Ó Briain, Diarmuid; Denieffe, David; Okello, Dorothy; Kavanagh, Yvonne

Article

Enabling models of Internet eXchange Points for developing contexts

Development Engineering

Provided in Cooperation with:
Elsevier

Suggested Citation: Ó Briain, Diarmuid; Denieffe, David; Okello, Dorothy; Kavanagh, Yvonne (2020) : Enabling models of Internet eXchange Points for developing contexts, Development Engineering, ISSN 2352-7285, Elsevier, Amsterdam, Vol. 5, pp. 1-8, https://doi.org/10.1016/j.deveng.2020.100057

Development Engineering 5 (2020) 100057

Contents lists available at ScienceDirect

Development Engineering

journal homepage: http://www.elsevier.com/locate/deveng

Enabling models of Internet eXchange Points for developing contexts

Diarmuid Ó Briain [a,*], David Denieffe [a], Dorothy Okello [b], Yvonne Kavanagh [a]

[a] engCORE, Institute of Technology Carlow, Ireland
[b] netLabs!UG, Makerere University, Kamlala, Uganda

ARTICLE INFO

Keywords:
Internet
Internet eXchange Point
Software Defined eXchange
Software Defined Networking

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 (Graham et al., 2015). 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 (NFV) 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 (CDN), 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 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.

1. Introduction

Very little changed in computer networking and telecommunications during the migration of Information Communications Technology (ICT) to cloud computing during the last decade with the exception of improvements in speed. However, disruptive technologies in the form of SDN and NFV are beginning to make their presence felt. These technologies have established the concept of elastic network and elastic functions which complement the elastic compute and elastic storage technologies underpinning the earlier cloud computing transformation. An SDN enabled elastic network is programmable through centralised management of the control plane to endow it with flexibility and dynamism, to achieve improved performance, that is not possible with traditional network management techniques. Elastic functions, enabled through NFV style architectures, leverages virtualisation and containerisation to enable nodes to be developed, as required, when required, and chain them together to achieve a larger purpose within a system.

Another change that has occurred quietly is a flattening of the Internet. Traffic patterns have switched from static text and picture based content to video which now accounts for over 75% of all traffic today (Bly, 2018). Of that, over half is strategically placed on CDNs located at local Internet Service Provider (ISP) and 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% (Cisco and Cisco Visual Netwo, 2019). 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

* Corresponding author.
E-mail address: diarmuid.obriain@itcarlow.ie (D. Ó Briain).

https://doi.org/10.1016/j.deveng.2020.100057
Received 2 June 2020; Received in revised form 23 September 2020; Accepted 19 October 2020
Available online 24 October 2020

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 dilemma (Global indicator fr, 2017). In the context of this larger problem, IXPs form a key component for the delivery of the Internet outside of capital cities. In developing countries the scale of IXPs, even regional IXPs in secondary cities and towns, are many times the size of IXPs in capital cities in East Africa. This scale provides the business case to fund their development while in East Africa the development of regional or local IXPs has not always been commercially and/or technically viable. South Africa provides an example for comparison where 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), (The peering database (pee). Thus the development of regional IXPs can remain hidden among the many other challenges to infrastructure provision associated with spatial planning.

1.1. Focus and scope

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. The experience of having worked at the Uganda Internet eXchange Point (UIXP) posits the question; *is the IXP in Kampala enough?* and *do not other towns and cities also deserve the facility of an IXP?* These questions form the root of a project to develop a PoC artefact to demonstrate the feasibility of a distributed set of IXPs (dIXP) 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 traditional IXP or Software Defined eXchange (SDX) modes. Skills gaps as well as the scalability of a set of IXPs, forming a dIXP, are catered for through a centralised management model. Future proofing is considered through the incorporation of SDN such that the IXP can also be deployed in a SDX mode.

