231>
  - Hostname : <**local preference**, i.e. OF-SW>
  - Domain name : <**local preference** from schema, i.e. netlabs.tst>

Check the install.

```
@OF-SW:~$ lsb_release -cidr
Distributor ID:   Ubuntu
Description:      Ubuntu 18.04
Release:          18.04
Codename:         bionic
```

This will generate a *netplan* file similar to this.

```
OvS:~$ cat /etc/netplan/01-netcfg.yaml
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    eno1:
      dhcp4: no
      addresses: [198.8.8.241/24]
      gateway4: 198.8.8.1
      nameservers:
        search: [netlabs.tst]
        addresses: [198.8.8.231]
```

Install OvS.

```
OF-SW:~$ sudo apt-get install -y openvswitch-switch
```

OvS can be configured via the *ovs-vsctl* and *ovs-ofctl* commands.

```
OF-SW:~$ sudo ovs-vsctl show
37d0d030-78c1-4c97-a0af-248ef8166d03
    ovs_version: "2.9.2"
```

Create a bridge.

```
OF-SW:~$ sudo ovs-vsctl add-br br0
```

Get list of interfaces.

```
OF-SW:~$ networkctl list --no-legend | awk '{print $2}'
eth1
eth2
eth3
eth4
eth5
eth6
eth7
eth8
```

The first interface is reserved for the management LAN and has already been configured so commit all the remaining interfaces to the bridge ***br0***. Though the second interface is considered the data trunk, the remaining interfaces are for peers. To ease identification at the *SC* the switch data interface OF number is *1500 + interface number* and the peer interfaces are *1000 + interface number*.

```
OF-SW:~$ sudo ovs-vsctl add-port br0 eth2 -- set interface eth2 ofport=1502
OF-SW:~$ sudo ovs-vsctl add-port br0 eth3 -- set interface eth3 ofport=1003
OF-SW:~$ sudo ovs-vsctl add-port br0 eth4 -- set interface eth4 ofport=1004
OF-SW:~$ sudo ovs-vsctl add-port br0 eth5 -- set interface eth5 ofport=1005
OF-SW:~$ sudo ovs-vsctl add-port br0 eth6 -- set interface eth6 ofport=1006
OF-SW:~$ sudo ovs-vsctl add-port br0 eth7 -- set interface eth7 ofport=1007
OF-SW:~$ sudo ovs-vsctl add-port br0 eth8 -- set interface eth8 ofport=1008
```

Define the *SC* and set the failmode to *standalone*. This means that if the *SC* is not available then the OvS with return to *traditional* Ethernet switching mode.

```
OF-SW:~$ sudo ovs-vsctl set-controller br0 tcp:198.8.8.235:6633
OF-SW:~$ sudo ovs-vsctl set-fail-mode br0 standalone
```

Setup the OvS for persistent operation via system start and stop scripts and an OvS configuration service.

```
OF-SW:~$ sudo mkdir /srv/ovs/

OF-SW:~$ cat <<EOM | sudo tee /srv/ovs/ovs-config_start.sh
#!/bin/bash

### OvS configuration script for OF-SW ###

ovs-vsctl --if-exists del-br br0
ovs-vsctl add-br br0
ovs-vsctl add-port br0 eth2 -- set interface eth2 ofport=1502
ovs-vsctl add-port br0 eth3 -- set interface eth3 ofport=1003
ovs-vsctl add-port br0 eth4 -- set interface eth4 ofport=1004
ovs-vsctl add-port br0 eth5 -- set interface eth5 ofport=1005
ovs-vsctl add-port br0 eth6 -- set interface eth6 ofport=1006
ovs-vsctl add-port br0 eth7 -- set interface eth7 ofport=1007
ovs-vsctl add-port br0 eth8 -- set interface eth8 ofport=1008
ovs-vsctl set-controller br0 tcp:198.8.8.235:6633
ovs-vsctl set-fail-mode br0 standalone
ip link set br0 up
ip link set eth2 up
ip link set eth3 up
ip link set eth4 up
ip link set eth5 up
ip link set eth6 up
ip link set eth7 up
ip link set eth8 up
EOM

OF-SW:~$ cat <<EOM | sudo tee /srv/ovs/ovs-config_stop.sh
#!/bin/bash

### OvS configuration script for OF-SW ###

ovs-vsctl --if-exists del-br br0
ip link set eth2 down
ip link set eth3 down
ip link set eth4 down
ip link set eth5 down
ip link set eth6 down
ip link set eth7 down
ip link set eth8 down
EOM
```

Change the ownership of the OvS start and stop scripts and make them executable.

```
OF-SW:~$ sudo chown root: /srv/ovs/ovs-config_start.sh
OF-SW:~$ sudo chmod +x /srv/ovs/ovs-config_start.sh
OF-SW:~$ sudo chown root: /srv/ovs/ovs-config_stop.sh
OF-SW:~$ sudo chmod +x /srv/ovs/ovs-config_stop.sh
```

Setup and enable the OvS configuration service.

```
OF-SW:~$ cat <<EOM | sudo tee /etc/systemd/system/ovs-config.service
[Unit]
Description=OvS Switch Configuration
After=systemd-user-sessions.service

[Service]
Type=forking
ExecStart=/srv/ovs/ovs-config_start.sh
ExecStop=/srv/ovs/ovs-config_stop.sh
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
EOM

OF-SW:~$ sudo systemctl daemon-reload
OF-SW:~$ sudo chown root: /etc/systemd/system/ovs-config.service
OF-SW:~$ sudo chmod +x /etc/systemd/system/ovs-config.service
OF-SW:~$ sudo systemctl start ovs-config.service
OF-SW:~$ sudo systemctl stop ovs-config.service
OF-SW:~$ sudo systemctl enable ovs-config.service

OF-SW:~$ systemctl status ovs-config.service
● ovs-config.service - OvS Switch Configuration
   Loaded: loaded (/etc/systemd/system/ovs-config.service; enabled; vendor
preset: enabled)
   Active: active (exited) since Sun 2019-03-10 13:32:25 EAT; 16min ago
  Process: 1374 ExecStart=/srv/ovs/ovs-config_start.sh (code=exited,
status=0/SUCCESS)

Mar 10 13:32:24 OF-SW ovs-vsctl[1503]: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-
port br0 eth2 -- set interface eth2 ofport=1102
Mar 10 13:32:24 OF-SW ovs-vsctl[1506]: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-
port br0 eth3 -- set interface eth3 ofport=1103
Mar 10 13:32:24 OF-SW ovs-vsctl[1508]: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-
port br0 eth4 -- set interface eth4 ofport=1104
Mar 10 13:32:24 OF-SW ovs-vsctl[1515]: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-
port br0 eth5 -- set interface eth5 ofport=1105
Mar 10 13:32:24 OF-SW ovs-vsctl[1516]: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-
port br0 eth6 -- set interface eth6 ofport=1106
Mar 10 13:32:24 OF-SW ovs-vsctl[1517]: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-
port br0 eth7 -- set interface eth7 ofport=1107
Mar 10 13:32:24 OF-SW ovs-vsctl[1518]: ovs|00001|vsctl|INFO|Called as ovs-vsctl add-
port br0 eth8 -- set interface eth8 ofport=1108
Mar 10 13:32:24 OF-SW ovs-vsctl[1519]: ovs|00001|vsctl|INFO|Called as ovs-vsctl set-
controller br0 tcp:198.8.8.235:6633
Mar 10 13:32:24 OF-SW ovs-vsctl[1544]: ovs|00001|vsctl|INFO|Called as ovs-vsctl set-
fail-mode br0 standalone
Mar 10 13:32:25 OF-SW systemd[1]: Started OvS Switch Configuration.
```

