Software Defined Networking for big-data science

Architectural models from campus to the WAN

Inder Monga, Eric Pouyoul, Chin Guok
Energy Sciences Network
Lawrence Berkeley Lab
Berkeley, California
{inder, lomax, chin}@es.net

Abstract— **University campuses, Supercomputer centers and R&E networks are challenged to architect, build and support IT infrastructure to deal effectively with the data deluge facing most science disciplines. Hybrid network architecture, multi-domain bandwidth reservations, performance monitoring and GLIF Open Lightpath Exchanges (GOLE) are examples of network architectures that have been proposed, championed and implemented successfully to meet the needs of science. Most recently, Science DMZ, a campus design pattern that bypasses traditional performance hotspots in typical campus network implementation, has been gaining momentum. In this paper and corresponding demonstration, we build upon the SC11 SCinet Research Sandbox demonstrator with Software-Defined networking to explore new architectural approaches. A virtual switch network abstraction is explored, that when combined with software-defined networking concepts provides the science users a simple, adaptable network framework to meet their upcoming application requirements.**

Keywords-Network Virtualization; OpenFlow; Software-Defined Networking; Multi-Domain; OSCARS; Science DMZ

I. INTRODUCTION

Science research has been increasingly data-driven as well as conducted in large collaborative partnerships involving researchers, instruments and high performance computing centers. Large-scale instruments like Large Hadron Collider or Square Kilometer Array [1] and simulations produce petabytes of data that is likely to be shared and analyzed by tens to thousands of scientists. Due to a variety of reasons, including political and funding constraints, data is typically not processed, analyzed and visualized at the location it is produced, but moved to more convenient regional locations for the geographically distributed researchers. The larger the science collaboration, the higher the dependence on a functioning distributed architecture. Efficient tools and high-performance network architectures that seamlessly move data between locations have quickly become the backbone of state-of-the-art science collaborations. For example, ESnet alone has observed approximately 60%+ annual growth in scientific data traffic for over 20 years.

The LHC ATLAS and CMS experiments are examples of large science collaborations that rely heavily on networks to distribute petabytes of data across the globe every year [2]. Similar paradigms are now rapidly emerging across many scientific disciplines from genomics to climate research to material sciences. For many of these disciplines, new experimental facilities are coming online like the Belle 2 High Energy Physics experiment in Japan that is expecting to collect at least 250 Petabytes (PB) in its first five years of operation [3]. In addition, existing facilities like X-ray synchrotrons are being upgraded with new detectors that are collecting data at unprecedented resolution and refresh rates. The current detectors can now produce raw data of petabytes per second or more, and the next generation is expected to produce data volumes many times higher. All told, the data intensity of many disciplines is projected to increase by a factor of ten or more over the next few years [2] [4].

As the amount of traffic increases, there is a greater need for simple, scalable end-to-end network architectures and implementations that enable applications to use the network most efficiently. The Research and Education network community has successfully abstracted current networking technologies, like MPLS and SONET, to develop a multi-domain, automated, guaranteed bandwidth service. This capability has enabled scientists to build guaranteed bandwidth virtual circuits from their campus to a destination end point, which could be another campus or a supercomputing data center across the country or continents. Multi-domain performance testing, championed and developed by the same community, ensures that any end-to-end network glitches like packet-loss are quickly identified and resolved.

As this architecture has been widely adopted, new challenges have emerged. First, the bottleneck for scientific productivity has shifted from the WAN to the campus and data center networks. Enterprise network architectures supporting a wide-variety of organizational missions has not been architected to support automated provisioning tools such as dynamic provisioning of guaranteed virtual circuits, end-to-end. In most cases, campus networks optimized for business operations are neither designed for nor capable of supporting the data movement requirements of science. Second, even though a lot

of manual interaction has been automated, important actions still need to be manually configured for end-to-end connectivity. Network topology (including service end-points) and VLAN translation at the borders between WAN and LAN are extremely important but maintained manually. Third, supporting large science collaborations with point-to-point circuits still requires active management or creation of long-term circuits (equivalent to a statically configured circuit). Fourth, dynamic exchange of policy and authorization between the source campus and destination campus is critical for any automated system to work. Before a wide area connection is setup or used, both campuses need to agree to authorize and allocate the local resources for data movement.

