Responding to the demands of big data scientific instruments through the development of an international software defined exchange point (SDX)

Julio Ibarra[1], Jeronimo Bezerra[1], Luis Fernandez Lopez[1,2], Heidi Morgan[3], Donald Cox[4]
[1]Florida International University, 11200 SW 8 St, Miami, Fl. 33199
Tel: +1-305-348-4105; email: Julio@fiu.edu, jbezerra@fiu.edu, llopez@fiu.edu
[2]Academic Network of São Paulo (ANSP), lopez@ansp.br
[3]University of Southern California, Los Angeles, USA, hlmorgan@isi.edu
Vanderbilt University, Nashville, USA, chip.cox@vanderbilt.edu

Abstract

Science is being conducted in an era of information abundance. The rate at which science data is generated is increasing, both in volume and variety. This phenomenon is transforming how science is thought of and practiced. This transformation is being shaped by new scientific instruments that are being designed and deployed that will dramatically increase the need for large, real-time data transfers among scientists throughout the world. One such instrument is the Square Kilometer Array (SKA) being built in South Africa that will transmit approximately 160Gbps of data from each radio dish to a central processor.

This paper describes a collaborative effort to respond to the demands of big data scientific instruments through the development of an international software defined exchange point (SDX) that will meet the network provisioning needs for science applications. This paper discusses the challenges of end-to-end path provisioning across multiple research and education networks using OpenFlow/SDN technologies. Furthermore, it refers to the AtlanticWave-SDX, a project at Florida International University and the Georgia Institute of Technology, funded by the US National Science Foundation (NSF), along with support from Brazil's NREN, Rede Nacional de Ensino e Pesquisa (RNP, and the Academic Network of Sao Paulo (ANSP). Future work explores the feasibility of establishing an SDX in West Africa, in collaboration with regional African RENs, based on the planned availability of submarine cable spectrum for use by research and education communities.

Keywords –

Software-Defined Networking; Software-Defined Exchange; Science Data Applications

1. Introduction

New scientific instruments are being developed in the southern hemisphere that will increase the need for large, real-time data transfers among scientists throughout the world. The Large Synoptic Survey Telescope (LSST) being built in Chile will produce 6.4 GB images that must be transferred to the U.S. in 5 seconds. The Square Kilometer Array (SKA) in South Africa will transmit approximately 160 Gbps (Gigabits per second) of data from each radio dish to a central processor. A Science Data Processor (SDP) receives data streams potentially as high as 3 Tbps (Terabits per second). The SDP is responsible for producing ready-for-science data products, which are then distributed to regional science centers for analysis.

Simultaneously, a parallel activity is the construction of new fiber-optic submarine cable systems in the South Atlantic, directly linking South America and Africa along the southern hemisphere. The South Atlantic Cable System (SACS) links Fortaleza, Brazil to Luanda, Angola. SACS is under construction and is scheduled to be ready for service by third quarter 2018. The South Atlantic Inter Link (SAIL), formerly CBCS, links Fortaleza, Brazil to Kribi, Cameroon. SAIL is scheduled to be ready for service in 2018.

Currently, most science data flows from Africa's southern hemisphere, destined to either Europe or the Americas, are transported north to either London or Amsterdam. Science applications that are delay sensitive are impacted, because of network latency, caused by distances across continents and oceans, and the number of network segments, involving multiple academic networks along the path, operated by different organizations. These new submarine cables will enable the construction of new network paths to potentially link the research and education communities in the southern hemisphere of Africa to Brazil, and other nations in South America, North and Central America, and the Caribbean. This is significant to big-data-generating science instruments like LSST and SKA that require large capacity bandwidth for high-throughput applications, and lower latency for delay-sensitive applications.

The network requirements for the SKA are significant. SKA literature refers to the following four types of network services (Ref): Science Data, Sync and Timing, Non-science data, and External connections. The network service for Science Data is required to provide high-throughput network transport, in order to move thousands of Gigabits of data per second. The network service for Sync and Timing requires low latency, high priority and low bandwidth. The network service for Non-science data is expected to carry a variety of network traffic types: Live observation critical data; testing, diagnostic and commissioning data; monitoring and control information; and general-purpose communications traffic (e.g., IP telephony). Finally, the network service for External connections will be required to support multiple 100G connections to external networks.

To achieve the aforementioned requirements, the end-to-end network path should provide high resilience, low delay, multiple paths, high bandwidth and an efficient control plane to act in all status changes (i.e., port status, devices outages, etc). Traditional networks possess limitations, such as sub-optimal resource utilization, forwarding based on destination MAC or IP address, etc. The operational complexity of managing different administrative domains, topologies, link technologies, devices, and requirements is challenging when using traditional network operations methodologies for provisioning, monitoring and operating networks. The AtlanticWave-SDX project aims to develop a capability to support applications, such as the LSST and the SKA, that have intensive network resource requirements.

The end-to-end path for the LSST will be composed of different academic networks, some of them supporting SDN and network programmability. Having information about network resources and control for programmability will enable LSST and SKA applications to react to network conditions in a more efficient way, sometimes even anticipating issues. For example, a link that will flap might be detected when a CRC/loss number increases. With network programmability, SKA applications will be able to provision multiple paths dynamically and on demand, apply QoS and prioritization policies, and manipulate flows at multiple levels.

