

The Network Development and Deployment Initiative: Expanding the Breadth and Reach of Internet2 Network Services Through the Development of the Open Science, Scholarship, and Services Exchange

Executive Summary:

Internet2, Indiana University (IU) and the Clean Slate Program at Stanford University have formed the Network Development and Deployment Initiative (NDDI), a partnership to create a new network platform and complementary software, which together will support global scientific research in a revolutionary new way. Through substantial investments by each of the partners, the NDDI will yield a new Internet2 Network service called the Open Science, Scholarship and Services Exchange (OS³E). OS³E and the NDDI capabilities will be developed and interconnected with links to Europe, Canada, South America and Asia, through coordinating international partners like RNP in Brazil, CANARIE in Canada, GÉANT in Europe, and JGNX in Japan, with additional service partners to be identified.

Abstract:

Internet2, in partnership with IU, has provided high bandwidth, superior quality, Layer 3 network services for the American research and education (R&E) community since its inception. It has also provided international transit network services across the Internet2 backbone to international peers. Internet2 and IU deployed advanced network services, such as IPv6 and QoS, long before they became commonplace on commodity networks, and have provided wide area test-beds for the network research community, including support for projects such as PlanetLab, HOPI, and GENI. Internet2 and IU have driven the development of new types of services, such as Layer 2 “circuit” services provisioned automatically through software (IDC/ION) and multi-layer multi-network performance monitoring services (perfSONAR). The unifying theme of Internet2’s network offerings has always included providing network connectivity by the best available network transport technologies.

As Internet2 begins deployment of its new network, three use cases have emerged that suggest a new suite of network services:

1. Internet2 has long seen strong demand for experimental networking in support of network research, for new paradigms to support data-intensive science and for broad-scale deployment of disruptive network development opportunities. The emerging GENI investment in OpenFlow and the GENI applications interface are both increasingly important tools for network researchers.
2. There has also been strong demand for broad access to VLAN-based network infrastructure dedicated to research that supports persistent, flexible, unrestricted VLANs.

3. There is a growing need for global reach to enable scientists to use these capabilities with collaborators around the world.

Internet2, IU, and the Clean Slate program believe a common infrastructure that provides platforms for network research, a distributed open science exchange point to support domain researchers, and global reach is possible through virtual “slices” on commodity hardware using Software-Defined Networking (SDN) architecture. No such suite of services currently exists beyond prototype networks, but it is within the Internet2 consortium’s reach to be the first to complete the software, implementation and operational leadership of an open science exchange with global reach to meet the needs of the R&E community.

This document proposes that emerging needs of the Internet2 membership that are described in the three aforementioned use cases—a point-to-point and multipoint VLAN service, a network to support and deploy experimental network research, and a distributed global science exchange capability —be met through the development of the Network Development and Deployment Initiative (NDDI) substrate and the Internet2 OSE, using SDN technologies such as OpenFlow. As this advanced networking platform and attendant services are deployed, network researchers will have the opportunity to reinvent the technical underpinnings of the Internet2 network, while domain researchers collaboratively pursue education and research across the globe.

Emerging Needs of the Internet2 Membership

The networking needs of the Internet2 membership have long included a national Layer 3 network. Concurrent with the development of the new Internet2 network, enabled by the ARRA BTOP investment in Internet2, Internet2 has identified three emerging use cases from the Internet2 membership.

Persistent VLAN Service with Global Reach

Community leaders have articulated a need for a Layer 2 service that allows flexible and persistent VLAN’s between points of access on the Internet2 network. Internet2’s ION service enables dynamically provisioned VLANS between points of access on the Internet2 network and through peer networks, but it is not currently configured to enable persistent VLANs. This use case has been advocated by the Architecture and Operations Advisory Council (AOAC) and investigated by the Layer 2 working group chartered by the Network Technical Advisory Committee (NTAC). A draft whitepaper entitled “NTAC Evaluation of Internet2 Layer-2 Service” has been endorsed by the NTAC and the AOAC.

