Novel Network Services for Supporting Big Data Science Research

Joaquin Chung, Sean Donovan, Jeronimo Bezerra*, Heidi Morganˆ, Julio Ibarra*, Russ Clark, Henry Owen
Georgia Institute of Technology. USA
*Florida International University. USA
ˆUniversity of Southern California. USA

***Abstract*—To interconnect research facilities across wide geographic areas, network operators deploy science networks, also referred to as Research and Education (R&E) networks. These networks allow experimenters to establish dedicated network connections between research facilities for transferring large amounts of data. Recently, R&E networks have started using Software-Defined Networking (SDN) and Software-Defined Exchanges (SDX) for deploying these connections. AtlanticWave/SDX is a response to the growing demand to support end-to-end network services spanning multiple SDN domains. However, requesting these services is a challenging task for domain-expert scientists, because the interfaces of the R&E networks have been developed by network operators for network operators. In this paper, we propose interfaces that allow domain-expert scientists to reserve resources of the scientific network using abstractions that focus on their data transfer needs for scientific workflow management. Recent trends in the networking field pursue better interfaces for requesting network services (e.g., intent-based networking). Although intents are sufficient for the needs of network operations, they are not abstract enough in most cases to be used by domain-expert scientists. This is an issue we are addressing in the AtlanticWave/SDX design: network operators and domain-expert scientists will have their own interfaces focusing on their specific needs.**

I. Introduction

Modern scientific instruments (e.g., particle accelerators, large telescopes, and genome sequencers) generate enormous amounts of data. These large datasets are analyzed at supercomputing centers, typically hundreds of kilometers away from the original research facility. For instance, a large telescope located in the Andes mountains in Chile, taking multiple gigabyte images to be transferred to the United States so that processing can be completed in time to distribute transient alert notifications will use a dedicated network connection between the two facilities.

To interconnect research facilities with supercomputing centers across long distances, network operators deploy scientific networks or Research and Education (R&E) networks. These networks allow experimenters to establish dedicated circuits between research facilities by using advance reservation systems [1]. However, the interface for requesting these types of reservations is very complex for domain-expert scientists who are not network operators. Arguably, many of these interfaces were developed *by* network operators, *for* network operators. Additionally, as reservations are defined by duration and bandwidth, the scheduling of resources is not flexible; that is, a reservation request will fail if the exact amount of bandwidth is not available within the specified time frame, which forces the scientist into a cycle of trial and error until a suitable time frame is found. Furthermore, manual provisioning of these connections, which could take from several days to several weeks [2], is sometimes limited by configuration overhead, poor scalability, and poor testing interfaces [3].

To alleviate the shortcomings of current advanced reservation systems, we require an architecture for R&E networks that allows agile programmability of end-to-end network services over multiple domains, protects access to reservations from end-to-end, and provides flexible reservation capabilities. In this paper we propose interfaces that allow domain-expert scientists and data workflow management systems to reserve network resources through a multi-domain Software-Defined Exchange (SDX) [4] such as AtlanticWave/SDX[5].

This paper is organized as follows: Section II provides the background for the technologies that support our architecture. Section III describes the architecture of AtlanticWave/SDX, while Section IV describes use cases. Section V presents a prototype of AtlanticWave/SDX. Section VI provides the related work. Finally, Section VII concludes and presents our future work.

II. Background

In this section we present the technologies that support the AtlanticWave/SDX architecture.

A. Software-defined Networking (SDN)

Under the software-defined networking (SDN) paradigm [6], the control and data planes of network devices are decoupled. This separation enables global network programmability, rapid innovation, and independent evolution of control and data planes. The Open Networking Foundation (ONF) proposed an abstraction for SDN that divides the architecture into three layers: the infrastructure layer, representing the data plane; the control layer, representing the control plane; and the application layer, representing network applications (e.g., switching, routing, or load balancing). Additionally, the ONF also proposed interfaces that enable communication

Presented at Gateways 2017, University of Michigan, Ann Arbor, MI, October 23-25, 2017. https://gateways2017.figshare.com/

between layers. The interface that allows the control layer to communicate with the infrastructure layer is commonly referred to as southbound interface (SBI), and the interface that allows the application layer to communicate with the control layer is generally called northbound interface (NBI).