Using information made available by all network devices on the path, SKA applications will be able to select the preferred paths from among several choices for sending its traffic from South Africa to the regional science centers around the world.

The end-to-end network path for SKA science flows will most likely be provided by multiple academic networks, in multiple countries, and in most cases, these networks are interconnected at academic exchange points. To achieve end-to-end programmability and control, all academic exchange points along the path must support network-aware applications. Fortunately, exposing network control capabilities to applications within a single SDN domain is now feasible and many academic networks (e.g., AmLight, Internet2, and ESnet) provide this capability today. This is not the case for applications that must span multiple domains. Most of the current Academic Exchange Points are still using traditional methodologies for forwarding (e.g., IP or MAC-based forwarding) and control (e.g., a NOC team controlling network devices through SSH and/or SNMP).

An academic exchange point supporting network-aware application features is called a Software-Defined Exchange, and it is considered the next step in the network evolution following the SDN line of thinking. This SDX must be open, programmable and resilient. All its external interfaces must also be secure and standard to support different kinds of network-aware applications.

The remainder of this paper is organized as follows. Section 2 presents background information and a literature review of previous SDX proposals. Section 3 describes the AtlanticWave-SDX architecture. In Section 4, a policy API is presented followed by a description of security challenges and how the AtlanticWave-SDX intends to respond to these challenges are in section 5. Finally, conclusions and next steps are presented in Section 6.

2. Background

Currently, there is no single, agreed upon definition of what a Software Defined Exchange (SDX) means. The spectrum of definitions ranges from Networking Exchanges to Cloud Service Exchanges, both capable of orchestrating resources across independent administrative domains. Moreover, below the SDX definition for networks, we can have: (1) Layer-3 SDXs that provide connectivity and routing between Autonomous Systems (AS) as in the case of an Internet Exchange Point (IXP); (2) Layer-2 SDXs for multi-domain Ethernet circuits; and (3) SDN SDXs to interconnect SDN islands. Likewise, the Cloud Service SDX provides access to compute and storage resources. In the next sections, we provide examples of recent Layer-3, Layer-2 and SDN SDXs, as those are more relevant to the AtlanticWave-SDX project; Cloud Service Exchanges could be seen as Federated Clouds or Hybrid Clouds. In Figure 1, we show a taxonomy for the Network Exchanges we consider in this paper, and examples under each category.

2.1 Layer 3 SDX

A Layer-3 SDX provides connectivity between different Autonomous Systems. The main characteristic of a Layer-e SDX is that a BGP process is required to handle the exchange of BGP routes. The minimum additional requirements are a SDN fabric and a SDN controller to install flows between the participants. It is desirable that the SDX has a Policy Manager to enrich the policies beyond what can be defined with BGP. Some examples of Layer-3 SDXs are SDN-IP (Lin et al 2013), Cardigan (Stringer et al 2013; 2014) and SDX (Gupta et al, 2014), which are described in more detail next.