Review the configuration as follows:

```
ubuntu@OF-SW:~$ sudo ovs-vsctl show
37d0d030-78c1-4c97-a0af-248ef8166d03
    Bridge "br0"
        Controller "tcp:198.8.8.235:6633"
        fail_mode: standalone
        Port "br0"
            Interface "br0"
                type: internal
        Port "eth2"
            Interface "eth2"
        Port "eth3"
            Interface "eth3"
        Port "eth4"
            Interface "eth4"
        Port "eth5"
            Interface "eth5"
        Port "eno6"
            Interface "eth6"
        Port "eth7"
            Interface "eth7"
        Port "eth8"
            Interface "eth8"
    ovs_version: "2.9.2"
```

Flows sent to the OvS bridge can be accessed as follows (this demonstrates the table-miss flow from the *SC*).

```
OF-SW:~$ sudo ovs-ofctl dump-flows br0
 cookie=0x0, duration=193.243s, table=0, n_packets=8, n_bytes=600,
priority=0 actions=CONTROLLER:65535
```

15.2.2) Netgear ProSAFE Switch

Default User: *admin* Password: <*no password*>

Configure the console to 9,600 baud (to match the other devices).

```
(M4300-28G)> enable
(M4300-28G)> configure
(M4300-28G)(Config)# line console
(M4300-28G)(Config-line)# serial baudrate 9600
```

Configure the switch hostname.

```
(M4300-28G)# hostname OF-SW
```

Configure the OOB management interface.

```
(OF-SW) # serviceport protocol none
(OF-SW) # serviceport ip 198.8.8.241 255.255.255.0 198.8.8.1
(OF-SW) # serviceport ipv6 address 2a98:8:8::241/48
(OF-SW) # serviceport ipv6 gateway 2a98:8:8::1
```

Define the IP address and TCP port number of the *SC* on the management LAN.

```
(OF-SW) (Config) # configure
(OF-SW) (Config) # openflow ip-mode serviceport
(OF-SW) (Config) # openflow controller 1985 6633 tcp
(OF-SW) (Config) # openflow variant openflow13
(OF-SW) (Config) # openflow enable
(OF-SW) (Config) # exit
```

Review the OF configuration (note the operational status of the connection).

```
(OF-SW) # show openflow

Administrative Mode.......................... Enable
Operational Status........................... Enabled
Disable Reason............................... None
IP Address................................... 198.8.8.241
IP Mode...................................... ServicePort IP
Static IP Address............................ 0.0.0.0
OpenFlow Variant............................. OpenFlow 1.3
Passive Mode................................. Disable
```

Review installed flows.

```
(OF-SW) # show openflow installed flows

Flow type "1DOT3"

Match criteria:
Flow table 60 : Priority 0
Actions:
Redirect:  CONTROLLER

Hard Timeout 0 : Idle Timeout 0

Status: Duration  1485 : Packet Count  843 : Byte Count 66941
Idle 520 : installed in hardware  1

Flow type "1DOT3"

Match criteria:
Flow table 60 : Priority  100
Ingress port 1/0/22
Actions:
Egress port 1/0/24

Hard Timeout 0 : Idle Timeout   0

Status:
Duration 537 : Packet Count  318 : Byte Count  24799
Idle 0 : installed in hardware  1
```

Show the *SC* configuration.

```
(OF-SW) # show openflow configured controller


     IP Address          IP Port      Connection Mode             Role
---------------  ---------------  --------------------     ------------
    1985             6633                  tcp            Equal
```

Save the configuration.

```
(OF-SW)# save
This operation may take a few minutes.
Management interfaces will not be available during this time.
Are you sure you want to save? (y/n) y
Config file 'startup-config' created successfully.
Configuration Saved!
```

16. IXP Peering members

IXP peering members, the ISPs, ICPs, ASPs, RENs, banks, Government entities and other members with ASN must configure routing on their respective IXP routers. The full configuration of these routers is beyond the scope of this document and can vary significantly from member to member; however, the BGP configuration on each is similar. Here are examples for common router vendors.

16.1. Cisco Router

Default Username: <*no username*> Password: <*no password*>

Add BGP instance and networks to be advertised.

```
ISP1_RTR(config)# router bgp 5111
ISP1_RTR(config-router)# bgp router-id 200.1.1.1
ISP1_RTR(config-router)# no bgp enforce-first-as
ISP1_RTR(config-router)# neighbor 199.9.9.1 remote-as 5999
ISP1_RTR(config-router)# neighbor 199.9.9.233 remote-as 5999
ISP1_RTR(config-router)# neighbor 2A99:9:9::234 remote-as 112
ISP1_RTR(config-router)# neighbor 2a99:9:9::1 remote-as 5999
ISP1_RTR(config-router)# neighbor 2a99:9:9::233 remote-as 5999
ISP1_RTR(config-router)# neighbor 2a99:9:9::234 remote-as 112
```

Configure IPv4 and IPv6 address families.

```
ISP1_RTR(config-router)# address-family ipv4
ISP1_RTR(config-router-af)# network 199.1.1.0 mask 255.255.255.0
ISP1_RTR(config-router-af)# neighbor 199.9.9.1 activate
ISP1_RTR(config-router-af)# neighbor 199.9.9.233 activate
ISP1_RTR(config-router-af)# neighbor 199.9.9.234 activate
ISP1_RTR(config-router-af)# exit-address-family

ISP1_RTR(config-router)# address-family ipv6
ISP1_RTR(config-router-af)# network 2a99:1:1::/48
ISP1_RTR(config-router-af)# neighbor 2a99:9:9::1 activate
ISP1_RTR(config-router-af)# neighbor 2a99:9:9::233 activate
ISP1_RTR(config-router-af)# neighbor 2a99:9:9::234 activate
ISP1_RTR(config-router-af)# exit-address-family
```

16.2. Juniper MX Series Router

Default Login: *root* Password: <*no password*>

Set the peer type to external BGP (eBGP).

```
user@ISP_2# set protocols bgp group external-peers type external
```