The AOAC charge enumerated the following desires: Access to the service should be provided through very inexpensive 10 GbE ports. The service should be built on very dense and inexpensive “throwaway” switches at each Point of Presence (PoP) with potential to support SDN (e.g. OpenFlow). VLAN configuration needs to be user-controllable. The service needs to have a priority queuing mechanism, even

though almost all of the traffic is expected to be best-effort. The service also needs a less-than-best-effort scavenger option.

Two candidate uses, which could legitimately be categorized as lying at opposite ends of a continuum, for this new service are: 1) the establishment of longer-term point-to-point paths (or possibly broadcast domains) which may be used for production services and 2) the configuration of traffic-engineered paths for high-bandwidth flows between two or more end hosts. The former could be characterized as “network-based” usage, whereas the latter could be characterized as “host-based” usage.

Independent of existing service offerings, a persistent VLAN service with global reach meeting the use case outlined above could be provided in several ways:

- The VLAN service offering could be an MPLS overlay on the existing Internet2 IP infrastructure. That would require backhaul for the additional 10G ports used for the VLAN service; a closely related alternative would be to drop smaller IP routers (such as Juniper MX 80s) near each connector to provide better resiliency. However, either solution wouldn't provide much additional redundancy against primary research connectivity failure, since the same ports and equipment are used; presumably if the primary path fails, there would be a large probability that this VLAN service would fail too.
- The service could be provided by a separate set of Ethernet switches connected by separate long-haul circuits, as NLR's FrameNet does today. The former instantiation of the Internet2 ION service (built atop Ciena CoreDirectors) represented a different instantiation of this idea, where the VLANs could be protected in a time-division fashion on the long haul circuits to provide absolute bandwidth and latency guarantees. Under this approach, switches would provide cheaper 10GE ports than a routed solution and could be distributed more widely than the current IP infrastructure, placed closer to connectors. This would come at the cost of the long-haul interconnection circuits, and additional space, power and maintenance. On the other hand, there are potentially fewer long-haul circuits needed for backhaul, and the costs for the components and power requirements are lower.
- Rather than using single-manufacturer Ethernet switches to provide a Layer 2 fabric, a set of switches with SDN capabilities (e.g. OpenFlow) could be used. The advantage of using SDN is that the switches become commodity components that can be more easily replaced as technology advances without requiring changes to the control software developed to manage the network.

The method by which VLANs are instantiated from a user (customer) perspective is independent of the particular implementation. Alternative implementations include:

- Manually instantiated by a NOC
- Automatically instantiated using OSCARS (the software which implements the ION service) or something resembling the Sherpa software used for NLR's Dynamic VLAN Service (originally designed for the separate Ethernet infrastructure)

Support for At-Scale Network Research

As noted in the mission statement for Global Environment for Network Innovations (GENI) [GENI], network researchers have long sought to support at-scale experimentation on shared, heterogeneous, highly instrumented infrastructure, to enable deep programmability throughout the network and to provide collaborative and exploratory environments. In order to transform how networks are built and operate, network researchers need a substrate on which to experiment with breakable, large scale infrastructure, akin to the environment that existed before NSFnet was commercialized, giving them the opportunity to explore radically new ideas without being burdened with incumbent, production traffic but with the option to introduce that production traffic where needed.

The core requirements for such a network substrate include:

- The ability to slice the network into multiple virtual networks, making at-scale infrastructure affordable to individual researchers
- The ability to provision point-to-point circuits with known properties (such as latency and bandwidth guarantees)
- The ability to define how networks function through user-controlled and written software
- Well-defined APIs through which to interface with inexpensive, commodity switches

A network research service meeting the use case outlined above could be provided in a few ways, including all the ways stated above for the VLAN service. In addition:

1. The point-to-point dedicated circuits could be provided by a point-to-point VLAN service with bandwidth guarantees, or by waves on the Layer 1 network.
2. A nationwide OpenFlow network, connected to interested regionals and campuses and international OpenFlow networks would allow network researchers to not only create point-to-point circuits, but to be able to control the path over the wide-area infrastructure in a safe manner.

This last approach directly supports the two OpenFlow use cases, and (given sufficient internode capacity and stability in the OpenFlow and OpenFlow virtualization (FlowVisor [FV]) implementations) could support the other network

research use cases as well. Those projects could also change over time to interface directly with OpenFlow and have more control over their wide-area path.