Software-defined networking, using the OpenFlow protocol, provides interesting and novel capabilities that can be leveraged, not only to solve the new challenges described above, but also to potentially simplify the implementation of existing solutions. This paper will examine the various architectural models of leveraging OpenFlow based switches and the software-defined networking model within the science-networking context and will describe a couple of demonstration scenarios that will be implemented for SC12 SCinet Research Sandbox.

II. BACKGROUND

A. Software-defined networking (SDN) and OpenFlow (OF)

The flow of data packets that constitute communication between any two devices connected over the network has been largely controlled by standard protocols defined for the different abstraction layers. These abstraction layers and their standard interfaces has been the root cause of highly successful scaling of the Internet. With this model, whenever a data packet arrives at a network device, the packets are switched based on standard forwarding rules which neither the network operator or application have full control over. Even though this standard packet or flow routing model has scaled for the general purpose Internet, some applications and network operators would like better and more granular control over the treatment of packets on a per application flow. This paradigm of providing an API for application or network management programs to programmatically control the hardware forwarding of packet flows is known as "Software-Defined Networking". OpenFlow is one of the control protocols specified by the Open Networking Foundation (ONF) [1] that enables the network hardware to provide such an API to the application programs and specifies a framework through which centralized control of flow forwarding rules can be orchestrated. Even though the traffic isolation with dynamic, automated, virtual circuits has been standardized and is being actively deployed by wide-area R&E network providers, the end-to-end path for a high-performance data flow typically traverses portions of the campus and data-center infrastructure that do not have any automated control to isolate the large data flows. With the adoption of Software-defined Networking paradigm and OpenFlow, the campus and data-center operators can provide a programmatic interface that can be leveraged to build an end-to-end network service with the same traffic isolation characteristics that is needed to meet the requirements of the big-data movers.

Even though Software-Defined Networking and OpenFlow are different, they are sometimes used interchangeably.

.

B. ScienceDMZ[2]

A laboratory or university campus network typically supports multiple organizational missions. First, it must provide infrastructure for network traffic associated with the organization's normal business operations including email, procurement systems, and web browsing, among others. The network must also be built with security features that protect the financial and personnel data of the organization. At the same time, these networks are also supposed to be used as the foundation for the scientific research process as scientists depend on this infrastructure to share, store, and analyze research data from many different external sources.

In most cases, however, networks optimized for these business and security operations are neither designed for nor capable of supporting the requirements of data intensive science. Multi-gigabit data flows over paths that have 100-200 millisecond RTT, as required by most international science collaborations, cannot be supported by typical LAN equipment and are explicitly impeded by stateful firewalls at domain boundaries. When scientists attempt to run data intensive applications over these so-called "general purpose" networks, the result is often poor performance - in many cases so poor that the science mission is significantly impeded.

By defining a separate part of the network that is designed to support data-intensive science flow, the Science DMZ model provides a framework for building a scalable, extensible network infrastructure that aims to eliminate the packet loss that causes poor TCP performance, to implement appropriate use policies so that high-performance applications are not hampered by unnecessary constraints, to create an effective on-ramp for local science resources to access wide area network services and to provide mechanisms for ensuring consistent performance.