B. Software-defined Exchange (SDX)

A Software-Defined Exchange (SDX) [4] is a meet-me point or marketplace where independent administrative domains can exchange computing, storage, and networking resources. Currently, networking researchers incorporate SDN technologies into the networking infrastructure of Internet exchange points (IXPs) and R&E Network exchange points for agile programmability of these infrastructures [4], [5], [7], [8].

III. AtlanticWave/SDX Architecture

In this section we present the high-level architecture of AtlanticWave/SDX, and the science gateway interfaces that allow domain-expert scientists to request network resources. Our proposed architecture is composed of the following components (see Figure 1):

1) Local controllers and local switches that reside at SDX domains.
2) An SDX controller that interconnects participating sites.
3) Users (e.g., domain-expert scientists or network operators) and applications (e.g., data workflow management systems) that consume end-to-end services composed by an SDX controller.

A. SDX User Interface

Some users of AtlanticWave/SDX are domain-expert scientists whom in most of the cases do not have expertise in network operations, but still have to request reservations to expedite their data transfers. Additionally, scientists use data workflow management systems (e.g., Globus [9]) to automate the process of moving and sharing data across research facilities.

In our reference architecture, both scientists and applications request end-to-end science network services from the SDX controller by using the *Network Service Science Gateway* that interfaces with and abstracts network infrastructure details. The SDX user interface allows a scientist or a data workflow management system to request services from AtlanticWave/SDX. To overcome rigid interfaces that only allow users to request a certain amount of bandwidth during a limited amount of time, we propose to describe a request based on data set size and a deadline for finishing the transfer. Details of the network parameters should be left for the SDX controller to decide. A domain-expert scientist, for example an astrophysicist, only wants to transfer *X* amount of gigabytes before a certain deadline. The *SDX user interface* may include negotiation capabilities that allow the SDX controller to provide an optimal solution to a user request, given the user constraints and the network state. Table I illustrates both a reservation request for a network operator of the AtlanticWave/SDX, and a request for a domain-expert scientists request in JSON format, and how the *SDX user interface* abstracts the complexities of the network for the experimenters.

B. Authentication and Authorization

Authentication and authorization are important topics in the AtlanticWave/SDX architecture. The SDX controller authenticates participant entities and users or applications coming from several independent administrative domains. Then, the SDX controller requires authorization before it issues any provisioning requests to participant domains. Digital certificates may be used for mutual authentication between the SDX controller and local controllers, and a federated identity management system (e.g., Shibboleth [10]) may be used for the SDX controller to authenticate users and applications. Another approach may be to integrate AtlanticWave/SDX with existing systems such as Globus. Under these models, users are not required to have an account at each location, but they are authenticated using their institutional credentials. We propose to enforce authorization by installing policies on each entity (i.e., participant R&E networks) that define what users are allowed to do, depending on their roles or affiliation (e.g., institution, project, or individual).

IV. Use Cases

A. Simplifying Current Science Network Services

Astronomers use instruments that generate sets of data on the scale of gigabytes, and they need to transfer these sets of data in a few seconds to be processed in a remote facility. To guarantee a high-throughput network connection between facilities, an experimenter may reserve network resources through AtlanticWave/SDX. After verifying the domains involved in this end-to-end network service, the SDX controller (see Fig. 1) contacts each local controller, querying whether a path with the constraints specified by the experimenter is possible. Each queried domain does not have global knowledge about the end-to-end path, but they can commit to guarantees of their portion of the end-to-end path. The SDX controller then evaluates whether a path that meets the end-to-end requirements can be formed. Otherwise, the SDX controller will try to find an alternative path, or will try to negotiate a path with alternative constraints.