Open Science Exchange Point

The High Energy Nuclear Physics (HENP) community has focused much of its efforts on supporting the several experiments of the Large Hadron Collider (LHC) project. The original LHC network design focused on a three-level tier-based hierarchy of sites for data distribution and computation, known as the MONARC [MONARC] model. There is growing consensus in the LHC Physics community that the original hierarchical tiered model for data distribution is not being used in practice, replaced by more of a mesh of flows from Tier 3 sites to any Tier 2 or from Tier 2's to any Tier 1 or any other Tier 2. Thus, having point to point circuits from a Tier 2 to "their" Tier 1 does not cover all (or perhaps even the majority) of traffic from that Tier 2 to other data storage sites. The current concept is to replace the point-to-point from a Tier 2 to a Tier 1 with a connection to a "LHC Open Network Exchange" (LHCONE) [LHCONE], and the Tier 2 could peer with many or all Tier 1's. This peering could be permanent, it could be dynamic, and either of those could be done using best-effort capacity, or with bandwidth guarantees.

The concept of an "open science exchange point" is not exclusive to LHC; it could be used by other similar "data intensive science" disciplines that have large data centers that need to communicate. For example, climate and astronomy applications have this characteristic.

There are a number of different ways to implement an open science exchange point service, but the VLAN service provides a direct template, with the addition that the science exchange points desire multipoint VLANs in addition to point-to-point. Thus VPLS might need to be used in the case of an overlay network. The other potentially differentiating characteristic is that this service is inherently international; data intensive science does not stop at the borders of any particular country, and requires transfers from centers in different countries. Thus any US service would be integrated with services provided in Europe, Asia, and elsewhere.

However, the rest follows directly: This service could be provided in a straightforward manner by a set of Ethernet switches, and similarly by the use of an OpenFlow switch base, so long as the ability to allocate bandwidth along the intervening path as needed is under the control of the LHC community as a whole.

NDDI Substrate

Internet2, Indiana University, and the Clean Slate program will build a broadly purposed and national scale National Design and Development Initiative (NDDI) substrate, atop its new ARRA BTOP-funded network infrastructure. Utilizing the software-defined networking approach to allow broad deployment of Layer 2 services and testing of new network ideas, the NDDI substrate will meet the network needs articulated in the aforementioned use cases: enable a sliceable

network research platform at scale, enable data intensive science, and support Layer 2 connectivity. The NDDI substrate will extend to global partners, including Europe, Asia and South America. This advanced networking platform will meet the needs of all three of the aforementioned use cases and position Internet2 to support the needs of the 21st century R&E community.

This single new service is envisioned to have the following properties:

1. Uses a common network infrastructure in a cost efficient manner.
2. Relies on infrastructures other than traditional Layer 3 routed infrastructures.
3. Enables the provisioning of Layer 2 circuits with and without bandwidth guarantees through software and/or web interface for short and long periods of time.
4. Enables the ability to “slice” the network into multiple virtual networks:
 - some of which are tuned for production high bandwidth flows;
 - some of which are designed to support innovative network research;
 - all of which are protected from and do not interfere with one another.
5. Moves towards reliance on less expensive (both in terms of capital costs and operational costs) Layer 2 switches rather than “big iron” hardware.
6. Is operated in an open and transparent fashion.