Set the local AS number.

```
user@ISP_2# set routing-options autonomous-system 5222
```

Specify the AS number of the external AS 5999.

```
user@ISP_2# set protocols bgp group external-peers peer-as 5999
```

Create the BGP group, and add the external neighbour addresses for the *CS* and the *RS*.

```
user@ISP_2# set protocols bgp group external-peers neighbor 199.9.9.1
user@ISP_2# set protocols bgp group external-peers neighbor 199.9.9.233
user@ISP_2# set protocols bgp group external-peers neighbor 2a99:9:9::1
user@ISP_2# set protocols bgp group external-peers neighbor 2a99:9:9::233
```

Specify the AS number of the external AS112.

```
user@ISP_2# set protocols bgp group external-peers peer-as 112
```

Create the BGP group, and add the external neighbour addresses for the *BS*.

```
user@ISP_2# set protocols bgp group as112-peers neighbor 199.9.9.234
user@ISP_2# set protocols bgp group as112-peers neighbor 2a99:9:9::234
```

16.3. MikroTik Router

Default Username: *admin* Password: <*no password*>

Add BGP instance and networks to be advertised.

```
/routing bgp
instance add name=ASN5111 as=5333 router-id=200.3.3.3
network add network=199.3.3.0/24
network add network=2a99:3:3::/48
```

Add BGP Peers for transit.

```
/routing bgp peer
add name=ixp_rs instance=AS5333 remote-as=5999 remote-address=199.9.9.1
add name=ixp_cs instance=AS5333 remote-as=5999 remote-address=199.9.9.233
add name=ixp_bs instance=AS5333 remote-as=112 remote-address=199.9.9.234

add name=ixp_rs instance=AS5333 remote-as=5999 address-families=ipv6
remote-address=2a99:9:9::1
add name=ixp_cs instance=AS5333 remote-as=5999 address-families=ipv6
remote-address=2a99:9:9::233
add name=ixp_bs instance=AS5333 remote-as=112 address-families=ipv6
remote-address=2a99:9:9::234
```

17. Maintenance

The IXPBuilder system has some key files upon which it is built that the IXP engineer should be aware of. These are outlined in Table 3.

Table 3: IXPBuilder files

File	Function
/ixp/tools/ixp-install.sh	Installs the IXPBuilder system on Ubuntu 10.04 LTS.
/ixp/tools/ixp-uninstall.sh	Uninstalls the IXPBuilder system from Ubuntu 10.04 LTS.
/opt/ixp/ixp	CLI command file.
/usr/local/lib/python3.6/dist-packages/ixp.py	IXP class and function library module.
/ixp/tools/ixp-backup.sh	Backs up the installation to ~/ixp/BACKUPS/.
/ixp/tools/ixp-fcm-grabber.py	Extracts function, class and method data from ixp.py.
/var/ixp/ixp_switch_13_template.py	Becomes ixp_switch_13.py on the SC container.

17.1. The IXP command program

The IXP command program imports the ixp.py class and function library module, instantiates the various classes as required by the command at the time and then passes the CLI data to the instance.

17.2. The IXP backup

The ixp-backup.sh tool generates a backup of the system when called. Backups are timestamped and stored in the ~/ixp/BACKUPS/ directory.

```
cIXP:~/ixp/tools$ ./ixp-backup.sh
20190424-1708

Backup script for IXP builder package
-------------------------------------
Backing up: ./ixp-install.sh
Backing up: /opt/ixp/ixp
Backing up: /usr/local/lib/python3.6/dist-packages/ixp.py
Backing up files in /var/log/ixp in
/home/ubuntu/ixp/BACKUPS/20190424-1708/var_log_ixp
Backing up files in /var/ixp in /home/ubuntu/ixp/BACKUPS/20190424-1708/var_ixp
Backing up files in /var/ixp/code in
/home/ubuntu/ixp/BACKUPS/20190424-1708/var_ixp_code
IXP Builder package files backed up in /home/ubuntu/ixp/BACKUPS/20190424-1708
```

17.3. The IXP module

The IXP functions, classes and method information can be extracted at any time. A timestamped file is generated in the ~/ixp/ directory.

```
cIXP:~/ixp/tools$ ./ixp-fcm-grabber.py

Lists of functions, classes and associated methods
have been saved to ../20190502-171233_def_list.txt
```

Here is a list at the time of writing. The number in parenthesis is the line number in the `ixp.py` module.

17.3.1) Independent functions

```
1   (129) _function_call_location()
    Function to check if a function is called via module or locally.

2   (142) _sudo_call()
    Function to allow sudo calls.

3   (185) _ixp_timestamp()
    Function presents current time, date and timestamp when requested.

4   (202) _site_number()
    Function to extract the site number and returns.

5   (219) _switching_type()
    Function to extract the switching type and returns.

6   (236) _site_type()
    Function to extract the site type and returns.

7   (253) _get_file_dir_info()
    Function to get information from file or directory.

8   (291) _is_empty()
    Function to test is a variable, list or dictionary is empty.

9   (307) _ixp_debug()
    Function to debug function, writes logs to the daily logfile.

10  (345) _ixp_schema_servers()
    Function to extract a list of servers from the IXP Schema.

11  (375) _validate_ip_net()
    Function to validate IP Networks and IP addresses to ensure DB is clean.

12  (430) _ixp_schema_network()
    Function to extract the network from the IXP Schema.

13  (464) _non_ip_test()
    Function to test non-IP related values.

14  (658) _ixp_database_drop()
    Function to drop an ixp database.

15  (687) _ixp_database_create()
    Function to create an ixp database, create DB table if it doesn't exist.

16  (725) _ixp_database_rebuild()
    Function to rebuild an ixp database, rebuilds DB tables.

17  (769) _ixp_database_insert()
    Function to insert into an ixp database.
```

```
18  (800) _ixp_database_delete()
    Function to delete an ixp database.

19  (861) _ixp_database_update()
    Function to update ixp database.

20  (946) _read_db()
    Function to read the DB tables.

21  (1040) _parse_range_list()
    Function to parse a comma-separated list of numbers and ranges.

22  (1073) _rangeStr()
    Function to convert two integers into a range start-end, or a single value if
they are the same.

23  (1089) _makeRange()
    Function to convert sequence of int to string with the ranges.

24  (1113) _stringifyRanges()
    Function to join sequence into string.

25  (1126) _string_search_replace()
    Function to open a file, search and replace and output to another file.

26  (1163) _ip_link_list()
    Function to list the available interfaces on the host computer.

27  (1203) _gai_build()
    Function to Get Address Information (GAI), default address selection for IPv6
(rfc3484).

28  (1231) _validate_options()
    Function to validate commandline options.

29  (1300) _remove_minus_from_options()
    Function to replace - and remove --.

30  (1321) _list_to_dict()
    Function to convert list to dictionary.
```