1.2. Research questions

Regional cities in developing countries deserve a fully developed Internet ecosystem that includes an active IXP formed the vision for the project, it was broken down into two research questions and are considered via a build-evaluate PoC cycle. Each cycle 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 drove 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.* Additionally, to resolve potential skills issues at remote IXPs, a mechanism for the day-to-day management to be carried out centrally is necessary. This created a requirement for *a method within the PoC for centralised management of the set of IXPs within a dIXP could be identified.* Once operational, consideration as to the potential for the inclusion of software-defined technologies, in the PoC, offering the basis for new services lead to the next question,

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

Which established the objective to *evaluate if SDN can be incorporated into the IXP models to create an SDX, to list the possible benefits and suggest solutions based on the East African context.*

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.

2. Research design

Building on the methodological principles defined in Action Design Research (ADR), Laboratory based ADR (LADR) and Design Science Research Methodology (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:1, (Sein et al., 2011; Ralph, 2014; Peffers et al., 2007).

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 SDG (Global indicator fr, 2017). In the context of this larger problem, this research focused on the IXP as a key component for the delivery of 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 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 as *IXPBuilder* in an operations manual as well as further documentation of the internal workings of the *IXPBuilder* PoC software. Conclusions and recommendations, for future work, were made from the overall research.

3. The PoC testbed

3.1. Functional specification

A high level functional specification was considered and it was identified that the resultant PoC must; (a) *be simple to build and maintain*, (b) *be capable of* supporting *different site scenarios*, (c) support *a distributed model for management while maintaining independence for peering* and (d) *incorporate SDN such that the IXP can operate as an SDX*.

3.2. Testbed development

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. 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 mIXPs from a centralised cIXP. Figure:2 is a reference tool to aid description. It outlines two sites, a cIXP and a remote mIXP with the

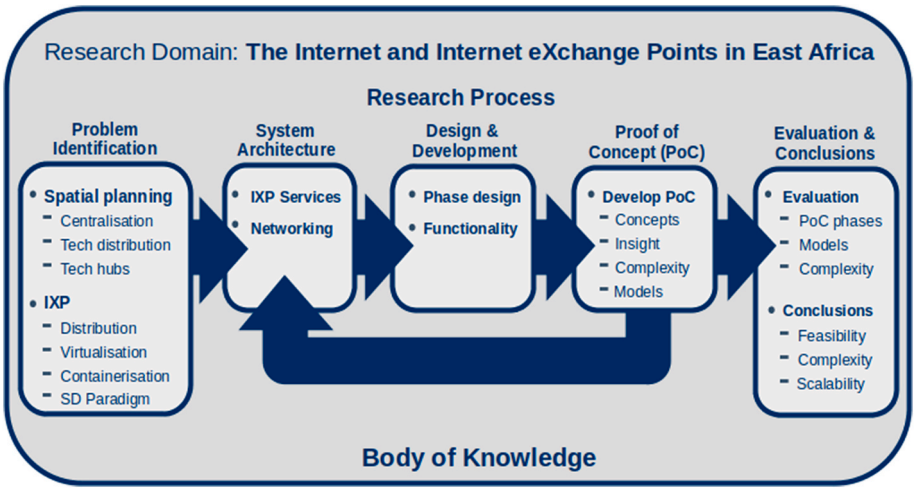

Fig. 1. Research design.