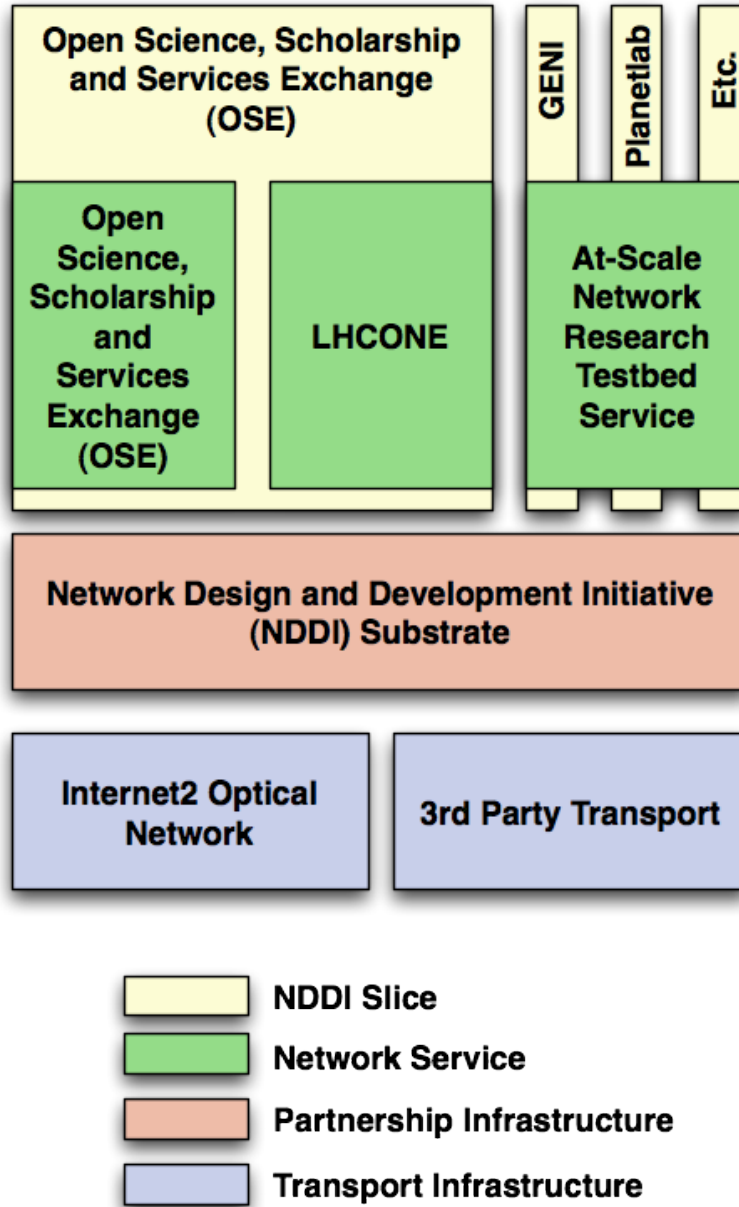


Figure 1: Architectural Overview

NDDI Substrate Node Design

Each NDDI substrate node is expected to consist of a single, fixed form-factor switch. A number of switches are becoming available that support 48 10 Gbps ports and, in some cases, 4 40 Gbps ports, in a 1-2 rack unit form factor and we anticipate that a number of these switches will support OpenFlow. That is more than enough ports to support a backbone with expansion capabilities, and multiple connections to connectors, external networks, virtual communities, and network research projects. The cost per 10 GbE port for these switches is likely to be in the \$250-\$500 range..

Some subset of NDDI substrate nodes will also have 1 or 2 servers. Some servers will serve as OpenFlow controllers. Other servers will serve as measurement nodes. It is unclear at this point how many nodes will require controllers or whether a single server can serve both as a controller and a measurement node. It might be prudent, for example, to organize NDDI substrate nodes into areas based on geography, and deploy redundant controllers in each of those areas. The requirements for such servers are not likely to be excessive, meaning that servers in the range of \$1,000 to \$2,000 are likely to suffice.

NDDI Substrate Network Footprint

The NDDI substrate will begin with a small number of nodes scattered around the United States, including nodes in Chicago and New York City. The NDDI substrate footprint is expected to grow to include all Internet2 router nodes, all Internet2 3-way junction nodes, all nodes at which an Internet2 Connector connects to the Internet2 backbone network, and all the major US exchange points. It is possible the NDDI substrate will also include a node in Vancouver, to interconnect Canadian LHC sites to LHCONE-NA. Likewise, it is possible the NDDI substrate will include a node in Miami, to support interconnection with network deployments in South America.

Each node is expected to be initially connected to a subset of other nodes on the new network footprint by 2 10 Gbps waves. The bandwidth between nodes is expected to grow to 4 10 Gbps waves over time.

A sample map of likely NDDI substrate nodes to be built over time is shown below. This map is an early draft and may change significantly before full implementation. It is expected that the total number of nodes will range between 30 and 40 sites.