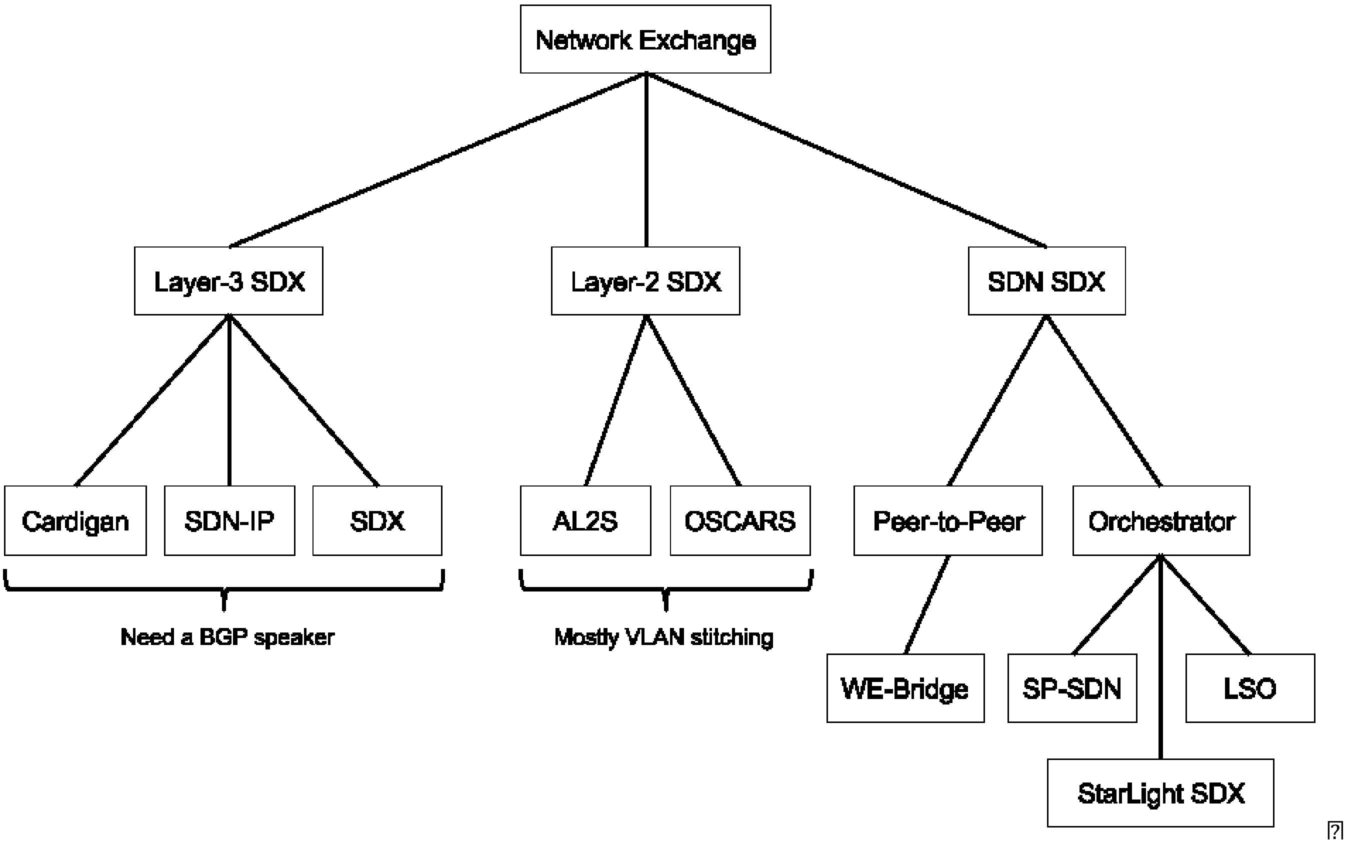

Figure 1 Network SDX Taxnomy

(Lin et al. 2013) proposed a solution to enable BGP peering between SDN and non-SDN Autonomous Systems. To achieve BGP peering, the centralized SDN control plane integrates a BGP process; turning the entire SDN AS into a single BGP router from the point of view of its peers. The solution was developed as an application in the ONOS controller, and tested using an emulated Mininet topology. Experiments tested how the number of Routing Information Base (RIB) entries affected the memory incremental cost. The authors concluded that SDN-IP could scale up to 10,000 RIB entries, processing 100 RIB updates per second.

Cardigan (Stringer et al 2013; 2014) described a distributed router based on RouteFlow and a mesh of OpenFlow switches that are represented as a single logical switch. The goal is to implement a SDN-based distributed routing fabric. Cardigan's datapath works in a full-mesh, like routers' line cards and fabric cross-connects using proactive flow installation. Cardigan was deployed connecting the Research and Education Advanced Network of New Zealand (REANNZ) to the Wellington Internet Exchange (WIX), handling 1134 flows with a TCP performance of 800Mbps approximately.

Gupta et al (2014) proposed the design, implementation and evaluation of SDX, to improve the network management capabilities of BGP participants in an Internet Exchange Point (IXP). The main idea behind SDX is to present a virtual SDX switch to each BGP participant, so they can realize high level tasks such as: application-specific peering, inbound traffic engineering, wide-area load balancing, and redirection through middle boxes all while ensuring isolation between the policies. For this solution, each participant sends its policies to the SDX controller; then the SDX engine compiles the individual policies and installs a single set of policies on the SDX switch. The authors claim that just adding a SDN switch and controller to an IXP, as in the previous examples, is not enough to realize a SDX. The first version of this SDX was implemented using Pyretic (Reich et al 2013) running on a POX controller, an enhanced version is being implemented using Pyretic and a Ryu controller (Ryu, 2015).

2.2 Layer 2 SDX

A Layer-2 SDX allows operators to create multi-domain circuits; typically using Layer-2 technologies like Ethernet VLANs. This scenario is mainly used in Research & Education Networks such as Internet2 and ESnet. For instance, Internet2's Advanced Layer 2 Service (AL2S, 2015) allows network operators to create their own Layer 2 circuits in the Internet2 AL2S backbone connection two or more endpoints. Similarly, the On-demand Secure Circuits and Advance Reservation System (OSCARS, 2015) accomplishes the same goal in the Department of Energy’s high-performance science network ESnet.

2.3 SDN SDX

The design objective of the SDN SDX is to interconnect SDN islands managed by different domains. The WE-Bridge (Lin et al, 2015) is a mechanism to enable different SDN administrative domains to peer and cooperate. WE-Bridge itself is not an inter-domain routing protocol, but a platform to exchange basic network information between different domains. The main goal is to improve inter-domain routing by announcing domain-views containing rich/fine-granularity information/policies, to enable various inter-domain innovations based on network information. This solution includes a network view virtualization, and a virtual network format and distribution using JSON. The peer relationships are established through a peer-to-peer control plane and a modified version of Link Layer Discovery Protocol (LLDP) to connect domain border switches. Contrary to the peer-to-peer approach used by the WE-Bridge, Mambretti et al. (2014; 2014) proposed a centralized Path Controller to manage the resources of federated controller in order to interconnect federated SDN islands.