B. Future Generation Science Network Services

The proposed architecture will allow science networks to provide more flexible services. For instance, let us consider an experimenter who wants to move telescope data every morning at 6:00 AM. Instead of reserving a dedicated connection for the experiment (that could last years), the scientific network may expose a bandwidth calendaring service in which the experimenter selects at what hours of the day she will need the reservation. We could take this use case one step further by correlating weather data with previous data transfer patterns, and suggest to the experimenter the optimal time frames for upcoming data transfers. For example, on a cloudy day, AtlanticWave/SDX will avoid provisioning a network reservation, as the telescope's view was obscured by clouds.

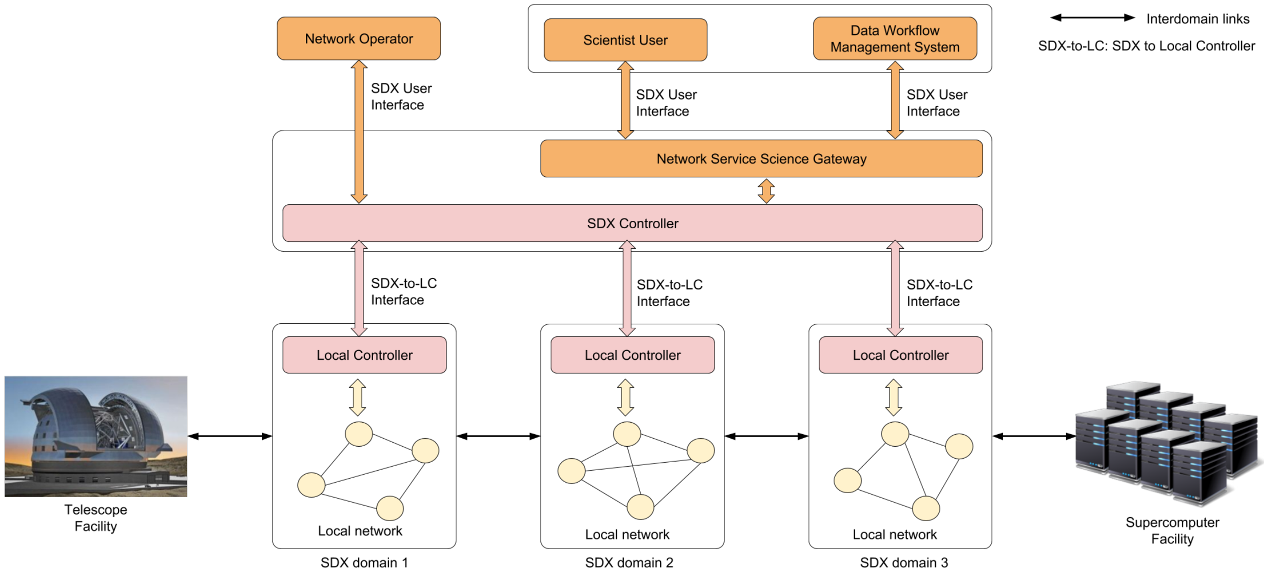

Figure 1. High-level architecture for AtlanticWave/SDX, with local controllers at three domains, and an SDX controller exposing services to users (e.g., domain-expert scientists, network operators, and data workflow management systems).

Table I
SDX USER REQUEST SAMPLES IN JSON FORMAT

Network Operator Request	**Domain-expert Scientist Request**

`{"l2tunnel":{`
`"starttime":"2017-10-23T23:20:50",`
`"endtime":"2017-10-25T23:20:50",`
`"srcswitch":"atl-switch",`
`"dstswitch":"mia-switch",`
`"srcport":5,`
`"dstport":7,`
`"srcvlan":1492,`
`"dstvlan":1789,`
`"bandwidth":1000}}`
	`{"dtntunnel":{`
	`"quantity":"7TB",`
	`"deadline":"2017-10-25T23:20:50",`
	`"srcdtn":"gt-dtn",`
	`"dstdtn":"fiu-dtn"}}`

Another example is to use machine learning (ML) on data transfer patterns and historical reservation data to create a predictive reservation service. In this case, the SDX will suggest a predefined reservation to an experimenter based on previous reservation and real usage patterns. The experimenter will be able to confirm or decline the reservation. A more aggressive approach is to provision the reservation with a lower priority, and wait for traffic to be sensed on the network. If no traffic is sent before a preset threshold, the reservation is eliminated.