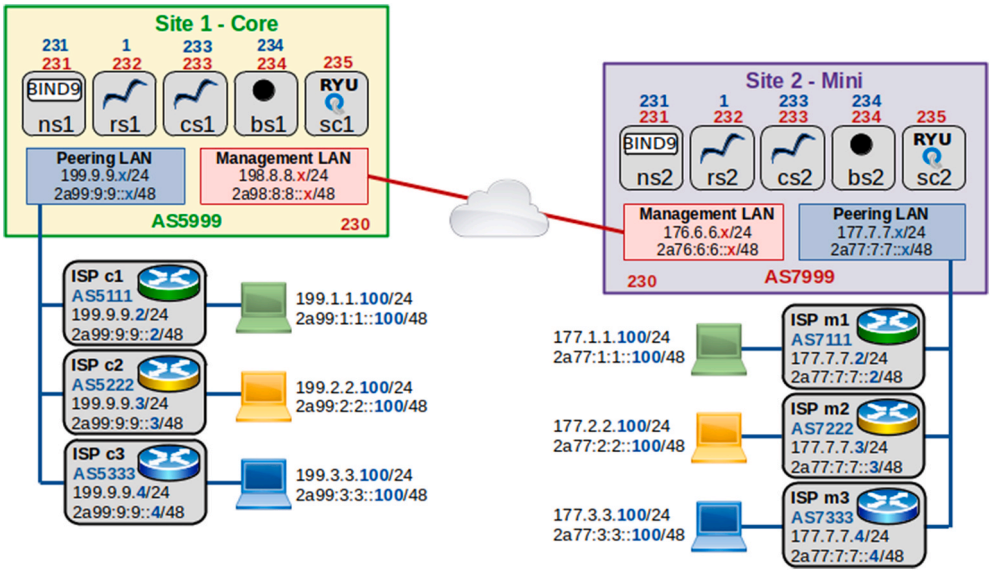

Fig. 2. IXPBuilder testbed.

various Internet Protocol (IP) prefixes and addresses used at each site. In the case of models employing traditional Ethernet switches there is no SC, the schema has addresses assigned but containers *sc1* or *sc2* are not built except in software-defined models.

4. The PoC system architecture

The PoC system architecture at each IXP node is illustrated in Figure:3 and it is 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

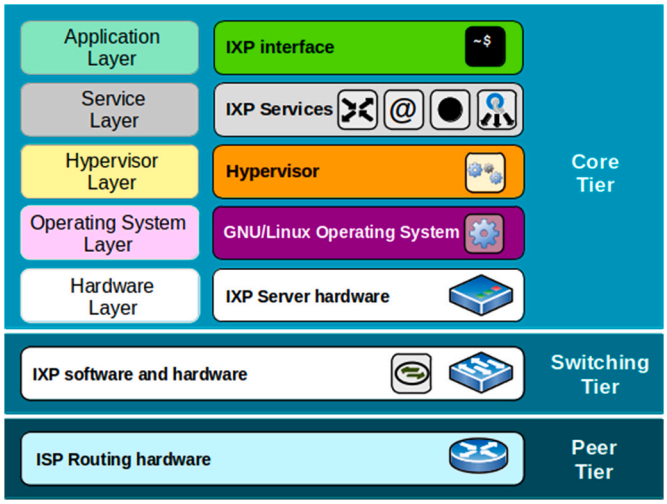

Fig. 3. PoC architecture.

which interfaces with an IXP library module, *ixp.py* to build and manage the IXP. The PoC build phases consisted of five iterations.

4.1. The baseline

There are many interdependent elements to the IXP and the PoC must be capable of building 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. The initial iterations of the development of the PoC involved the building of a basic IXP system with IXP peering members, a GNU/Linux OS hosting both a Route Collector Server (CS) function, to collect routes for statistical purposes and a Route Server (RS), to exchange routes between members with an open peering policy (Jasinska et al., 2017). Both functions are based on the BIRD Internet Routing Daemon (BIRD) (Filip et al., 2018). Configurations were developed for a number of popular routers from Cisco, Juniper, MikroTik and Netgear at the peer tier. This established a baseline peer tier for later iterations when the focus shifted to development at the switching and core tiers.

A second iteration of the baseline focused on the virtualisation of the service layer of the core tier, initially using the VirtualBox Virtual Disk Image (VDI) format with virtualisation in the form of Virtual Machines (VM) like the Kernel VM (KVM); however, this was ultimately replaced by containerisation. LXD and LXC were 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 ("LXD crushes KVM in density and speed," 2015).