C. XSP: eXtensible Service Protocol

The eXtensible Session Protocol (XSP) [5] has been designed as a flexible protocol architecture for managing the interactions of applications and network-based services, and among the devices that provide those services. Residing in layer-5 of the OSI network model, the notion of a session provides a natural scope for state, including establishment and lifetime of connections, forwarding rules, or any other network

[1] https://www.opennetworking.org/

[2] http://fasterdata.es.net/science-dmz/

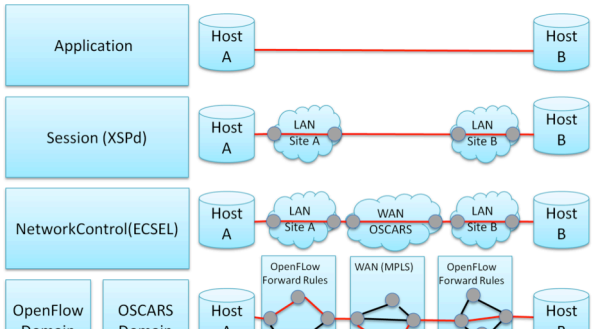
configuration related to the activity defined by the given application. Applications can refer back to that state, make changes, and signal the network to suit their particular requirements.

In order to implement XSP as part of a high-performance, end-to-end, data transfer mechanism, we leveraged OpenFlow to implement a end-site broker and co-ordination software ECSEL, described in the next section below. The XSP libraries and API provide a standard interface for applications to specify parameters that define network paths. The realization of these paths is then managed by our XSP daemon (XSPd) that signals the underlying provisioning service while providing feedback to the application. A transparent XSP wrapper, or ``shim'', library gives existing applications the ability to signal XSPd without source code modifications. Since XSP is implemented at the session layer, it has the flexibility to support multiple transport protocols – TCP, parallel TCP (as implemented in GridFTP), or RDMA. These protocols can be chose dynamically depending on the data transfer purpose, capability and the capabilities of the underlying network.

III. Previous Work: End-to-End Service at Layer2 (ECSEL)

Deploying loss-sensitive, high-performance applications across the WAN poses the challenge of providing an end-to-end layer-2 circuit with no loss, guaranteed bandwidth, and stable latency. OSCARS can be used to provide layer-2 connectivity across the WAN. However, due to administrative and technical constraints, OSCARS cannot control the path from the site border to the endpoints, unless implemented explicitly within the campus itself as a separate domain. The data-transfer hosts or DTNs[3] have no knowledge of the LAN topology they are connected to, and the LAN needs to be configured explicitly to transport the Ethernet frames from the end host to the ingress point of the OSCARS circuit. Also, allowing or denying traffic across the WAN needs to be controlled by the local campus policy, which based on authentication and authorization decides what resources is assigned to users, projects, or class of applications. Before admitting a request, both sites must agree to use common WAN resources, the guaranteed bandwidth virtual circuit created by OSCARS. Campuses with OpenFlow capability can run an isolated domain, using the local policy to allow or deny which wide-area traffic may be forwarded.

To implement the concept above, we leverage the application layer, so that an end-to-end network middleware can broker and manage WAN OSCARS circuits as well as control the local OpenFlow domain. The application understands a very simple topology, itself and the remote host, typically, in the form of a DNS name. The middleware, manages a more complex topology, reflecting all the administrative domains involved in the path between the endpoints. The network controller, ECSEL, which manages both the LAN topology and the site-specific WAN resources, provides this topology.



A. ECSEL, a Site-to-Site IDC

ECSEL (End-to-End Circuit Service at Layer 2), is our implementation of an Inter-Domain Controller (IDC)[4] that negotiates local and remote network resources while keeping intact the administrative boundaries; each site maintain full control on local resources and how they are utilized. There are two categories of local resources: the LAN and WAN OSCARS circuit connection to the remote site. OSCARS circuits, registered with ECSEL, are associated with metadata describing the authorized usage of each of the circuits. For example, circuits can be limited to certain users or projects. The OSCARS circuits are reservations referring to either already provisioned circuits or advanced reservations. The LAN resources are managed by an OpenFlow controller that discovers the local network topology, computes the proper path between the end host and the proper OSCARS circuit, and applies the forwarding rules, thus establishing the layer-2 circuit. The ECSEL OpenFlow controller leverages the OpenFlow discovery protocol to not only learn the topology of switches and routers of the LAN, but also the endpoint itself. The end host listens to Link Layer Discovery Protocol (LLDP) packets broadcast by the OpenFlow switches to discover how it is contacted to the network, and, in turn, registers itself to ECSEL. (LLDP is used by network devices to advertise their identity, capabilities, and neighbors on an Ethernet LAN). Figure 2 shows the various ECSEL components.