In another scenario, a dataset is hosted in several locations. An experimenter only needs to know the name of the needed dataset. The *SDX user interface* of AtlanticWave/SDX will locate the closest repository and request a dedicated network connection between the repository facility and the experimenter's facility. Moreover, if the closest facility to the experimenter is congested, AtlanticWave/SDX may negotiate a more distant facility with better network conditions to provide the data set. We could take this example one step further and expose an energy efficiency score, and the SDX controller may be able to compose end-to-end green paths as a novel scientific network service.

V. ATLANTICWAVE/SDX PROTOTYPE

In this section we present the prototype of the AtlanticWave/SDX controller [11], which is written in Python, using the Ryu SDN Framework [12] as an OpenFlow [13] speaker, and has a REST API and web application for management. The controller is divided into three components, that will run at the initial three locations of the prototype (i.e., Atlanta, Miami, and Sao Paulo). The main three components of the AtlanticWave/SDX controller are the following:

- **Participant Interface(s):** The participant interfaces are where network operators and scientists install rules that dictate how network flows behave. This interface correspond to the *network science service gateway* described in Section III. For the initial prototype, we provide a Web portal that allows users to create and install rules (see Figure 2).

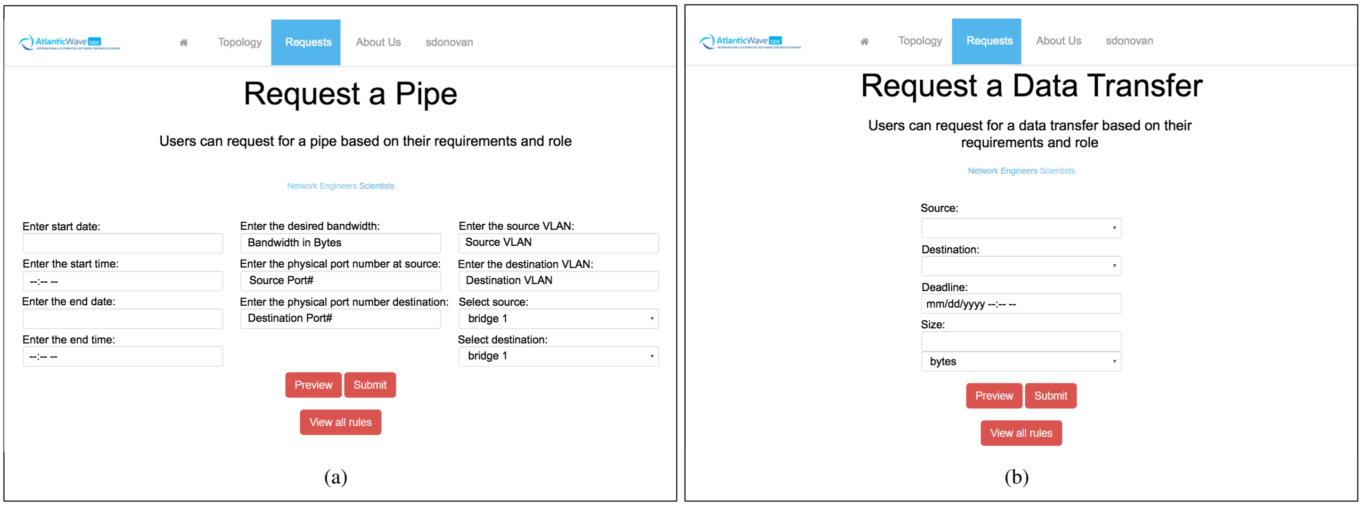

Figure 2. AtlanticWave/SDX Web portal for requesting science network services: (a) network operator view, and (b) domain-expert scientist view.