Similar approaches are the Service Provider SDN (SP-SDN) (Kempf et al, 2014) and MEF’s Lifecycle Service Orchestration (LSO, 2015). Both proposals envision a service orchestration layer on top on the SDN control layer, which span different administrative domains. Some application examples presented in these projects are: elastic WAN, network slices on-demand, VPN circuits on-demand, and end-to-end Network-as-a-Service.

2.4 SDX Characteristics

As we have seen, a SDX could exchange BGP routes, Layer-2 circuits, computing and storage capacity. More generally, an important characteristic of an SDX is its ability to exchange networking, computing or storage resources in a common point, between independent administrative domains. Furthermore, the capability to apply richer policies to the exchange of these resources is another important characteristic of the SDX. Finally, in terms of security, strong isolation of constituent data and control interfaces is a desirable characteristic of a SDX.

3. Architecture

The AtlanticWave-SDX project is working to extend the SDX concept to a production deployment of a multi-domain international SDX involving initially three academic exchange points, which include SouthernLight (São Paulo/Brazil); AMPATH (Miami/USA) and SoX (Atlanta/USA). AtlanticWave-SDX will provide application users with an end-to-end service that supports the traffic policy requirements of the application across multiple Autonomous Systems and physical exchanges.

There are several alternatives for providing such an end-to-end capability. Figure 2 shows the proposed topology with three options of deployment. Option 1 assumes a single SDX controller that manages multiple IXP switch fabrics. While this approach is the simplest technical option, it is not ultimately viable in a distributed, multi-party environment. Option 2 introduces an intermediate slice manager, such as FlowVisor (Sherwood et al 2009) or Flowspace Firewall (FlowSapce Firewall, 2015), which allows individual controllers to be handed a slice of the network resources to be managed while isolating those resources from others. Option 3 creates a hierarchy of controllers with a local controller at each exchange being managed by a separate higher-level controller. We expect Option 2 to be the most practical approach for the near term and intend to focus here for the initial implementation and deployment. In this work, we are extending our previous work (Gupta et al 2014) in SDX design to include both lower layer concepts (e.g. VLAN stitching) and upper layer concepts (e.g., application-based routing, load balancing, QoS, etc). We are designing and implementing a software toolkit with APIs for application developers to tell the controller what demand they will introduce, at what times, and with what performance requirements, so the controller can plan/schedule the use of resources with prior knowledge of "when" and "what". The software developed in this project will be based on the SDX controller presented in Gupta et al (2014) and available from GitHub (Ryu, 2015).

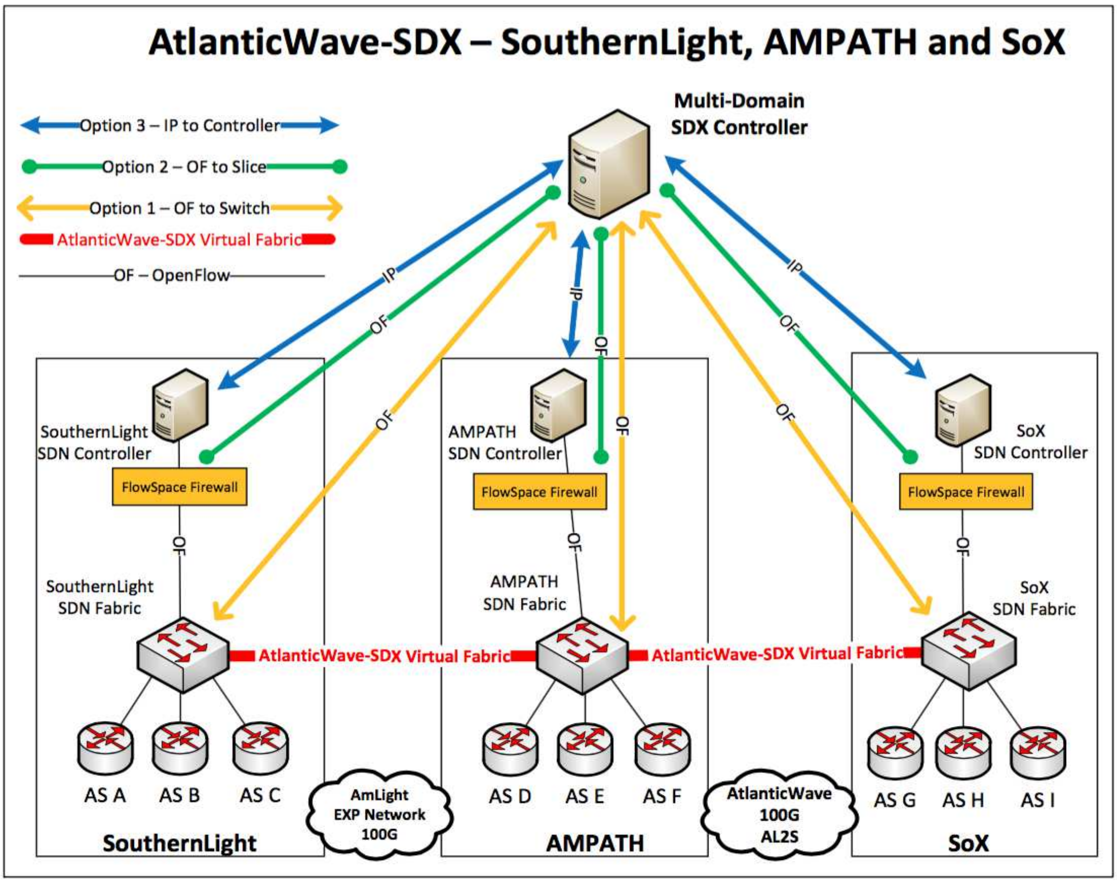

Figure 2 AtlanticWave-SDX network architecture

This software is being actively used and extended, including ongoing work to deploy it on GENI (Berman et al 2014). The AtlanticWave-SDX project includes significant effort in "hardening" this software to make it production-ready and in extending it beyond the current Pyretic-based policy language to include programmable APIs for developers that support the specific application use cases identified here.