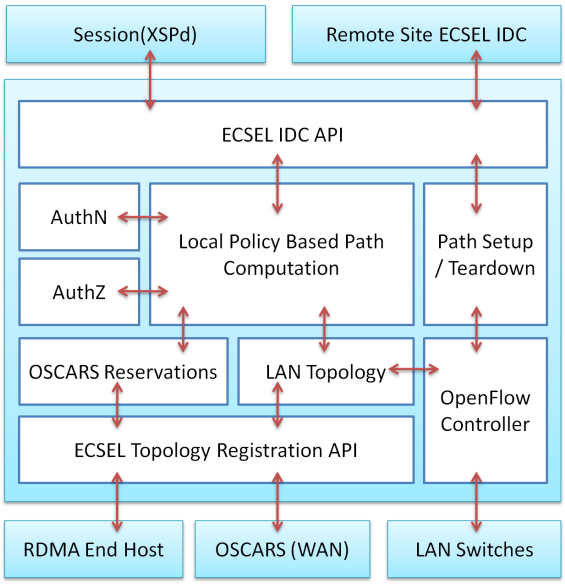


Figure 2 ECSEL Architecture

[3]

[4] IDC, http://www.controlplane.net

B. ECSEL Workflow

When the session layer, via XSPd, requests a layer-2 path between two end hosts that support RDMA over Ethernet, it supplies a high-level topology that is based on the identity of the end hosts as well as their domains, along with requested bandwidth, start time, and duration. Using X509-based authentication and authorization representing the requesting application or project or end-user, the local policy will select an available WAN OSCARS circuit and then, using the IDC protocol, contact its peer on the remote domain. The remote domain ECSEL will then verify if the request is compliant with the local policy, and if so, the OSCARS circuit is then reserved. Meanwhile, both of the end hosts are listening for Link-Layer Discovery Protocol (LLDP)[5] packets sent on the wire by the OpenFlow switches they are connected to, and through parsing them the OpenFlow switch identifier and port number is discovered. The hosts register themselves to the site's ECSEL, completing the LAN topology built by the OpenFlow controller. At the time the circuit reservation begins, the OpenFlow controller computes the best path between the end hosts and the end point of the OSCARS circuit, and begins forwarding RDMA frames and performing the appropriate VLAN translation, thus establishing the RDMA flow between the two hosts. In the absence of any form of link layer flow control, establishment of an end-to-end circuit through services like ECSEL is necessary, in order to differentiate such loss-sensitive traffic from other best-effort flows, particularly if the best possible RoCE performance is desired.

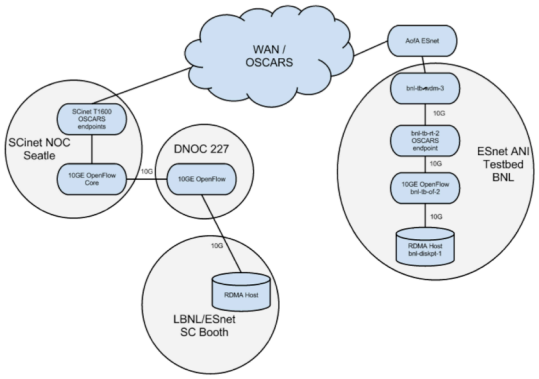


Figure 3 Network architecture for SC11 ECSEL demonstrator

C. SC11 SCinet Research Sandbox demonstration[6]

The SC11 SCinet SRS demonstration consisted of establishing a 10G layer 2 path between a data transfer node (DTN) located at BNL, Long Island, and another on the LBNL booth on the show floor, in Seattle, Washington. Each DTN utilized RoCE, a layer 2 RDMA protocol,, in order to achieve the necessary performance. Provisioning of the network was coordinated by ECSEL, using ESnet's OSCARS service for the intra-continental path, and OpenFlow switches for the LAN architecture. Full automation of data transfer was provided by the middleware and service Globus Online.