Figure 2: Aspiration NDDI Substrate Network Footprint

NDDI Substrate Costs

The cost to design and develop the NDDI substrate has not yet been determined, but will include capital expenditures, software development costs, and operating expenses. Capital expenditures will include switches (up to 40 switches at up to \$30,000 each), servers (up to 80 servers at up to \$2,000 each), and long-haul optics (costs TBD). Operational expenditures will include power (mostly DC) and space (up to 4U per node) (costs TBD). Software development costs include making OpenFlow capable of supporting production network traffic (likely born by Indiana University), developing Flowvisor (likely born by the Clean Slate Program), and integrating OSCARS with OpenFlow (likely born by Internet2) (costs TBD).

NDDI Substrate Cost Recovery Model

The cost recovery model for the NDDI substrate is under development. It is expected that the fee model will be specific to each service implemented on the substrate, and evolve over time as a richer understanding of the market emerges. Initially, the fee structure is likely to resemble a port fee model, such as is commonly employed at exchange points today, including a portion to cover long haul optical costs.

Implementation Plan

An implementation roadmap for the NDDI substrate is still to be determined.

At a high level, it is envisioned that the NDDI substrate will evolve over time along multiple axes, including hardware, software, bandwidth, reach, business model, and feature set. Moreover, it is envisioned that related Internet2 services (e.g. ION, prototype LHCONe-NA) and NSF-funded research projects (e.g. DYNES) will evolve over time and become integrated with the NDDI substrate.

Initial thoughts on the roadmap for NDDI substrate hardware and software are provided below.

NDDI Substrate Hardware

Nodes: The initial set of switches selected to support the NDDI substrate are likely to be replaced within 12-18 months. The initial set of nodes is likely to include one measurement server per node and two controller servers in total.

Footprint: The NDDI substrate will begin initially with a small (~6) number of nodes, including NYC and Chicago. Over time it will grow into a full deployment of 30-40 nodes.

Bandwidth: The NDDI substrate will begin initially with a ring of 2x10 Gbps waves. Over time it will grow into a partial mesh of 4x10 Gbps waves.

NDDI Substrate Software

The first phase of NDDI substrate software will provide the Open Science, Scholarship and Services Exchange (OS³E), an intra-domain dynamically configured layer 2 virtual circuit service, allowing users to provision VLANs on an OpenFlow based infrastructure across the Internet2 OpenFlow domain. QOS support in the initial phase of the project is expected to be limited due to limitations in the OpenFlow 1.0 specification. The service is expected to include both a GUI that users can access to create and alter VLAN configurations, as well as an API that can be used for programmatic provisioning of virtual circuits. It is also anticipated including some degree of path resiliency support in this phase, allowing virtual circuits to be automatically re-routed in the event of a backbone link failure, where capacity is available. The software required to deliver this service is expected to be developed and deployed into production in a 6-month timeframe.

The second phase of NDDI substrate software (which will be developed in parallel with the first phase but implemented once the first phase stabilizes) will support OS³E inter-domain circuit provisioning by leveraging the existing OSCARS implementation of the IDC protocol (expected to evolve over time to conform with the OGF NSI WG protocol as it emerges) and a hardened FlowVisor to allow for network research slices. This may support a future service to support the creation of at-scale network research testbeds. Additionally, further development will add richer QOS support as the OpenFlow standard matures and evolves. The timeframe for delivery of the QOS capabilities is dependent on OpenFlow specification and

possibly hardware constraints, although this functionality is expected to be available in the next major revision of the OpenFlow specification.

Appendix 1: Software-Defined Networking Overview

According to the Open Networking Foundation (ONF), Software-Defined Networking (SDN) is a new approach to networking that gives network operators better control over their networks allowing them to optimize network behavior to best serve their and their customers needs. The SDN approach was developed through a research collaboration among Stanford University, the University of California Berkeley, the University of Washington, Princeton University, Washington University in Saint Louis, and MIT. On March 21st, 2011, the Open Networking Foundation was announced to continue the standardization of the OpenFlow protocol, with a mission to promote the SDN approach to networking.