4. Towards a Policy API for SDX

Before talking about SDX policies, it is necessary to know what kind of applications can be deployed in an SDX. In Gupta et al (2014) the authors proposed four applications: application-specific peering, inbound traffic engineering, wide-area load balancing and redirection through middle boxes. In general, the four applications match fields of the TCP/IP header and apply actions accordingly. However, in Big Data science network service requirements, such as for LSST and SKA, the application needs to comply with certain latency and bandwidth requirements. These requirements cannot be defined using only fields of the TCP/IP header or the network topology status; the SDX controller requires external information sources such as SNMP, sFlow or perfSONAR (perfSONAR, 2015) measurements.

Taking into account the conditions described previously, there are several candidates for a Policy API for SDX. In Gupta et al (2014), the authors opted for Pyretic, a high level programming language for SDN. Similarly, the ONOS controller introduced the concept of intents for network policy specification (ONOS 2015). On the other hand, WE-Bridge (Lin et al, 2015) proposed JSON as its policy API. Other valid contenders for a Policy API are RESTful and XML interfaces. To illustrate what SDX policies would look like, we present three examples: application specific peering, on-demand circuit provisioning and bandwidth calendaring.

4.1 Application Specific Peering

Consider three Autonomous Systems (A, B and C) connected to an SDX. Both B and C are advertising the same IP prefix to SDX's Route Server (See Figure 3). SDX's Route Server decides which is the best BGP path for these prefixes and advertises it to A.

In this example (Coursera 2015), routes advertised by B are preferred over C, for instance, because of the AS-path length. For example, A might want its traffic destined for port 80 (dstport 80) to go to B, while traffic destined for port 4321 or port 4322 to go to C. This policy could be implemented as follows:

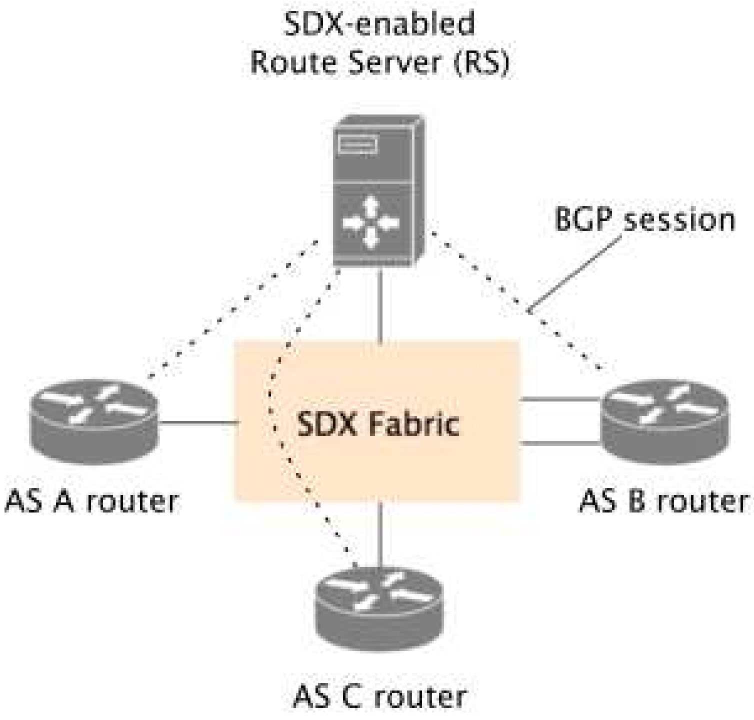

Figure 3 SDX network topology

```
if (dstport == 80)
    forward to B
else if (dstport == 4321 || dstport == 4322)
    forward to C
```

This may be implemented in Pyretic as follows:

```
match(dstport = 80) >>  fwd(B) + match(dstport=4321/4322) >> fwd(C).
```

4.2 On-Demand Virtual Circuit Provisioning

This application provides the capability of provisioning virtual circuits on demand like Internet2's AL2S and ESnet's OSCARS. However, the SDX controller could take advantage of Network Measurement Systems, such as perfSONAR, to define Service Level Agreement (SLA) compliance and elastic WAN services, enriching the policies. In this scenario the SDX policy might look like:

```
if (current_latency > SLA_latency)
    secondary = findSecondaryPath()
while (current_latency > SLA_latency)
    LoadBalance(primary, secondary)
```

The while loop represents a dynamic policy. This behavior could be represented using Pyretic Dynamic Policies, ONOS Intents, an active polling mechanism, or a reactive triggered signal coming from the Network Monitoring System (NMS) (e.g. SNMP Traps). Another option is to use state machines as proposed by Kim et al. in Kinetic (Kim et al). The ideal scenario will be as follows: (1) The application sets an SLA (i.e. latency less than 10ms and packet loss lower than 2%; (2) the SDX controller sets an alert in the monitoring system to receive notification via SNMP traps or JSON messages; (3) whenever the SDX controller receives and alert, it will reconfigure the network fabric.