17.3.2) Classes and methods

```
1   : (1349) _ManagedFile
     _ManagedFile class.

    1   : (1352) __init__()
          _ManagedFile class constructor method.

    2   : (1361) __str__()
          _ManagedFile class __str__ method.

    3   : (1368) __repr__()
          _ManagedFile class __repr__ method.

    4   : (1375) __enter__()
          _ManagedFile class __enter__ method.

    5   : (1383) __exit__()
          _ManagedFile class __exit__ method.


2   : (1397) IxpHelp
     IXP Help class.

    1   : (1400) __init__()
          IxpHelp class constructor method.

    2   : (1407) __str__()
```

```
      IxpHelp class __str__ method.

3   : (1414) __repr__()
      IxpHelp class __repr__ method.

4   : (1425) ixp_help()
      Method to provide ixp help.

5   : (1453) _ixp_schema_help()
      Method to provide ixp schema help.

6   : (1475) _ixp_schema_show_help()
      Method to provide ixp schema show help.

7   : (1511) _ixp_schema_default_help()
      Method to provide ixp schema default help.

8   : (1534) _ixp_schema_build_help()
      Method to provide ixp schema build help.

9   : (1567) _ixp_host_help()
      Method to provide ixp host help.

10  : (1587) _ixp_host_build_help()
      Method to provide ixp host build help.

11  : (1611) _ixp_switch_help()
      Method to provide ixp switch help.

12  : (1632) _ixp_switch_delete_help()
      Method to provide ixp switch delete help.

13  : (1653) _ixp_switch_set_help()
      Method to provide ixp switch set help.

14  : (1685) _ixp_switch_set_speed_help()
      Method to provide ixp switch set speed help.

15  : (1708) _ixp_server_help()
      Method to provide ixp server help.

16  : (1731) _ixp_server_build_help()
      Method to provide ixp server build help.

17  : (1765) _ixp_server_show_help()
      Method to provide ixp server show help.

18  : (1799) _ixp_server_start_help()
      Method to provide ixp server start help.

19  : (1828) _ixp_server_stop_help()
      Method to provide ixp server stop help.

20  : (1857) _ixp_server_delete_help()
      Method to provide ixp server delete help.

21  : (1892) _ixp_software_help()
      Method to provide ixp software help.

22  : (1912) _ixp_software_install_help()
      Method to provide ixp software install help.

23  : (1943) _ixp_software_configure_help()
      Method to provide ixp software configure help.

24  : (1974) _ixp_sdn_help()
      Method to provide ixp sdn help.

25  : (1993) _ixp_sdn_list_help()
      Method to provide ixp sdn switch list help.

26  : (2013) _ixp_sdn_list_switch_help()
```

```
          Method to provide ixp sdn switch list help.

    27  : (2034) _ixp_sdn_list_flows_help()
          Method to provide ixp sdn switch flows list help.

    28  : (2055) _ixp_peer_help()
          Method to provide ixp peer help.

    29  : (2078) _ixp_peer_add_help()
          Method to provide ixp peer add help.

    30  : (2104) _ixp_peer_delete_help()
          Method to provide ixp peer delete help.

    31  : (2125) _ixp_peer_list_help()
          Method to provide ixp peer list help.

    32  : (2145) _ixp_peer_status_help()
          Method to provide ixp peer status help.

    33  : (2165) _ixp_peer_route_help()
          Method to provide ixp peer route help.

    34  : (2185) _ixp_remote_help()
          Method to provide ixp remote help.

    35  : (2210) _ixp_remote_keys_help()
          Method to provide ixp remote help.

    36  : (2236) _ixp_remote_site_help()
          Method to provide ixp remote site help.

    37  : (2258) _ixp_remote_site_add_help()
          Method to provide ixp remote site add help.

    38  : (2284) _ixp_remote_site_delete_help()
          Method to provide ixp remote site delete help.

    39  : (2303) _ixp_remote_command_help()
          Method to provide ixp remote command help.


3   : (2332) _IxpLxc
      IXP LXC class.

    1   : (2335) __init__()
          IxpLxc class constructor method.

    2   : (2342) __str__()
          _IxpLxc class __str__ method.

    3   : (2349) __repr__()
          _IxpLxc class __repr__ method.

    4   : (2360) _lxc_exec()
          Method to access lxc exec.

    5   : (2409) _lxc_start()
          Method to start an LXC container.

    6   : (2427) _lxc_restart()
          Method to restart an LXC container.

    7   : (2445) _lxc_stop()
          Method to stop an LXC container.

    8   : (2463) _lxc_status()
          Method to get the status of an LXC container.

    9   : (2485) _lxc_status_up()
          Method to get the status of an LXC container, if 'Stopped' bring up.
```

```
    10  : (2519) _lxc_delete()
          Method to delete an LXC container.

    11  : (2573) _lxc_upgrade()
          Method to upgrade the software on the servers.

    12  : (2621) _lxc_file_test()
          Method to check if lxc file exists.

    13  : (2642) _lxc_rm()
          Method to remove files from container.

    14  : (2663) _lxc_clone()
          Method to copy the container template to the required container.

    15  : (2681) _lxc_pull()
          Method to pull files from container.

    16  : (2704) _lxc_push()
          Method to pull files from container.

    17  : (2725) _lxc_container_boot_delay()
          Method to allow the container enough time to boot.

    18  : (2742) _lxc_hypervisor_containers()
          Method to get list of containers on hypervisor.

    19  : (2760) _lxc_internet_test()
          Method to check Internet connectivity.

    20  : (2794) _bird_restart()
          Method to restart the bird server.

    21  : (2821) _bind9_restart()
          Method to restart the bind9 server.


4   : (2843) IxpSchema
     IXP Schema Class.

    1   : (2846) __init__()
          IxpSchema class constructor method.

    2   : (2901) __str__()
          IxpSchema class __str__ method.

    3   : (2908) __repr__()
          IxpSchema class __repr__ method.

    4   : (2920) _ixp_schema_help_handler()
          Schema help handler.

    5   : (2940) _ixp_schema_build()
          Method to build the IXP Schema and IXP Site info.

    6   : (3208) _ixp_schema_show()
          Method to display contents of the IXP Schema.

    7   : (3259) _ixp_schema_test_options()
          Method to Test 'ixp_schema' cli options.

    8   : (3368) ixp_schema()
          Method to handle ixp schema' commands.


5   : (3444) IxpHost
     IXP Host class.

    1   : (3447) __init__()
          IxpHost class constructor method.

    2   : (3479) __str__()
```

```
          IxpHost class __str__ method.

    3   : (3486) __repr__()
          IxpHost class __repr__ method.

    4   : (3498) _ixp_host_help_handler()
          host help handler.

    5   : (3516) _ixp_host_build_core()
          Method to build core servers.

    6   : (3594) _ovs_lxd_config_file_manager()