The demonstration was very successful. ECSEL was easy to deploy in an extremely dynamic environment such as Super Computing, thanks to its core code borrowed from OSCARS. The application itself, gridFTP using RoCE demonstrated that we could use a high performance layer 2 protocol with well behaved characteristics. It provided better than TCP throughput with very low CPU utilization. This was the motivation for following up on the ESCEL model.

IV. ARCHITECTURAL MODELS

As described above, ECSEL provides a model for how a campus implementing a data transfer service can leverage OpenFlow to provide flexible, end-to-end connectivity. For SC12, we explore new architectural models to support science that leverage SDN/OpenFlow by exploring more complicated use-cases, a) supporting multiple-science disciplines as typical in a supercomputing data center and b) flexible creation of virtual networks over the wide-area for dynamic science collaborations.

A. Muliple Science Centers

1) Description and Challenges

A typical supercomputing center or a large campus, like the National Labs, typically hosts multiple science disciplines and collaborations. Most of the science disciplines either produce data using the supercomputers by running simulations or move data for analysis to the supercomputing center from remote instruments. The data generated from the simulation, the raw data from the instruments and the analysis/results is then typically hosted at the supercomputer center for other scientists from the collaboration to access or transferred to another computing location for further analysis or local visualization. As mentioned before, large collaborations like the LHC produce vast amounts of data that are then analyzed and replicated among all continents and analyzed by thousands of scientists.

In order to facilitate network transfers, there are two models that are usually employed. The supercomputer centers, funded to support multiple projects, end up setting a ScienceDMZ enclave with a common infrastructure configured for high-speed data transfer (also called Data Transfer Nodes or

[5] http://en.wikipedia.org/wiki/Link_Layer_Discovery_Protocol

[6] http://esnetupdates.wordpress.com/2011/11/04/ecsel-leverages-openflow-to-demonstrate-new-network-directions/

DTNs). This shared resource is available to all science disciplines, but typically requires scheduling, manual management of data transfers from local storage to this dedicated DTN infrastructure and manual interaction with the local infrastructure management teams that own these resources. On the contrary, in campuses, the hardware is funded through projects and each science discipline deploys their own DTNs that they usually do not share with other science areas. To implement an effective ScienceDMZ, the local campus IT folks need to work closely with the scientists to put into place a secure, high-performance mechanism to transfer the data securely between the DMZ and the computational/storage infrastructure behind the security perimeter. Consequently, in many scenarios either the data transfer hosts are stuck behind the low-performance firewall limiting their throughput, or are a shared resource requiring manual intervention from multiple teams.

In addition to sharing the DTN resources, there is no central management system that allocates and rebalances the WAN bandwidth in response to the data transfers in progress. Most of these data transfers then end up using the routed IP network, and are subject to the performance vagaries of best effort networking.

To create a more automated and resilient Science DMZ model for multiple science collaborations on campus, we propose to explore an architectural model that leverages the capabilities of OpenFlow/SDN.

2) OpenFlow/SDN ScienceDMZ architectural model

The design of this architectural model is based on two important premises:

1. The science data is hosted by the scientist behind the security perimeter and only the portion of data that needs to be transferred to another location is exposed within the hosts on the ScienceDMZ
2. Wide Area resources like virtual-circuits are subject to use policies implemented and enforced by the local site administration.

The architectural model proposes putting a DTN redirector within the science DMZ. When the DTN redirector gets a data transfer request, the flow is redirected to the appropriate DTN within the security perimeter using flow rules. The firewall functions are bypassed by encapsulating the flow in one of the pre-approved VLANs.