4.3 Bandwidth Calendaring

As proposed in Kempf et al (2014), bandwidth calendaring will allow the SDX to reserve bandwidth for particular times. This is particularly relevant for the LSST because images are going to be sent each night. However, the circuits used could be in different time zones, making the reservation a more interesting problem. A possible representation of the policy is:

```
scheduled_time = 21:00:00 GMT -5
if (current_time == scheduled_time) {
    BW = 90 // Bandwidth in Mbps
    t = 60 // Reservation time
    OnDemandVC(BW, t)
}
```

Once again, Pyretic Dynamic Policies, ONOS Intents, or Kinetic style state machines are the candidates for implementation.

5. Security Concerns for SDX

Whenever new components are introduced in a network architecture, we also introduce new vulnerabilities; SDX is no exception. Considering the three types of SDX, we could say that the Layer-3 SDX will inherit all BGP vulnerabilities, the Layer-2 SDX will carry the same vulnerabilities of a shared Ethernet domain, and finally the SDN SDX will also introduce controller vulnerabilities. Such threats include DDoS attacks, attack inflation, exploitation of logically centralized controllers, compromised controllers (affecting the entire network), malicious controller applications, and negative impacts on recovery speeds (Kreutz et al 2015). Moreover, SDX introduces its own vulnerabilities as the SDX controller is a middle man that every participant has to trust, and there is a possibility that some participants will declare policies that interfere with the proper function of other participants. As a result, a trust relationship must be established between the applications loaded on the controller and the devices the controller manages (Shin et al 2014), (SDX Central 2015).

The security issues with BGP are: prefix hijacking, TCP specific attacks, and manipulation of BGP attributes. Prefix hijacking occurs when an AS mistakenly or maliciously announces a prefix that has not been assigned to it. Some common TCP attacks are eavesdrop, man-in-the-middle, and DDoS (which can cause route flapping). Controllers are even more susceptible to TCP-based attacks since few controllers actually use secure TCP connections (Kreutz et al 2015). Surprisingly, we observe that this issue occurs in spite of the OpenFlow protocol (McKeown et al 2008) allowing for an SSL secure channel between controller and switch. Already, several solutions (i.e., Resource Public Key Infrastructure or RPKI (Bailey et al 2014) and Secure BGP or S-BGP (Boldyreva et al 2012) have been proposed to make BGP more secure and eliminate prefix hijacking. In consideration of these security requirements, Bailey et al. (2014) combined RPKI and CARDIGAN to enforce the consistency of BGP announcements with its forwarding rules. Subsequently, mechanisms must also be developed to establish trust between controllers in order to ensure proper forwarding or detect malicious elements before a misconfiguration can occur and damage the network (FlowSpace Firewall 2015). Equally important is the need for fast recovery after a link failure to mitigate packet loss and time sensitive science data flows, such as the 17 second intervals required for the LSST telescope. This requires that mechanisms be incorporated throughout the network to notify the SDX controller of failures, so it can flush its flow entries and select new routes (Sharma et al 2011).

Concerning Layer-2 SDXs, LAN switches must be securely configured since switches in a shared Ethernet network are more vulnerable to malicious packets. A few examples of layer-2 attacks include MAC flooding, VLAN hopping, man-in-the-middle (via MAC address spoofing), and hijacking (Altunbasak et al 2005). Unfortunately, with SDN, detecting and mitigating these attacks now becomes the responsibility of the network controller. While we are working on methods for detecting rogue DHCP servers and spoofed MAC addresses within the SDN framework, such methods require additional compute resources from SDN controllers and may raise scalability concerns (Giotis et al 2014).

Finally, from the Policy perspective, we would like for the policies of each SDX participant to only affect its own policy space. As a consequence, strong isolation is one of the main security requirements. Furthermore, each SDX controller becomes the middle man that every participant has to trust. Thus, the controller functionality is a potential point of failure. For these reasons, controller resiliency and policy verification are desirable. Other countermeasures should include access control, attack detection, event filtering, firewall and IDPS, flow aggregation, forensics support, packet dropping, rate limiting, and shorter timeouts (Shin et al 2014; SDX Central 2015). Regrettably, most of these countermeasures are not yet fully supported and work is ongoing to implement them (Kreutz et al 2015; SDX Central 2015).

6. Conclusions and Next Steps

While an exact definition for a Software Defined Exchange (SDX) has yet to reach a consensus, astronomy projects, such as LSST and SKA with data-intensive high-throughput network requirements, present important use cases for furthering the development of SDX. In this paper, we discussed the AtlanticWave-SDX project's goals, design, policy API, and security concerns. Once complete, the AtlanticWave-SDX will provide for an international long-haul network interconnecting Chile to the U.S., and potentially Africa, in the future. Additionally, with network programmability, applications of astronomical instruments will be able to provision multiple paths dynamically and on demand, apply QoS, prioritize policies, and manipulate flows at multiple levels. Furthermore, by using information made available by all network devices along the path, these applications will be empowered to choose preferred paths from multiple transit options between the northern and southern hemispheres.