4.2. The switching tier

The next set of iterations, were characterised by significant experimentation, particularly at the switching tier, building on the baseline phase, with the addition of two Linux Bridges to facilitate the creation of two Virtual Local Area Networks (VLAN), one for management and the second for peering. This facilitated the separation of the management and member device peering traffic at the peer tier. Data from these bridges combined on one VLAN trunk interface to the managed Ethernet switch with VLAN tags identifying the source of the different frames. Configurations were produced for a number of popular managed Ethernet switch OS, at the switching tier, from Cisco, Juniper, MikroTik and Netgear.

4.3. Service layer functions

Additional functions, as LXCs, were added to the service layer in the core tier. In addition to the RS and CS functions, the IXPs are expected to have have a Domain Name System (DNS) server (NS) and an AS112 Blackhole Service (BS) to deal with DNS reverse lookup queries for IP addresses which are part of private address space. The Berkeley Internet Name Domain (BIND) from the Internet Systems Consortium (ISC) was employed for the DNS function in the NS and joined the BIRD function in the BS (Mockapetris, 1987; Abley and Sotomayor, 2015).

An operations, python3 functional library, module called *ixp.py* and a python3 based CLI was developed to incorporate automation functionality as well as an installation tool *ixp-install.sh*. The operations program supports the building of an IXP schema which is used to identify, install and configure the required software packages.

4.4. A revisit to the switch

The single interface between the computer and the managed traditional Ethernet switch was identified as a limiting factor and it was necessary to acquire a Common Of The Shelf (COTS) server hardware with multiple Ethernet interfaces. Replacing the Linux Bridge functionality with Open virtual Switch (OvS) became necessary. OvS could still be used to maintain traditional Ethernet switching models but could also be used to explore the incorporation of the SDN paradigm to develop an SDX.

4.5. IXP models to operate with traditional switches

The addition of OvS lead to a new iteration of the research design focused on the development of new traditional Ethernet models of operation. These models allow the PoC to operate with traditional managed Ethernet switches. Earlier iterations had been based on single and low Ethernet interface numbers on the hardware so peers were connected to external managed Ethernet switches in what was named model 'A' for a single interface and 'B' for a two interface server hardware. The COTS server acquired for the hardware layer incorporated eight Ethernet interfaces and facilitated the development of new models which reserved the first two interfaces for In Band and Out Of Band (OOB) management functions. A 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. 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. A hybrid model 'D', as illustrated in Figure:4, provides flexibility with some interfaces providing connections for managed traditional Ethernet switches and the remainder facilitating the direct connection of peers. These models are summarised in Table:1.

4.6. Remote management of mIXPs from the core

A facility to manage remote mIXPs from the cIXP within the dIXP set was also necessary. As management traffic levels are small, a facility that securely traverses any network connection, such as the open Internet, was created within the IXP library module *ixp.py*. The functionality is operated through additional commands made available on the cIXP that enable the monitoring and management of the day-to-day operations of local mIXP PoC instances. This communication is secured using a Rivest, Shamir and Adleman (RSA) pubic/private key authentication mechanism. 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.

4.7. SDN models to create an SDX

Another set of models were created to incorporate SDN functionality which effectively turned the PoC into an SDX. To facilitate this, an SC function, based on Ryu, a python3 component-based SDN framework was added at the service layer. This involved the development of a new class within the IXP library module *ixp.py* as well as an SDX sub-class of the Ryu framework *ryu.base.app_manager* class.

This new sub-class, called *ixp_switch_13*, provides 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 OF switches in the switching tier as shown by the thick dashed black lines in Figure:5. 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. It should be noted that the SC has no port on the peering LAN, it controls OF switches via the control-channel over management LAN.