A key instantiation of the SDN approach is a technology called OpenFlow [OF]. Modern switches and routers are made up of two distinct parts, a control-plane and a data-plane. The control-plane is the operating system responsible for running services such as routing and switching protocols and it typically runs on a general purpose CPU. The data-plane is responsible for actually forwarding packet and is typically supported by a special purpose chip or ASIC. In most network devices, the interface between the control-plane and data-plane is proprietary and strictly internal to the device. This limits the ability of network operators to customize the packet forwarding behavior to a large degree.

OpenFlow remedies this constraint in a very simple and elegant manner. OpenFlow defines a protocol by which an external device, commonly called a controller, can add, remove and modify forwarding table entries in the data-plane of a switch. A controller, which is typically external to the switch and usually a simple PC, can make entries in the forwarding tables of one or more switches, directing where traffic is to flow. The communication between the switch and the controller is through a secure channel using the OpenFlow protocol. The OpenFlow table consists of a set of rules for forwarding packets. Each rule consists of a match for multiple header fields in a packet, an action to take if a packet matches the rule, and a set of statistics associated with each rule. The actions are extensible, but typically consist of operations like: 1) drop the frame, 2) forward the frame to an output port, or 3) send the frame to the controller, for example to redirect the output of such frames that follow later. Additional actions might be to rewrite parts of the frame, or to add the frame to some priority class. The fundamental impact of the external controller is to have control that can be changed and experimented relatively easily through software programming in a standard PC development environment.

An OpenFlow switch therefore consists basically of three components: one or more flow tables with associated actions and counters, a secure channel for communicating with the controller, and support for the OpenFlow protocol. Note

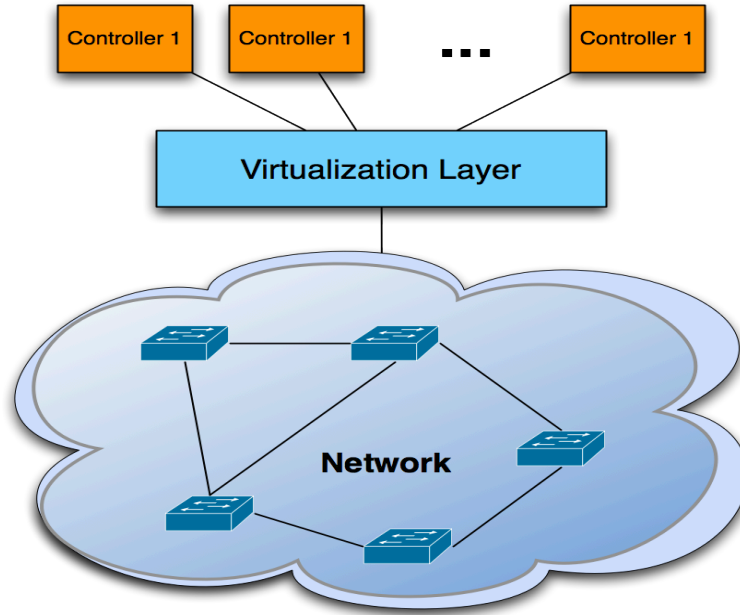
that most of this discussion pertains to an Ethernet switch, almost always thought of as a Layer 2 device, but an OpenFlow switch is essentially layerless in concept. Its flow tables have information from the port level to the TCP layer. Note that OpenFlow also exists for circuit devices that use SONET (see [COF], for example). The flow tables for such devices include information about lamdas, VCGs, time-slots, and signal type. This document, however, will focus on Ethernet switches.

It is clear that a classic Ethernet switch would need to be modified to support OpenFlow, adding the flow tables, and both the secure communications channels and the OpenFlow protocol. When OpenFlow was first being developed, there were no vendors supporting the requirements of OpenFlow. Today OpenFlow is supported by several major switch/router vendors.

How does one build a network with OpenFlow? The basic concept is to deploy OpenFlow Ethernet switches and at least one controller. Typical implementations of OpenFlow on an Ethernet switch isolate the common operations of the switch from the OpenFlow component. This means that the usual operations of the switch are not disrupted by the OpenFlow operations, and therefore can be deployed in networks that are used in production. Several campuses have deployed such switches in local, building level networks. The controllers can also be deployed in a flexible manner. One could use a single controller, for example, or one could deploy a controller for each switch. The basic concept remains the same, (at least one) controller sitting above a network and controlling the Ethernet switches.