          Method to manage OvS configuration file.

    7   : (3740) _ixp_host_build_soft()
          Method to build host software-defined model.

    8   : (3959) _ixp_host_build_trad()
          Method to build traditional model.

    9   : (4217) _ixp_host_show()
          Method to show 'ixp_host' interfaces.

    10  : (4253) _ixp_host_test_options()
          Method to test 'ixp_host' cli options.

    11  : (4449) ixp_host()
          Method to handle the 'ixp host' command.


6   : (4535) IxpSwitch
     IXP Switch Class.

    1   : (4538) __init__()
          IxpSwitch class constructor method.

    2   : (4554) __str__()
          IxpSwitch class __str__ method.

    3   : (4561) __repr__()
          IxpSwitch class __repr__ method.

    4   : (4573) _ixp_switch_help_handler()
          switch help handler.

    5   : (4595) _ixp_switch_test_options()
          Method to handle the 'ixp switch' command.

    6   : (4671) ixp_switch()
          Method to handle the 'ixp switch' command.


7   : (4959) IxpServer
     IXP Server Class.

    1   : (4962) __init__()
          IxpServer class constructor method.

    2   : (4974) __str__()
          IxpServer class __str__ method.

    3   : (4981) __repr__()
          IxpServer class __str__ method.

    4   : (4993) _ixp_server_help_handler()
          server help handler.

    5   : (5015) _non_schema_instances_test()
          Method to check if there are non schema instances.

    6   : (5104) _nic_profile()
          Method to test for and/or assign dualnic profile to container method.
```

```
    7   : (5141) _container_netplan_yaml()
          Method to test for, create and apply netplan for containers.

    8   : (5300) _lxc_container_image()
          Method to download a container template.

    9   : (5382) _ixp_server_build()
          Method to build the server.

    10  : (5477) _ixp_server_test_options()
          Method to test 'ixp_server' cli options.

    11  : (5601) ixp_server()
          Method to handle 'ixp server' dict_['cmd']s.


8   : (5741) IxpSoftware
     IXP Software Class.

    1   : (5744) __init__()
          IxpSoftware class constructor method.

    2   : (5813) __str__()
          IxpSoftware class __str__ method.

    3   : (5820) __repr__()
          IxpSoftware class __repr__ method.

    4   : (5832) _ixp_software_help_handler()
          software help handler.

    5   : (5848) _replace_nameserver()
          Method to replace the upstream server with NS addresses.

    6   : (5902) _bind_structure_test()
          Method to check if BIND9 directory structure exists and backup if so.

    7   : (5959) _nameserver_configure()
          Method to configure nameserver.

    8   : (6299) _bird_configure()
          Method to build the BIRD configuration on containers.

    9   : (6398) _birdseye_configure()
          Method to configure Birdseye.

    10  : (6461) _bs_software_configure_build()
          Method to configure the blackhole AS112 server.

    11  : (6689) _sc_software_configure_build()
          Method to configure the SDN Controller server.

    12  : (6759) _ixp_software_configure()
          Method to configure software on the servers.

    13  : (6822) _ixp_software_install()
          Method to install software on the servers.

    14  : (7037) _ixp_software_test_options()
          Method to test 'ixp_software' cli options.

    15  : (7158) ixp_software()
          Method to handle 'ixp software' commands.


9   : (7185) IxpSdn
     IXP SDN Class.

    1   : (7188) __init__()
          IxpSdn class constructor method.
```

```
    2   : (7202) __str__()
          IxpSdn class __str__ method.

    3   : (7209) __repr__()
          IxpSdn class __repr__ method.

    4   : (7221) _ixp_sdn_help_handler()
          sdn help handler.

    5   : (7241) _ixp_sdn_list_switch()
          Method to list the SDN switches from the SDN Controller.

    6   : (7285) _ixp_sdn_list_flows()
          Method to list the SDN flows from the SDN Switches.

    7   : (7331) _scrape_rest_api()
          Method to scrape RESTful API.

    8   : (7359) _get_switch_maps()
          Method to get list of OpenFlow switches from Controller.

    9   : (7392) _dpid_port_map()
          Method to get switch and port numbers from the database and match dpid.

    10  : (7437) _add_delete_flow()
          Method to add flows to OpenFlow switches via RESTful API.

    11  : (7486) _clear_flows()
          Method to clear all flows from an OpenFlow switch via RESTful API.

    12  : (7526) _ixp_sdn_test_options()
          Method to test 'ixp_sdn' cli options.

    13  : (7631) ixp_sdn()
          Method to handle 'ixp sdn' commands.


10  : (7660) IxpPeer
     IXP Peer Class.

    1   : (7663) __init__()
          IxpPeer class constructor method.

    2   : (7678) __str__()
          IxpPeer class __str__ method.

    3   : (7685) __repr__()
          IxpPeer class __repr__ method.

    4   : (7697) _ixp_peer_help_handler()
          Peer help handler.

    5   : (7721) _ixp_peer_sync()
          Method to add a peer to a route server.

    6   : (7840) _ixp_peer_delete()
          Method to delete a peer from the database.

    7   : (7913) _ixp_peer_add()
          Method to add a peer to a route server.

    8   : (8009) _ixp_peer_list()
          Method to get peer data from the database.

    9   : (8126) _ixp_peer_status()
          Method to get BGP status from BIRD servers.

    10  : (8250) _ixp_peer_route()
          Method to get route tables from BIRD servers.

    11  : (8428) _ixp_peer_test_options()
          Method to test 'ixp_peer' cli options.
```

```
    12  : (8545) ixp_peer()
          Method, handles 'ixp peer' command.


11  : (8592) IxpRemote
     IXP Remote Class.

    1   : (8595) __init__()
          IxpRemote class constructor method.

    2   : (8637) __str__()

    3   : (8642) __repr__()

    4   : (8651) _ixp_remote_help_handler()
          remote help handler.

    5   : (8675) _ixp_remote_command()
          Method to execute remote commands.

    6   : (8750) _ixp_remote_site_add()
          Method to add remote mIXP database entries.

    7   : (8777) _ixp_remote_site_delete()
          Method to delete remote mIXP database entries.

    8   : (8800) _ixp_remote_site_show()
          Method to show remote mIXP database entries.

    9   : (8872) _ixp_remote_keygen()
          Method to generate SSH key pair.

    10  : (8923) _ixp_remote_key_enter()
          Method to add SSH public key to mIXP.

    11  : (8963) _ixp_remote_key_show()
          Method to show the SSH public key.

    12  : (8975) _ixp_remote_test_options()
          Method to test 'ixp_remote' cli options.

    13  : (9258) ixp_remote()
          Method to handle 'ixp remote' commands.
```