Of the new SDX models, model 'S' offers a similar level of support, for smaller sites, as model 'C'. Model 'U' is considered similar to model 'E' as

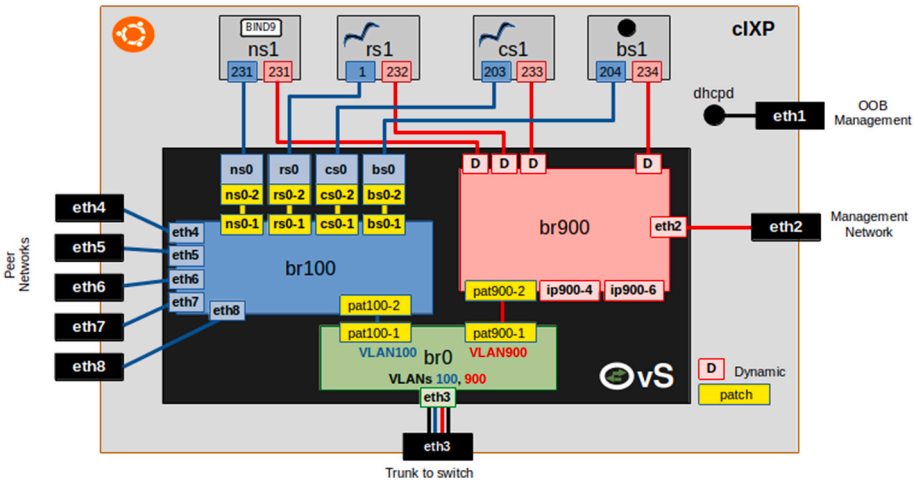

Fig. 4. Traditional switching, Model D block diagram.

Table 1
Summary of IXP models.

Model	Switch	Network Interfaces	Interface(s)				
			1	2	3	4	5+
A	Traditional	1	Trunk				
B	Traditional	2	OOB	Trunk			
C	Traditional	>3	OOB	Mgmt	Peers		
D_1	Traditional	>3	OOB	Mgmt	Trunk	Peers	
D_2	Traditional	>3	OOB	Mgmt	Mix of Peers & Trunks		
E_1	Traditional	3	OOB	Mgmt	Trunk		
E_2	Traditional	>3	OOB	Mgmt	Trunks		
S	OF	>3	OOB	Mgmt	Peers		
T_1	OF	>3	OOB	Mgmt	OF 1 CDIP	Peers	
T_2	OF	>3	OOB	Mgmt	Mix of CDIPs & Peers		
U	OF	>3	OOB	Mgmt	CDIPs		

it facilitates the connection of external OF Ethernet switches and model 'T', whose OvS internals are portrayed in Figure:6, offers a mix of options in a similar way to model 'D'. It can be seen that the SC has a physical interface in the management bridge, *br900*; however, it manages *br100*, the peering LAN, through the management connection on TCP port 6633. However, these SDX models offer further functionality as the SC hosts a Representational State Transfer (REST) Application Programming Interface (API) *ryu.app.ofctl_rest* which permits the manipulation of tables in both internal and external OF Ethernet switches. This interface is called a North Bound Interface (NBI). 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. These SDX models are also summarised in Table:1 alongside the traditional models.