The controller is typically operated by some administrative group or virtual community, for example a network research group examining new network protocols, or a scientific group providing high bandwidth paths. What if an additional administrative domain wants to control such a network? The developers of OpenFlow have created a virtualization layer in between the network devices and the OpenFlow controllers. The virtualization layer is provided by devices called FlowVisors (see [FV]), allowing multiple OpenFlow controllers to control the switches. A FlowVisor allocates and isolates slices of the network to each controller, where one controller cannot interfere with another controller's slice. Slices can be a set of ports on each switch, or a set of switches, etc.

The basic idea behind an OpenFlow network is pictured in the following diagram, with multiple controllers associated with different communities sitting above virtualization layer that provides slices of the network to each of the controllers:



Note that much of the software associated with the controllers and the FlowVisor is open source software and can be implemented on inexpensive PCs. Moreover, there is additional work evolving to insure redundancy. For example, having one controller take over from an existing controller, either by administrative command, or in the case of failure.

Consider a national scale network of OpenFlow-enabled switches sitting under an OpenFlow virtualization layer. Above that layer could sit multiple controllers each having a slice of the network. One might provide a classic user controlled Ethernet VLAN network, or one might provide services of the Internet2 ION network. Yet another might provide an experimental network for network researchers, and yet another might provide bandwidth for, and be totally controlled by, a virtual community such as the LHC. One might even provide traditional IP peering. Each of these controllers would be independent and be isolated from each other. Moreover, each of these communities can create their own software platforms that sit above OpenFlow rather than writing the software to particular hardware platforms. If OpenFlow switches are upgraded to switches from a different vendor, no changes should be required to the control software because OpenFlow is a multi-vendor standard.

One thing that is missing from OpenFlow is the traditional inter-domain functionality. At this time there is no obvious way to inter-connect two different OpenFlow networks under different administrative domains and have the traditional isolation and protection methods in place mechanisms in place. For example, there are no signaling capabilities associated with OpenFlow for crossing different administrative domains. This does not affect the internal workings of the

network, however, and there are several approaches to dealing with this issue. One is to simply inter-connect from an internal OpenFlow point of view, with any particular controller domain participating or not, and to use the capabilities of the FlowVisor to limit access. The other is to develop a minimal approach to inter-domain capabilities using some of the concepts already developed through other Internet2 projects (for example, the OSCARS software that supports ION). Either approach could provide a cohesive network with global-reach.

Another significant possibility for OpenFlow is the commoditization of network components. There are already commodity switches, with a relatively minimal and potentially open-source firmware that only supports OpenFlow that could be with open-source controller software and open-source software to support classic Ethernet switching, creating essentially an open software switch that can be customized to meet the requirements of the Internet2 community. Other existing protocols, including routing, or even additional new network protocols, could be layered on top of that system. Such a system could be less expensive than existing switches because they would only include the basic functionality required by the Internet2 community. This is exactly the path taken by the PC world and many of today's PCs, especially in the server world, run open source software in very important and fundamental ways.

References

- [FV] *FlowVisor: A Network Virtualization Llayer*,
<http://www.OpenFlow.org/downloads/technicalreports/OpenFlow-tr-2009-1-flowvisor.pdf>, Rob Sherwood, Glen Gibb, Kok-Kiong Yap, Guido Apenzeller, Martin Casado, Nick McKeown, Guru Parulkar, October 14, 2009
- [OF] *OpenFlow: Enabling Innovation in Campus Networks*,
<http://www.OpenFlow.org/documents/OpenFlow-wp-latest.pdf>, Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner, March 14, 2008
- [COF] *Packet and Circuit Network Convergence with OpenFlow*,
http://www.OpenFlow.org/wk/images/4/46/OpenFlow-OFC10_invited.pdf, Saurav Das, Guru Parulkar, Nick McKeown, Preeti Singh, Daniel Getachew, Lyndon Ong,
- [GENI] <http://www.geni.net/>
- [MONARC] <http://monarc.web.cern.ch/MONARC/>
- [LHCONE] <http://lhcone.net/>