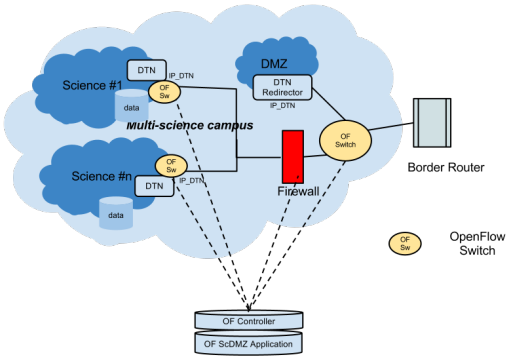


Figure 4 OpenFlow-based end-site architecture to support multiple science disciplines

These pre-approved VLANs are locally configured to bypass the firewall function. The VLAN translation function of OpenFlow is leveraged to change the incoming header, and flow is re-directed without rewriting the IP header. This approach enables science collaborations to maintain their DTN alongside the data/storage infrastructure. The site network administrators have full flexibility to manage policies, including security, at the centralized OpenFlow controller. All local policy functions, the data transfer workflow, authentication/authorization functions and configuration interfaces can be implemented as an application on the OpenFlow controller, which we name as the "OpenFlow ScienceDMZ Application". This approach is a natural extension of ECSEL, which primarily supported end-to-end layer 2 protocols between two end-points, and allows it to scale across multiple science disciplines.

The proposed workflow also reduces the ongoing support burden to the campus IT administrators while improving the security characteristics as well. The firewall is configured statically to bypass certain VLANs that are local to the site though the VLAN encapsulation and assignment managed by the OpenFlow controller application.

The sites supporting multiple science disciplines will benefit a lot from this approach. All the science data will now be managed at one location and by the science, while the complexity of staging the data at the DMZ DTN from the primary data store protected behind the security perimeter can be eliminated. Each science discipline can now not only manage their own data and DTN architecture, but also support custom data transfer protocols that may be unique to that collaboration.

B. Dynamic Science Collaborations

1) Description and Challenges

Science collaborations tackling large problems like Climate or sharing one-of-a-kind instruments like the LHC develop a more formalized architecture for hosting and distributing data. Even though these architectures are highly dependent on a high-performance network, they are typically managed

completely by the scientists or IT administrators that are part of the collaboration itself. The networking-savvy collaborations typically leverage the point-to-point inter-domain guaranteed bandwidth circuits provided by the R&E networks to get better performance, but they are in the minority. There are many others that just use the general IP routing service and experience poor performance, or take to alternate means of transporting their data like shipping hard-drives.

Existing technologies like IPSec, VPLS or L2TP are currently used to bridge multiple sites together in a virtual private network (VPN). The multiple sites connected by this tunneling protocol or virtual routers can share IP routing information between the sites, so they can route traffic across the VPN. Some of these technologies are overlay technologies that do not require participation by the service provider other than providing a routed IP connection. Other technologies, are VPNs offered by the service provider, and require complex routing configurations in order to work.

In most of the science collaborations, an approach that involves a lot of complex site configuration, management by on-site IT teams is not feasible due to the management complexity, and the ephemeral nature of these collaborations. In addition, it is not easy to apply policies to verify that only science traffic is utilizing these inter-site connections, especially when each site is managed by a different administrative entity.

The goal is to create an infrastructure on-demand for a flexible science collaboration tied together with a virtual network ensures all existing capabilities of high-performance. In addition, the requirements from the science collaborations is 1) flexibility for them to manage their wide area bandwidth 2) easy virtual private network setup between collaborators that does not require huge configuration overhead 3) flexibility to use either layer 2 or routed IP services between the collaboration sites, providing higher performance to standard IP applications.

2) 'One-virtual-switch' network virtualization model

For dynamic science collaborations, it is common practice to establish connectivity between the sites through dynamic or static point-to-point circuits. Thus, for large science collaborations, each site ends up supporting multiple point-to-point circuits with customized access, IP addresses, and security policies, all managed manually.