4.8. Summary of IXP and SDX models

A summary of all 11 models and sub-models developed is outlined in

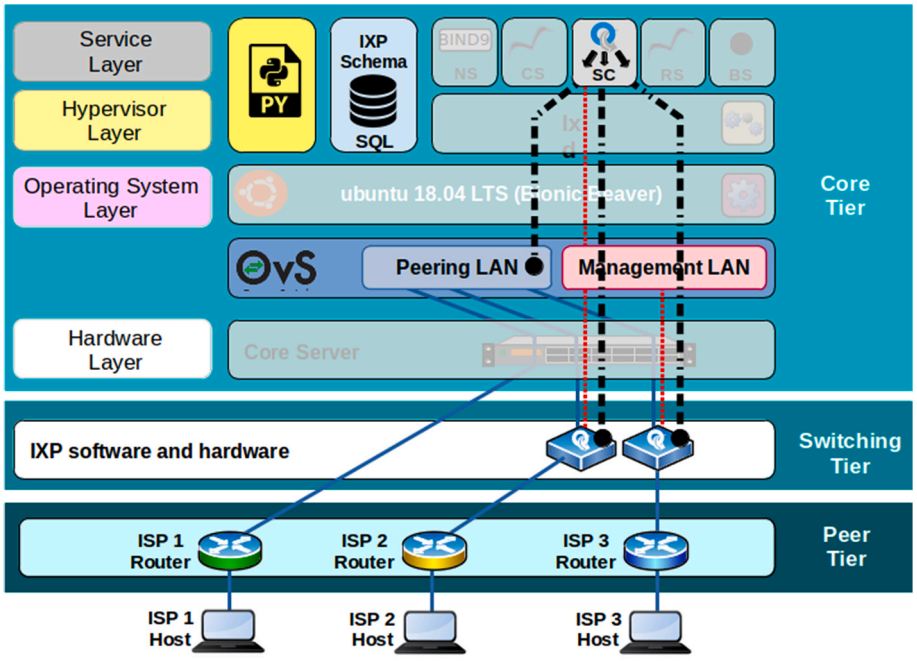

Fig. 5. PoC SDX architecture.

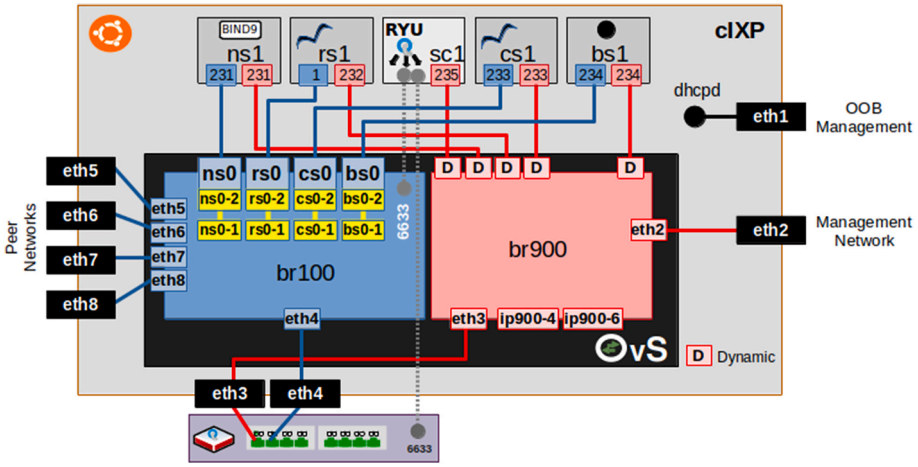

Fig. 6. Software-defined switching, Model T block diagram.

Table:1. 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, readily available, traditional managed Ethernet switches further enhances the ease of implementation and affordability of these systems.

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. PoC testing and results

Testing of the PoC was two fold, functionality and usability testing. Communications between end-user hosts within an IXP and between end-user hosts at different IXP nodes. As the ISP nodes in the testbed do not have a transit mechanism as a fallback to the IXP substrate, end-users within an IXP node should have connectivity and between IXP nodes should not.

A usability test was carried out by giving the PoC software, *IXPBuilder*, along with the *IXPBuilder* manual to two separate groups of undergraduate BSc in Telecommunications Engineering students at the College of Engineering, Design, Art and Technology (CEDAT), Makerere University to test the install-ability and the ease of build of the PoC. The PoC demonstrates that through the containerisation of functions, with an OvS and novel IXP models implemented through custom software, *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 via a NBI. 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* that would be created if the very ISPs that made these IXPs successful would now consider the IXP as competition in the transit and back-haul space, while also resolving any skills gap, that may exist, in towns hosting an mIXP through centralised management from the cIXP.

5.1. 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 test network in Figure:2 a connectivity test was completed between each subscriber host for both IPv4 and IPv6. This has been summarised in Table:2. Before testing, three peers were established on each IXP, connected to the CS, RS and BS.

Hosts of each ISP 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 does not occur through the IXP.