17.4. The SDN switch v1.3 subclass

The `/var/ixp/ixp_switch_13_template.py` file becomes the `ixp_switch_13.py` file on the *SC* LXC after install. This is called by Ryu at start time and becomes the `IxpSwitch13` subclass of the Ryu `ryu.base.app_manager` class.

```
1   : (42) IxpSwitch13
     IXP switch v1.3 class.

    1   : (45) __init__()
          IxpSwitch13 class constructor method.

    2   : (64) _switch_features_handler()
          Method for initial communications with switch.

    3   : (109) _packet_in_handler()
          Method to handle Packet in messages.

    3   : (185) _add_flow()
          Method to add flow_mod to the switch.

    3   : (208) _clear_flows()
          Method to clear flows from the switch.
```

18. References

Abley, J. and Sotomayor, W. (2015) 'RFC7534. AS112 Nameserver Operations'. Internet Engineering Task Force. Available at: https://tools.ietf.org/pdf/rfc7534.pdf.

BIND9 - Domain Name System (DNS) (no date). Available at: https://www.isc.org/downloads/bind/ (Accessed: 30 November 2017).

Birdseye micro-service (no date) *INEX A Simple Secure Micro Service for Querying Bird (JSON API)*. Available at: https://github.com/inex/birdseye (Accessed: 20 May 2016).

Bonica, R. et al. (2017) 'RFC8190. Updates to the Special-Purpose IP Address Registries'. Internet Engineering Task Force. Available at: https://tools.ietf.org/pdf/rfc8190.pdf.

Cotton, M. et al. (2013) 'RFC6890. Special-Purpose IP Address Registries'. Internet Engineering Task Force. Available at: https://tools.ietf.org/pdf/rfc6890.pdf.

Ondrej Filip et al. (no date) 'BIRD Programmer's Documentation'. Available at: http://bird.network.cz/?get_doc&f=prog.html (Accessed: 9 October 2017).

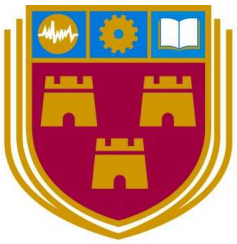
Oren Ben-Kiki and Clark Evans (2009) YAML Ain't Markup Language (YAMLTM) Version 1.2, yaml.org. Available at: http://www.yaml.org/spec/1.2/spec.html (Accessed: 23 January 2017).

Rekhter, Y. et al. (1996) 'RFC1918, Address Allocation for Private Internets'. Available at: https://tools.ietf.org/pdf/rfc1918.pdf (Accessed: 8 January 2017).

RYU project team (no date) RYU SDN Framework — Ryu documentation Release 4.30, Ryu documentation Release 4.30. Available at: https://media.readthedocs.org/pdf/ryu/latest/ryu.pdf (Accessed: 30 November 2018).

The BIRD Internet Routing Daemon Project (no date). Available at: http://bird.network.cz/ (Accessed: 17 January 2017).

'Ubuntu 18.04 (Bionic Beaver)' (2018). Canonical Limited. Available at: https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes.



INSTITUTE of TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

Enabling models of Internet eXchange Points to support spatial planning: the case for East Africa

Appendix B1

Political economy study

- Interview Guide
- Participant Information Leaflet

Institute of Technology, Carlow

gamecore
engaging people with technology
Research Centre
Department of Computing & Networking,
INSTITUTE OF TECHNOLOGY, CARLOW

netLabs!UG
Research Centre
College of Engineering, Design, Art and Technology
MAKERERE UNIVERSITY, KAMPALA

Enabling affordable models of Internet eXchange Points

Diarmuid Seosamh Ó Briain

B.Sc. (hons), M.Sc.

Political-economy study

Political-economy study on the Internet in East Africa

Interviewer | Diarmuid Ó Briain

Interviewee |

Date | **Time** |

Location |

Subject | **The Internet in East Africa and the future Cloud Integrated Network (CIN)**

Time (Min)	Question / Objective	Response notes
1 – 2	**Objective - Open interview - Introduction** • Thank you for giving this time, I'm glad you've agreed to be interviewed. I want to explain how this will work. We'll do about a 45 minute interview that will be recorded, transcribed, and then edited into a "profile" that will include only your words, with my questions edited out. These words along with those of the other interviewees will be analysed so I can form an impression of the thinking of the tech leaders in East Africa on the development of the Internet here in the region. • In the interview, I will direct questions that will develop why you got involved in an Internet focused role in the first place, your understanding if the Internet today, where it is going in the future with a particular focus on East Africa. • I want to keep this as open as possible so please feel free to develop your answers.	
3	**Question 1 – Ease in – Personal to Interviewee** What sparked your passion for Technology and communications ? **Follow-up**	
5	**Question 2 – Internet today** What are your thoughts on how the Internet is delivered across East Africa today ? **Follow-up**	
5	**Question 3 – traffic nature** What are your thoughts on the direction traffic on the Internet is taking, does this present problems in the future ? **Follow-up**	

Interviewee |

Date |

Time (Min)	Question / Objective	Response notes
5	**Question 4 – Internet eXchange Points** Where do you see the IXPs within the Internet ecosystem today and have they a place in the future ? **Follow-up**	
5	**Question 5 – Software-defined** A trend towards "Software-defined", networks, functions is now emerging, how do you think this will shape the Internet of the future ? **Follow-up**	
5	**Question 6 – Internet evolution** How do you see the evolution of the Internet in the future across the East Africa region ? **Follow-up**	
5	**Question 7 – Internet ecosystem change** What changes do you think are necessary in the overall ecosystem to facilitate this evolution ? **Follow-up**	
1 – 2	**Objective - Conclude interview** • "Thank you again for the time you have given. I will send you a copy of the transcript of the interview for your giving this time". • Summarise the interview for and ask if the interviewer has any questions..	

Challenges to the delivery of future Internet applications to the regions of developing nations.

I would like to invite you to take part in a research study. Before you decide you need to understand why the research is being carried out and what it would involve for you. Please take time to read the following information carefully. Ask questions if anything you read is not clear or if you would like more information. Take time to decide whether or not to take part.

WHO I AM AND WHAT THIS STUDY IS ABOUT

My name is Diarmuid Ó Briain and I am a Ph.D. candidate at the Institute of Technology, Carlow in Ireland but based at the College of Engineering, Design, Art & Technology, Makerere University in Uganda. I am working on a research project to investigate the potential impact of future networking paradigms like Software Defined Networking (SDN) and Network Functions Virtualisation (NFV) to the delivery of networked applications in developing countries. The purpose of this research study is to explore your experience of the delivery of the Internet today in East Africa and your thoughts on the Cloud Integrated Network (CIN) of tomorrow. The project aims to understand where the networking technology is in Uganda now and what are the options for future development given the national requirements, infrastructure and position in East Africa. Your responses will form part a research project designed to address this.

WHAT WILL TAKING PART INVOLVE?