We tackle this limitation by creating a new abstraction – "one virtual switch" representing the WAN [Figure 5]. This abstraction is simple, atomic, ***programmable***, and network-wide. The abstraction enables multiple sites belonging to a collaboration to build logical circuits to the virtual ports of the WAN virtual switch. The virtual switch, in turn, exposes an OpenFlow programmatic interface that enables the sites to easily program and redirect data flows to the appropriate site and end-point dynamically.

One of the key elements of this architecture is the implementation of the physical infrastructure that can be virtualized in software to appear as a single switch. This is accomplished by deploying OF enabled devices at the customer facing edge of the WAN network with a programmable WAN interconnect fabric implemented by guaranteed bandwidth OSCARS circuits connecting them. The circuits can be modified dynamically behind the scenes, and protected in order to implement a very adaptable and reliable fabric. Unlike port identifiers of a real switch that are typically vendor and switch specific, the ports on the virtual switch can be abstract and named logically, in order to make it much more intuitive and usable. This provides the topological foundation over which the virtual infrastructure is built.

To provide dynamic control of the infrastructure, we create a simplified OpenFlow control API that exposes the virtual switch and its logical topology to the end-sites. Using OpenFlow, the end-sites or science collaborations can dynamically program switching of flows between the various end-sites using the virtual topology, and shielded from the complex multi-domain physical connectivity view. With this element of control, the collaboration is able to multiplex multiple flows over a single site connection and virtually switch them to the appropriate destination as the data flow needs of the collaboration change.

The previous model of site-to-site negotiation, as implemented in ECSEL, is still supported to ensure authorized admission control as well as for resource management of the shared wide area resources, namely the bandwidth reserved for the site's connectivity into the virtual switch. It is unclear at the moment if the entire science collaboration can work with one controller, or if it is more advantageous for each site to manage their resources, with their own controller. Tradeoffs of each approach will be part of active experimentation once the virtualization architecture is implemented for the demonstration.

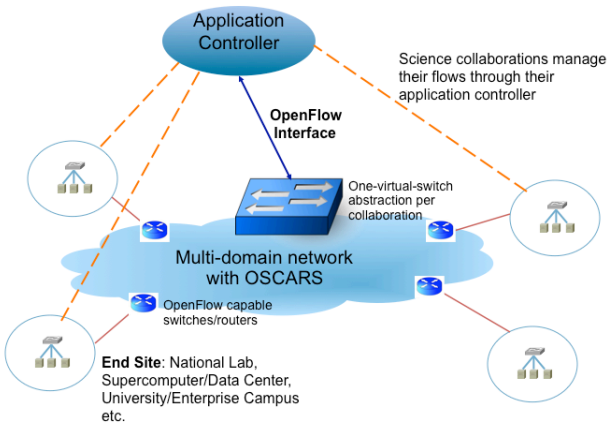


Figure 5 one-virtual-switch abstractions for multi-science collaborations

The figure below [Figure 6] shows implementation details of how the virtual view and the physical implementation relate to each other. The one-virtual-switch abstraction, like a physical OpenFlow switch, provides both a management interface and a control interface. The control interface, using the OpenFlow protocol, enables dynamic configuration and control of flows, similar to a real switch. The management interface allows site-administrators to provide the parameters of the collaboration enabling the virtual switch abstraction to be setup for the applications.

This approach to network virtualization, through creation of a simple 'one-virtual-switch' abstraction enables the research collaborators and network administrators to treat the wide-area network as a local switch with similar methods of switching and control:

- The OSCARS wide-area traffic-engineering service creates a flexible backplane between the various OpenFlow switches, enabling the multipoint transport between the edges of the WAN.
- The OpenFlow switches in the WAN, at the customer edges, perform flexible flow switching and VLAN translation, hiding the multi-point complexity from the end-sites and the application.
- The end-site OpenFlow enabled applications program flows on the one-virtual-switch in the same way done locally.