Acknowledgment

The authors will like to thank the National Science Foundation for their support of this work directly through the AtlanticWave-SDX project (award #ACI-1451024). The authors would like to thank the *Georgia* Institute of Technology, the Academic Network of São Paulo (ANSP), Rede Nacional de Ensino e Pesquisa (RNP), Florida LambdaRail, and Internet2, for their support.

References

P. Lin, J. Hart, U. Krishnaswamy, T. Murakami, M. Kobayashi, A. Al-Shabibi, K.-C. Wang, and J. Bi, 'Seamless interworking of SDN and IP,' *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM - SIGCOMM '13*, 2013, no. Figure 2, p. 475.

J. P. Stringer, C. Lorier, and N. Zealand, 'Cardigan : Deploying a Distributed Routing Fabric,' *Proc. Second ACM SIGCOMM Work. Hot Top. Softw. Defin. Netw. - HotSDN '13*, pp. 169–170, 2013.

J. Stringer, D. Pemberton, Q. Fu, C. Lorier, R. Nelson, J. Bailey, C. N. A. Corrêa, and C. E. Rothenberg, 'Cardigan: SDN Distributed Routing Fabric Going Live at an Internet Exchange,' *Computers and Communication (ISCC), 2014 IEEE Symposium on, 2014*, pp. 1–7.

A. Gupta, E. Katz-Bassett, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, and R. Clark, 'SDX,' *ACM SIGCOMM Comput. Commun. Rev.,* vol. 44, no. 4, pp. 551–562, Aug. 2014.

J. Reich, C. Monsanto, N. Foster, J. Rexford, and D. Walker, 'Modular SDN Programming with Pyretic,' *USENIX, vol. 38*, pp. 40–47, 2013.

'Ryu Based SDX Controller.' [Online]. Available: https://github.com/sdn-ixp/sdx-ryu. [Accessed: 09-AUG-2015]

'Advanced Layer 2 System.' [Online]. Available: http://www.internet2.edu/products-services/advanced-networking/layer-2-services/. [Accessed: 06-AUG-2015]

'On-demand Secure Circuits and Advance Reservation System.' [Online]. Available: http://www.es.net/engineering-services/oscars/. [Accessed: 06-AUG-2015]

P. Lin, J. Bi, S. Wolff, Y. Wang, A. Xu, Z. Chen, H. Hu, and Y. Lin, ‘A West-East Bridge Based SDN Inter-Domain Testbed,’ *IEEE Commun. Mag., vol. 53, no. February*, pp. 190 – 197, 2015.

J. Mambretti, J. Chen, and F. Yeh, ‘Software-Defined Network Exchanges (SDXs) and Infrastructure (SDI): Emerging Innovations In SDN and SDI Interdomain Multi-Layer Services and Capabilities,’ *Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), 2014 First International*, 2014, pp. 1–6.

J. Mambretti, J. Chen, and F. Yeh, ‘Software-Defined Network Exchanges (SDXs): Architecture, Services, Capabilities, and Foundation Technologies,’ *Proceedings of the 2014 26th International Teletraffic Congress (ITC)*, 2014, pp. 0–5.

J. Kempf, M. Körling, S. Baucke, I. Más, and O. Bäckman, ‘Fostering Rapid, Cross-domain Service Innovation in Operator Networks through Service Provider SDN,’ *IEEE International Conference on Communications, 2014*, pp. 3070–3075.

“The Third Network: Lifecycle Service Orchestration Vision”. [Online]. Available: https://www.mef.net/Assets/White_Papers/MEF_Third_Network_LSO_Vision_FINAL.pdf. [Accessed: 09-AUG-2015]

R. Sherwood, G.Gibb, K. K. Yap, G .Appenzeller, M. Casado, N. McKeown, and G. Parulkar, ‘Flowvisor: A network virtualization layer.’ *OpenFlow Switch Consortium, Tech. Rep.* 2009

FlowSpace Firewall. [Online]. Available: http://globalnoc.iu.edu/sdn/fsfw.html. [Accessed: 13-AUG-2015]

M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, I. Seskar, ‘GENI: A federated testbed for innovative network experiments,*’ Computer Networks, vol. 61,* pp. 5-23, ISSN 1389-1286, 2014.

‘perfSONAR.’ [Online]. Available: http://www.perfsonar.net/. [Accessed: 05-AUG-2015]

ONOS Wiki, ‘Intent Framework.’ [Online]. Available: https://wiki.onosproject.org/display/ONOS/Intent+Framework. [Accessed: 12-AUG-2015]

Coursera SDN Course, ‘SDX Assignment.’ [Online]. Available: https://docs.google.com/document/d/1wLF3RZEwMCRioyvaVl73kjXeNgqIaA3LfUlGsI2mkb4/edit?usp=sharing. [Accessed: 10-AUG-2015].