- **SDX Controller:** The SDX controller is responsible for authentication and authorization of participants and local controllers, taking rules from the participant interfaces and breaking them down into per-location rules for the local controllers, handling federation challenges from many participants installing rules on a shared network, and providing an interface for the participants.
- **Local Controllers:** Each location in the AtlanticWave/SDX will have a local controller that controls the local switch(es). The local controller has one main job: take the abstract rules from the SDX controller, and translate them to a switch friendly protocol (e.g., OpenFlow). The local controller also bootstraps the configuration of the switch to establish connectivity between the local controller and the SDX controller.

We implemented Shibboleth from Internet2 for handling authentication at the front end of the AtlanticWave/SDX controller. It is important to note that Shibboleth is not used outside of North America. As a result, we will test other identity and access management (IAM) systems. For example, eduGAIN [14], developed by GEANT, is the standard cross-domain IAM system used outside of North America.

VI. Related Work

Researchers have already proposed the use of SDN for enhancing scientific application resource management and performance over a WAN connection. For instance, the Lark project [15] proposed a flexible and fine-grained mechanism to manage network resources in high-throughput computing (HTC) systems [16]. Similarly, the DANCES project uses SDN to enhance scientific application resource management and performance of a WAN connection by developing applications with networking capabilities via end-to-end SDN [17]. However, these solutions do not provide interfaces that allow domain-expert scientists to request scientific data transfer services abstracting network details. While the Lark project built an SDN controller that allows scheduling of high-throughput computing (HTC) jobs in a HTCondor system [18], the DANCES project uses the same abstraction of endpoints, start time, end time, and requested bandwidth that current science network reservation systems use. Our architecture proposes to provide abstractions that enable domain-expert scientists to request end-to-end services on scientific networks while hiding the details of the network. Furthermore, Lark and DANCES focus on single domain SDN, while our solution focuses on multi-domain SDN.

Inside the SDN community, members are aligned with the development of intent-based networking interfaces [19], [20] that use a prescriptive rather than descriptive approach to network configuration; that is, network operators and applications describe a goal, and the SDN controller decides how to implement it. Within the research community, Kiran et al. [21] proposed the iNDIRA (Intelligent Network Deployment Intent Renderer Application) tool [21], which uses natural language processing to capture the network service requirements of the user. iNDIRA has been deployed on the Energy Science Network (ESNet), where it interacts with Globus data transfer tools. Similarly, the *SDX user interface* of AtlanticWave/SDX seeks to implement intents by describing the network services in a high level language, and enabling negotiation between the SDX and local controllers.

VII. Conclusions and Future Work

In this paper we presented an architecture for end-to-end service orchestration in multi-domain scientific networks, that leverages SDX, and provides interfaces that allow domain-expert scientists and data workflow management systems to reserve resources of the scientific network. The SDX controller exposes network services that domain-expert scientists and data workflow management systems can easily consume through the *user/application interface*. We provided use cases that illustrate how domain-expert scientists can use the interfaces of AtlanticWave/SDX for easily requesting end-to-end network services. Furthermore, we proposed future generation science network services such as bandwidth calendaring, predictive reservation services, and green path reservation. We also described the initial prototype of the AtlanticWave/SDX controller, with special emphasis on the user/application interface, and the authentication mechanism of the AtlanticWave/SDX Web portal. For our future work we propose to study the composition of end-to-end services at the SDX controller and how these services should be exposed to users and applications through the *user/application interface*. Another area of research is the rule translation from user-level, local controller-level, and switch-level abstractions. We will continue integrating and testing IAMs for the AtlanticWave/SDX prototype. Finally, we have initiated the deployment of the AtlanticWave/SDX infrastructure in the three initial locations (i.e., Atlanta, Miami, and Sao Paulo), and we will study how domain-expert scientists interact with the AtlanticWave/SDX controller for requesting novel science network services.

Acknowledgments

This work is supported by the NSF award 1451024. The authors would like to thank John Skandalakis and Ankita Lamba for their work on the AtlanticWave/SDX prototype. The authors will also like to thank Raj Kettimuthu for his comments.