The research study will involve an interview of approximately 45 minutes to gather your thoughts on the development of the Internet to the Cloud Integrated Network (CIN) of tomorrow. The interview will involve questions relating to the role of Internet eXchange Points (IXP), Local Service Providers (LSP) and Global Service Providers (GSP) today and in the CIN of tomorrow. Also questioned will be the changing nature of traffic on the Internet over the last few years and the potential for ever further changed traffic loads in the future.

WHY HAVE YOU BEEN INVITED TO TAKE PART?

You have been invited to take part due to your experience and position of leadership in the field of the study. The study will involve participants across political, Service Provider (SP) and IXPs across East Africa. Each participant will be interviewed separately.

The information gained from this research will be used to inform my work at netLabs!UG Research Centre.

DO YOU HAVE TO TAKE PART?

Participation is completely voluntary and of course you have the right to refuse participation, refuse any question and withdraw at any time without any consequence whatsoever.

WHAT ARE THE POSSIBLE RISKS AND BENEFITS OF TAKING PART?

The study will present the views of a selection of industry notables on the transition from the Internet of today and the CIN of tomorrow. It will serve to inform my work. I cannot foresee any risk to taking part.

WILL TAKING PART BE CONFIDENTIAL?

The interview will be recorded digitally and then transcribed to text. Your response will be treated with full confidentiality. The interviews will be analysed by using a computer package by myself to produce anonymised results. Hence your anonymous responses cannot be identified or removed at a later date. At the end of the research I will write a report and the results may be published in peer reviewed journals and conference presentations. No research participant will be identifiable from any publications. This study has been reviewed and approved by the Research Ethics Committee at the Institute of Technology Carlow. Non-anonymised data in the form of signed consent forms and audio recordings are collected and retained as part of the research, with the audio files stored securely and then destroyed at the end of the study.

HOW WILL INFORMATION YOU PROVIDE BE RECORDED, STORED AND PROTECTED?

As already described the interview will be recorded for the purpose of transcription to text. The data will be available to me as primary researcher and to the project supervisors, David Denieffe, Dr. Yvonne Kavanagh of the Institute of Technology Carlow and Eng. Dr. Dorothy Okello of Makerere University. All data will be stored on a secure password protected hard drive. At the end of the study the audio files and transcripts will be retained for the purpose of the project assessment by Institute of Technology Carlow and then destroyed. A transcript of interviews in which all identifying information has been removed will be retained for a further two years after this. Under freedom of information legislation in Ireland you are entitled to access the information you have provided at any time.

WHAT WILL HAPPEN TO THE RESULTS OF THE STUDY?

The results of this study may be the subject of a conference paper and it will form part of the end of project thesis. The results will be published as part of this research.

WHO SHOULD YOU CONTACT FOR FURTHER INFORMATION?

If you need any further information now or at any time in the future, please contact: .

If you have concerns about this study and wish to contact an independent person, please contact:
Phone: +353 59 917 5000

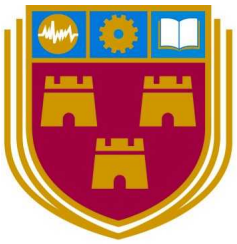
Dorothy Okello, Makerere University, Uganda | e-mail: dkokello@cedat.mak.ac.ug
David Denieffe, Vice-President for Academic Affairs and Registrar | e-mail: david.denieffe@itcarlow.ie
Yvonne Kavanagh, Assistant Registrar | e-mail: yvonne.kavanagh@itcarlow.ie

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

Signature of Participant _____ Date: _____

Printed Name of Participant:

Signature of Witness _____ Date: _____



INSTITUTE of TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

Enabling models of Internet eXchange Points to support spatial planning: the case for East Africa

Appendix B2

Political economy study

└ Internet Survey

Institute of Technology, Carlow

Challenges to the delivery of future Internet applications to the regions of developing nations

This survey can optionally be completed online at: **https://tinyurl.com/internet-africa**

Who am I and what is this study about

My name is Diarmuid Ó Briain and I am a Ph.D. candidate at the Institute of Technology, Carlow in Ireland but based at the College of Engineering, Design, Art & Technology, Makerere University in Uganda. I am working on a research project to investigate the potential impact of future networking paradigms to the delivery of networked applications in developing countries, particularly as they apply to Internet eXchange Points (IXP). The purpose of this research study is to explore your experience of the delivery of the Internet today in Africa and your thoughts on the Internet of tomorrow. The project aims to understand where the inter-networking technology is in Africa now and what are the options for future development given the regional and national infrastructure requirements. Your responses will form part a research project designed to address this.

Will taking part be confidential

This element of my research is part of a qualitative study and as such the credibility of the work is very important. It is also subject to the stringent requirements of the Institute Ethics policy. This cover page is not part of the study itself and none of the questions on pages 1 – 3 can identify the participant. All of the responses will be blended together and the anonymised results form part of the study alongside interviews that were carried out with various technology leaders across the East African Community (EAC) in particular. Below I ask for your contact details. This is optional but I ask you to please fill it out. The details will ONLY be held by me so that I can (a) verify who is responding from a credibility perspective, so I can stand over my work and (b) in case I wish to clarify an answer with you afterwards. Once this part is completed I will move the data from pages 1 – 3 into digital form and these hardcopies will be destroyed.

Who should I contact for further information?

If you need any further information now or at any time in the future, please contact me at:

diarmuid.obriain@itcarlow.ie

If you have concerns about this study and wish to contact an independent person, please contact:

Phone: +353 59 917 5000

Dorothy Okello, Makerere University, Uganda | e-mail: dkokello@cedat.mak.ac.ug
David Denieffe, Vice-President for Academic Affairs and Registrar | e-mail: david.denieffe@itcarlow.ie
Yvonne Kavanagh, Assistant Registrar | e-mail: yvonne.kavanagh@itcarlow.ie

Personal information

Family Name	Given Name

Phone number	Email

gamecore
engaging people with technology

netLabs!UG

African Internet, Political-Economy Study

Instructions

There are 7 sections to the survey and in each a number of options are presented. Questions are either:

(1) ☐ Multiple choice, pick **only one answer** from a range of options by placing a mark on the preferred option ☑

(2) Yes | No | Maybe questions select **one preferred option** and **if** it is either ***Yes*** or ***Maybe*** then feel free to specify why you made that choice in the box provided.

(3) Vote ☐ for **three** of the given options **in order of priority**, 1, 2 and 3. 3 ☐ 1 2 ☐ ☐

Education and Job function

The purpose of this section is to discover the education level and type of Internet practitioners across the region.

Nationality - Select your nationality.

☐ Ugandan ☐ Kenyan ☐ Rwandan ☐ Tanzanian ☐ Burundian ☐ South Sudanese ☐ South African
☐ Batswana ☐ Mosotho ☐ Malawian ☐ Mozambican ☐ Namibian ☐ Zimbabwean ☐ Zambian
☐ Other African _____ ☐ Other _____

Country of Employment - Select the country where your job is based.