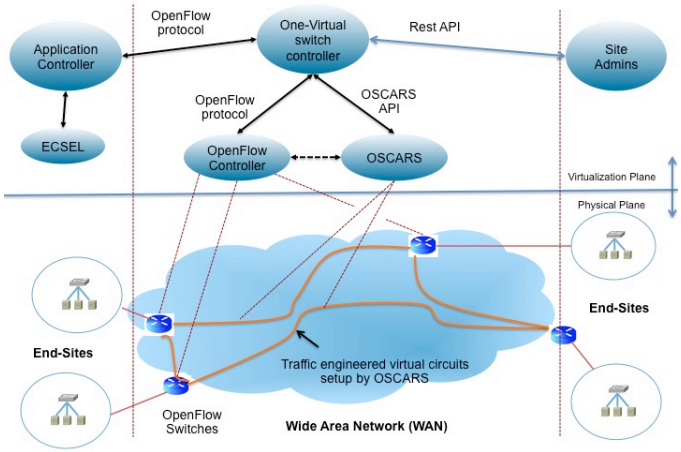


Figure 6 End-to-end virtualization using One-Virtual-Switch, ECSEL and OSCARS

V. SC12 DEMONSTRATION

Building upon the already demonstrated SDN concept for end-site, ECSEL at SC11, the SC12 demonstration will focus on the dynamic science collaboration architectural model as defined above. An extension of ECSEL to show the multi-science campus is also a stretch goal. These two demonstrations will showcase implementations that will enable scientists to deal with the data-deluge in a flexible and scalable manner. We will leverage existing data transfer protocols like GridFTP to demonstrate the conceptual model described above.

VI. RELATED WORK

The broad concepts and challenges for data-intensive science described in this paper have been widely understood and is a topic of active research in the R&E community. Some of the widely accepted research solutions that have transitioned into production networks include Hybrid Networking, OSCARS[6], IDC among others. New projects have recently tried to tackle some of the end-site issues for non-OpenFlow technologies like ESCPS [7] funded by DOE. OpenFlow in the wide-area has been recently gaining a lot of attention – Internet2's NDDI [8] project creates point to point links over the wide-area and Google[7] has recently announced using OpenFlow to be the prime network technology responsible for moving large data sets between their data centers over the wide-area. Since the OpenFlow and Software-defined paradigm are fairly new, exploring how these paradigms will apply to existing problem sets, and create new value is a worthwhile enterprise.

ACKNOWLEDGMENT

We would like to acknowledge our collaborators for the SC11 demonstration: Matrin Swany, Ezra Kissel, Ahmed Hassany and Brian Tierney, William Johnston's detailed comments and DOE ASCR network research funding to explore OpenFlow's applicability to the wide-area.

REFERENCES

[1] William Johnston, "The Square Kilometer Array – A next generation scientific instrument and its implications for networks," TNC 2012, https://tnc2012.terena.org/core/presentation

[2] Eli Dart, Brian Tierney, editors, "HEP (High Energy Physics) Network Requirements Workshop, August 2009 - Final Report", ESnet Network Requirements Workshop, August 27, 2009, LBNL LBNL-3397E.

[3] T. Abe et. al. "BELLE II Technical Design Report", KEK Report 2010- 1, 2010 , http://arxiv.org/abs/1011.0352v1

[4] Eli Dart, Brian Tierney, editors, "BER (Biological and Environmental Research) Network Requirements Workshop, April 2010 - Final Report", ESnet Network Requirements Workshop, April 29, 2010.

[5] E. Kissel and M. Swany. The extensible session protocol: A protocol for future internet architectures. Technical Report UDEL-2012/001, Dept. of CIS, University of Delaware, 2012

[6] C. Guok, D. Robertson, M. Thompson, J. Lee, B. Tierney, and W. Johnston. Intra and interdomain circuit provisioning using the oscars

[7] http://gigaom.com/cloud/how-google-is-using-openflow-to-lower-its-network-costs/

reservation system. In Third International Conference on Broadband Communications Networks and Systems, IEEE/ICST, October 2006

[7] End Site Control Plane Service, https://plone3.fnal.gov/P0/ESCPS

[8] Network Development and Deployment initiative, http://www.internet2.edu/network/ose/