H. Kim, J. Reich, A. Gupta, M. Shahbaz, N. Feamster, and R. Clark, ‘Kinetic: Verifiable Dynamic Network Control,’ pp. 1–11.

D. Kreutz, F. M. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, ‘Software-defined networking: A comprehensive survey,’ *Proceedings of the IEEE, vol. 103*, no. 1, pp. 14–76, 2015

S. Shin, Y. Song, T. Lee, S. Lee, J. Chung, P. Porras, V. Yegneswaran, J. Noh, and B. B. Kang, ‘Rosemary: A robust, secure, and high performance network operating system,’ *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014*, pp. 78–89.

SDX Central, 'SDN Security Challenges in SDN Environments.' [Online]. Available: https://www.sdxcentral.com/resources/security/security-challenges-sdn-software-defined-networks/. [Accessed: 10-AUG-2015].

N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, 'OpenFow: Enabling innovation in campus networks,' *SIGCOMM Comput. Commun. Rev., vol. 38*, no. 2, pp. 69-74, Mar. 2008.

J. Bailey, D. Pemberton, A. Linton, and C. Pelsser, 'Enforcing RPKI-Based Routing Policy on the Data Plane at an Internet Exchange,' *HotSDN 2014,* pp. 211–212, 2014.

A. Boldyreva and R. Lychev, 'Provable Security of S-BGP and other Path Vector Protocols: Model, Analysis and Extensions,' *ACM Conference on Computer and Communications Security 2012*, pages 541–552, 2012.

Sharma, Sachin, Dimitri Staessens, Didier Colle, Mario Pickavet, and Piet Demeester. 'Enabling fast failure recovery in OpenFlow networks.' *Design of Reliable Communication Networks (DRCN), 2011 8th International Workshop on the, pp. 164-171. IEEE*, 2011.

H. Altunbasak, S. Krasser, H. Owen, Grimminger, H. Huth, and J. Sokol. 'Securing Layer 2 in Local Area Networks. Networking - ICN 2005.' *P. Lorenz and P. Dini, Springer Berlin Heidelberg.* 3421: 699-706.

Giotis, K., Christos Argyropoulos, Georgios Androulidakis, Dimitrios Kalogeras, and Vasilis Maglaris. 'Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments.' *Computer Networks 62 (2014)*: 122-136.

Biographies

Julio E. Ibarra, PhD is the Assistant Vice President for Technology Augmented Research at FIU. He is responsible for furthering the mission of the Center for Internet Augmented Research and Assessment (CIARA) – to contribute to the pace and the quality of research at FIU through the application of advanced Cyberinfrastructure. He is responsible for strategic planning and development of advanced research networking services, including the development and management of the AMPATH International Exchange Point for Research and Education networks, in Miami, Florida. He holds B.S. and M.S. in Computer Science from FIU, and Ph.D. in Telematics and Information Technology from Twente University.

Jeronimo Bezerra is the Chief Network Engineer for CIARA at Florida International University. He has been involved with academic networks for the last 12 years. He is responsible for the operation of the AMPATH International Exchange Point in Miami, and the design and operation of the international network connections linking the research and education communities of the U.S., Brazil, Latin America and the Caribbean. He holds an MSc in Mechatronics and BS in Computer Science by the Federal University of Bahia/Brazil,

Prof. Luis Fernandez Lopez holds Ph.D. in Mathematical Physics. Currently, he is a professor at USP (Medicine School, University of São Paulo) and at FIU (Florida International University – Miami, USA). He is also the NARA Coordinator (Center for Advanced Networking Applications) of USP and Principal Investigator of the Project ANSP (Academic Network at São Paulo), funded by FAPESP (Foundation for Research Support of the State of São Paulo) and NSF (National Science Foundation). He has published over 50 academic papers with more than 400 citations. In August 2016 he was awarded one of the most prestigious awards "The Peacemaker Medal (Medalha do Pacificador)" of Brazil for his dedication and professional ability.

Heidi L. Morgan, PhD. is Senior Computer Scientist, Information Sciences Institute (ISI), University of Southern California (USC), and Research Scientist Associate, Center for Internet Augmented Research and Assessment (CIARA) at Florida International University. She is a Co-PI for several NSF funded projects including SwitchOn – Exploring and Strengthening US-Brazil Collaborations in Future Internet Research (switchon.ampath.net), Americas Lightpaths: Increasing the Rate of Discovery and Enhancing Education across the Americas (amlight.net) and the AMPATH International Exchange Point in Miami. Heidi enjoys working to advance research and education networking initiatives in the Caribbean, Mexico, Central and South America and collaborating with likeminded professionals in the US and around the world.

Donald "Chip" Cox, III, PhD is Chief of Operations for the AMPATH International Exchange Point. He is also Adjoint Professor of Physics, Department of Physics and Astronomy, Vanderbilt University and Fisk University. He is Co-PI for the NSF funded projects AmLight Express and Protect (AmLight-ExP) and OpenWave, which explores the operation of the integration of spectrum with existing 100G network services, multi-domain optical provisioning and network management, and network virtualization and SDN applications. Dr. Cox holds Bachelors in Engineering Sciences and a Masters in Business Administration from Vanderbilt University, and a Doctorate in Philosophy from the University of Western Australia.