☐ Uganda ☐ Kenya ☐ Rwanda ☐ Tanzania ☐ Burundi ☐ South Sudan ☐ South African
☐ Botswana ☐ Lesotho ☐ Malawi ☐ Mozambique ☐ Namibia ☐ Zimbabwe ☐ Zambia
☐ Other African _____ ☐ Other _____

Education Level - Select the education level that most appropriately matches your tertiary education.

☐ Doctorate ☐ Masters ☐ Honours Degree ☐ Diploma ☐ Certificate ☐ Apprenticeship ☐ Other

Education Discipline - Select the discipline that most appropriately matches your tertiary education.

☐ Computer Science ☐ Computer Engineering ☐ Telecommunications Engineering ☐ Information Security
☐ Other computing related discipline ☐ Other non computing related discipline

Current job role - Select the jobrole that most appropriately matches your current employment.

☐ Engineering Manager ☐ Project Manager ☐ Engineer ☐ Technician ☐ Other _____

Current employer type - Select the most appropriate employer type.

☐ Traditional Telecom Provider ☐ Internet Service Provider ☐ Mobile Network Operator ☐ Data Centre
☐ Internet eXchange Point ☐ Cloud Provider ☐ Content Provider ☐ Enterprise
☐ Education Network ☐ Academic / Research ☐ Government entity ☐ Vendor
☐ Application Provider ☐ Integrator ☐ Reseller ☐ Other

Experience - How many years have you worked in industry.

☐ 1 – 5 years ☐ 6 – 10 years ☐ 11 – 15 years ☐ 16 – 20 years ☐ 21 – 25 years ☐ Greater than 25

The Internet today in Africa

What are your thoughts on how the Internet is delivered across the region today?

The state of the Internet in the region over the last 10 years - Select your opinion.

☐ Very improved ☐ Slightly improved ☐ No Change ☐ Slightly disimproved ☐ Very disimproved

The key catalysts for change on the Internet has been:
Select your 1, 2 and 3 key catalysts for change over the last 10 years.

Vote (only 1 of each) 1 2 3

☐	☐	☐	☐	☐	☐	☐
Submarine cables landing in Africa	National terrestrial fibre networks	Fibre to the Home (FTTH)	Mobile Network Operators	Satellite availability	Cloud services	Internet eXchange Points (IXP)

African Internet, Political-Economy Study

The nature of traffic on the African Internet

What are your thoughts on the direction traffic on the Internet is taking, does this present problems in the future?

Streaming video services in the region over the last 10 years are: - Select your opinion.

☐ Very improved ☐ Slightly improved ☐ No Change ☐ Slightly disimproved ☐ Very disimproved

In your opinion what has had the most significant impact on video services in the region over the last 10 years? - Select your opinion.

Vote (only 1 of each) 1 2 3

☐ Caching/Content Delivery Networks (CDN) ☐ Content Provider WAN links ☐ Internet Service Providers (ISP) ☐ Mobile Network Operators (MNO) ☐ IXPs ☐ Government Regulation

There is more traffic today carried on less Autonomous Systems (AS) than 10 years ago, why? - Select your opinion.

Vote (only 1 of each) 1 2 3

☐ Cloud services ☐ IXPs ☐ Large Content Providers ☐ Service Provider mergers ☐ Government regulation

This trend of more traffic carried by less and less AS will cause problems in the future? - Select your opinion.

☐ Yes ☐ No ☐ Maybe if ('Yes' | 'Maybe'): WHY ?

Internet eXchange Points (IXP)

Where do you see the IXPs within the Internet ecosystem today and have they a place in the future?

IXPs have had a positive effect on the Internet in the region over the last 10 years: - Select your opinion.

☐ Strongly Agree ☐ Agree ☐ Undecided ☐ Disagree ☐ Strongly Disagree

IXPs have: - Select your opinion.

☐ no future, they will be obsoleted by cloud based services. ☐ a future to share local content. ☐ a future as a location for CDNs. ☐ a future continuing to act as a point where ISPs peer. ☐ a future continuing to act as a point where ISPs & Application Service Providers (ASP) peer.

☐ I don't know / No preference

IXPs should be ran: - Select your opinion.

☐ by Government ☐ by the local ISP Association ☐ by an organisation/entity of peering members at the IXP ☐ as a private business

☐ I don't know / No preference

In the future there will be a need for the establishment of IXPs in regional towns apart from national capitals: - Select your opinion.

☐ Yes ☐ No ☐ Maybe please expand on your answer

African Internet, Political-Economy Study

Software-defined

A trend towards “Software-defined”, networks and functions is now emerging, how do you think this will shape the Internet of the future?

Software Defined Networking (SDN) is pretty much hype and nothing will change? - Select your opinion.

☐ Strongly Agree ☐ Agree ☐ Undecided/ Don’t know ☐ Disagree ☐ Strongly Disagree

Network Functions Virtualisation (NFV) is the last throw of the dice for Operators, they have lost the application layer battle? - Select your opinion.

☐ Strongly Agree ☐ Agree ☐ Undecided/ Don’t know ☐ Disagree ☐ Strongly Disagree

Internet Evolution in Africa

How do you see the evolution of the Internet in the future across the region?

The top three technology drivers of the Internet evolution over the next 5 to 10 years in the region will be:

Vote (only 1 of each) 1 2 3

☐ FTTH	☐ 5G NR	☐ Content caching	☐ New satellite developments	☐ 4G Long Term Evolution (LTE)
☐ Government Regulation	☐ FINincial TECHnology (FINTECH)	☐ Operator convergence	☐ Cloud provider convergence	☐ Social media

Internet ecosystem change

What changes do you think are necessary in the overall ecosystem to facilitate this evolution ?

What are the top three changes you think are necessary in the ecosystem to facilitate Internet evolution in the region in the next 5 to 10 years?

Vote (only 1 of each) 1 2 3

☐ FTTH implementation	☐ Creation of local content	☐ Deployment of more CDNs	☐ Regulation of the 'big' networks; Google, Microsoft, Facebook, etc.	☐ Pan regional regulation
☐ Cheaper access devices	☐ Government facilitate infrastructure build	☐ Village and town hotspots	☐ Cheaper power	☐ 5G NR and beyond	☐ More IXPs

Abbreviations

4G LTE	4G Long Term Evolution	FTTH	Fibre to the Home
5G NR	5G New Radio	ISP	Internet Service Providers
AS	Autonomous System	IXP	Internet eXchange Points
ASP	Application Service Providers	MNO	Mobile Network Operators
CDN	Content Delivery Networks	NFV	Network Functions Virtualisation
EAC	East African Community	SDN	Software Defined Networks
FINTECH	FINincial TECHnology	WAN	Wide Area Networks