References

[1] N. Charbonneau, V. M. Vokkarane, C. Guok, and I. Monga, "Advance reservation frameworks in hybrid IP-WDM networks," *Communications Magazine, IEEE*, vol. 49, no. 5, pp. 132–139, 2011.

[2] J. Ibarra, J. Bezerra, H. Morgan, L. Fernandez Lopez, M. Stanton, I. Machado, E. Grizendi, and D. Cox, "Benefits brought by the use of OpenFlow/SDN on the AmLight intercontinental research and education network," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, May 2015, pp. 942–947.

[3] S. Tepsuporn, F. Al-Ali, M. Veeraraghavan, X. Ji, B. Cashman, A. J. Ragusa, L. Fowler, C. Guok, T. Lehman, and X. Yang, "A multi-domain sdn for dynamic layer-2 path service," in *Proceedings of the Fifth International Workshop on Network-Aware Data Management*, ser. NDM '15. New York, NY, USA: ACM, 2015, pp. 2:1–2:8. [Online]. Available: http://doi.acm.org/10.1145/2832099.2832101

[4] J. Chung, R. Clark, and H. Owen, "SDX architectures: A qualitative analysis," in *IEEE SoutheastCon 2016*. IEEE, 2016, pp. 1–8.

[5] J. Chung, J. Cox, J. Ibarra, J. Bezerra, H. Morgan, R. Clark, and H. Owen, "AtlanticWave-SDX: An international sdx to support science data applications," *Software Defined Networking (SDN) for Scientific Networking Workshop, SC'15*, pp. 1–7, Nov 2015.

[6] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[7] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "SDX: A software defined internet exchange," in *Proceedings of the 2014 ACM conference on SIGCOMM*. ACM, 2014, pp. 551–562.

[8] J. Mambretti, J. Chen, and F. Yeh, "Software-defined network exchanges (SDXs): Architecture, services, capabilities, and foundation technologies," in *26th International Teletraffic Congress*. IEEE, 2014, pp. 1–6.

[9] "Globus - research data management system," https://www.globus.org/, accessed: 2017-01-20.

[10] "Shibboleth," https://shibboleth.net/, accessed: 2017-01-22.

[11] S. Donovan, J. Skandalakis, and A. Lamba, "Atlanticwave/sdx controller prototype," https://github.com/atlanticwave-sdx/atlanticwave-proto, accessed: 2017-08-11.

[12] "Framework community, Ryu SDN Framework," http://osrg.github.io/ryu, 2015.

[13] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355746

[14] "eduGAIN," https://www.geant.org/Services/Trust_identity_and_security/eduGAIN.

[15] Z. Zhang, B. Bockelman, D. W. Carder, and T. Tannenbaum, "Lark: Bringing network awareness to high throughput computing," in *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*, May 2015, pp. 382–391.

[16] J. Basney, M. Livny, and T. Tannenbaum, "High throughput computing with Condor," *HPCU news*, vol. 1, no. 2, p. 1, 1997.

[17] V. Hazlewood, K. Benninger, G. Peterson, J. Charcalla, B. Sparks, J. Hanley, A. Adams, B. Learn, R. Budden, D. Simmel, J. Lappa, and J. Yanovich, "Developing applications with networking capabilities via end-to-end SDN (DANCES)," *XSEDE16*, pp. 1–7, July 2016.

[18] "What is htcondor?" https://research.cs.wisc.edu/htcondor/description.html, accessed: 2017-01-19.

[19] W. W. Cao, T. Zhou, S. Jiang, Y. Xiao, and S. Hares, "NeMo an application's interface to intent based networks," http://www.nemo-project.net/.

[20] D. Lenrow, "Intent-based networking seeks network effect," https://www.sdxcentral.com/articles/contributed/intent-based-networking-seeks-network-effect-david-lenrow/2015/09/.

[21] M. Kiran, E. Pouyoul, A. Mercian, B. Tierney, C. Guok, and I. Monga, "Enabling intent to configure scientific networks for high performance demands," *Future Generation Computer Systems*, pp. –, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X1730626X