5.2. Usability testing

A key requirement of the PoC from the functional specification is that it is 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, was given to two separate groups of undergraduate BSc in Telecommunications Engineering students at the CEDAT, 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 an early build of *IXPBuilder* software and the associated *IXPBuilder* manual. 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 the final

Table 2
Host connectivity test.

Host	C1		C2		C3		M1		M2		M3	
	v4	v6	v4	v6	v4	v6	v4	v6	v4	v6	v4	v6
C1			✓	✓	✓	✓	×	×	×	×	×	×
C2	✓	✓			✓	✓	×	×	×	×	×	×
C3	✓	✓	✓	✓			×	×	×	×	×	×
M1	×	×	×	×	×	×			✓	✓	✓	✓
M2	×	×	×	×	×	×	✓	✓			✓	✓
M3	×	×	×	×	×	×	✓	✓	✓	✓		

version of *IXPBuilder* and the corresponding manual 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; however, both were, initially, unable to reach *ISP3*; however, after some troubleshooting the team rectified the problem and all clients passed connectivity tests to each other.

6. Discussion

Using the design specification the PoC was developed over five iterations. 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 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. The results indicate that the PoC is both useable 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 OF Ethernet switches. Through the selection of appropriate COTS hardware to host the PoC as well as through leveraging free of cost open source software, the PoC can also be considered affordable.

7. Conclusion

7.1. IXP Proof of Concept

The 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.

Additionally, having developed stand-alone IXP models, remote IXPs can be centrally managed without interfering with their independence from a peering perspective through 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 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. 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 based Ethernet switches from an SC or in the case of a small SDX, the management of the internal OF switch. Adding SDN functionality also exposes a new RESTful 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 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. 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 few 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 and P4 language specifica, 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 developing and open source implementation of P4 called Stratum (O'Connor, 2018). 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.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abley, J., Sotomayor, W., May 2015. RFC7534. AS112 Nameserver Operations.
Bly, R.W., 2018. The Digital Marketing Handbook: A Step-by-step Guide to Creating Websites that Sell. Entrepreneur Media Incorporated/Entrepreneur Press.
Cascone, C., Sep. 2018. P4 and P4Runtime Technical Introduction and Use Cases for Service Providers.
Cisco, Cisco Visual Networking Index: Forecast and Trends, Feb. 2019, 2017–2022.
U.N., Global Indicator Framework for the Sustainable Development Goals and Targets of the 2030 Agenda for Sustainable Development, Jul. 2017.
Jasinska, E., Hilliard, N., Raszuk, R., Bakker, N., Sep. 2017. RFC7947. Internet Exchange BGP Route Server.
Lodhi, A., Larson, N., Dhamdhere, A., Dovrolis, C., Claffy, K., 2014. Using Peeringdb to Understand the Peering Ecosystem Vol. vol. 44, ACM..
LXD crushes KVM in density and speed, 2015. Canonical.
Mockapetris, P., Nov. 1987. RFC1034. Domain Names - Concepts and Facilities.
O'Connor, B., Dec. 2018. Stratum: an Overview.
P4, P4 Language Specification, Nov. 2018.
Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S., 2007. A design science research methodology for information systems research. J. Manag. Inf. Syst. 24 (3), 45–77.
Ralph, P., 2014. Lab-based action design research. In: Companion Proceedings of the 36th International Conference on Software Engineering. ACM, pp. 528–531.
Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R., 2011. Action design research. MIS Q. 37–56.
Shahbaz, M., Choi, S., Pfaff, B., Kim, C., Feamster, N., McKeown, N., Rexford, J., 2016. Pisces: a programmable, protocol-independent software switch. In: Proceedings of the 2016 ACM SIGCOMM Conference. ACM, pp. 525–538.
Filip, O., Machek, P., Mares, M., Zajicek, O., 2018, BIRD User's Guide.
The peering database (peeringdb). https://www.peeringdb.com.

Further reading

Graham, M., Andersen, C., Mann, L., 2015. Geographical imagination and technological connectivity in east africa 40 (3), 334–349.