

A Case for Expanding OpenFlow/SDN Deployments On University Campuses

Lead Author:

Matt Davy, Indiana University

Contributors:

Guru Parulkar, Stanford University

Johan van Reijendam, Stanford University

Dan Schmiedt, Clemson University

Russ Clark, Georgia Tech

Chris Teng, Princeton University

Ivan Seskar, Rutgers University

Patrick Christian, University of Wisconsin

Ian Cote, University of Washington

George China, Big Switch Networks

I. Executive Summary

In May of 2009, the Clean Slate Program at Stanford University, in partnership with 7 other universities, proposed a project to deploy the OpenFlow technology on campus networks in support of GENI. The proposal for the project laid out the following vision:

“This project is motivated by the belief that if we can open up campus networks for innovation by researchers and network administrators, we will unleash tremendous untapped potential and will fundamentally change the field of networking. If we can move from a culture of closed, proprietary, expensive infrastructure (with very long innovation cycles) to an open infrastructure enabling rapid innovations by all stakeholders, we will create a market place for ideas allowing the best ideas to win”

This was certainly a very bold vision for a relatively immature technology, but what has happened in the past 2 years is nothing short of amazing with much progress being made towards this vision.

This idea of open, programmable networks that enable rapid innovation has clearly resonated throughout the marketplace. In April of this year, the Open Networking Foundation (ONF) was created with 17 initial members to adopt standards for Software Defined Networking (SDN) technologies such as OpenFlow. Just two months later, the ONF had 36 members including every major networking vendor. At Interop in May, 15 vendors demonstrated switches and controllers supporting OpenFlow, NEC's OpenFlow switch won “Best of Show” and according to one source Interop could have been called the “OpenFlow Show”. The 7 campuses that have deployed OpenFlow have seen first-hand the potential it has to solve many of the difficult challenges they are facing in managing their campus networks and they are very anxious to expand their efforts.

The timing and conditions are ideal to take the next step towards fully realizing the vision laid out in the original project proposal. Universities are ideally positioned to take a leadership role in developing and deploying the next-generation of networking technologies based on the SDN paradigm. In this paper we describe the major challenges campus networks are facing, explain how SDN can help solve these challenges, describe the current status of SDN and OpenFlow and present a strategy for expanding SDN deployments to many more campuses.

II. Problem Statement

In this section we present the challenges facing university network administrators in providing state-of-the-art network infrastructure to the many stakeholders on their campuses. The key point is that the lack of access to innovate in the network results in the inability to keep pace with user needs and to keep costs under control.

a. Mismatched Innovation Cycles

For network architects and engineers at large universities, the complex set of problems they face in managing today's university networks can be overwhelming. Everything is networked. From vending machines to exercise machines, from gaming systems to heart monitoring systems, from cellphones to fire alarms, they all require IP networking. Today's university networks must be secure enough to support medical and payment card systems, but open enough to support cutting edge research and rapidly evolving consumer technologies. In a few short years, mobile devices such as cellular phones and tablets have overtaken PCs as the primary means for users to access the network. Virtual servers, with the flexibility to rapidly create, move and remove servers, have become the norm. Large universities represent one of the most complex and challenging network environments in the world and many of the changes that have contributed to this have happened in just the last 5 years.

During this same timeframe networking technology has changed relatively little in comparison to the computing technology it is supporting. Today's networks are built and managed largely the same way they were 5 years ago. Technologies such as MPLS VPNs have been added to extend a virtual network beyond a single VLAN. Vendors have added virtualization technology to products such as firewalls that allow multiple contexts to coexist on a single piece of hardware. But these are incremental advances, whereas the advances in computing, such as virtualization and mobility, have been revolutionary.

b. Vertical Integration

One major difference between the computing and networking industries is that the networking industry is vertically integrated much like the mainframe era in computing. Networking vendors bundle hardware, operating system, and a full suite of applications in a fully integrated product. Unlike the computer industry, you cannot buy hardware from one vendor, an operating system from a second vendor, and applications from any number of third-party vendors.

In the computer industry, if someone has a new idea, as VMware did with virtualization, they can develop a solution and release it into the marketplace as an independent product because there are open interfaces to hardware and operating systems. In the networking industry, if someone has a new idea, they either have to introduce a standard to one of the standards bodies, which could take years to ratify and then implement, convince one of a small number of vendors to implement it as proprietary solution, or create a new company that builds an entire product (i.e. hardware, operating system and a complete set of applications). Vertical integration and lack of open APIs form a barrier to innovation, and have undoubtedly contributed to the mismatch between the pace of innovation in computing and the pace of innovation in networking.

c. Cost Savings Not Keeping Pace with Growth

Networking costs at universities have risen dramatically over past 5 to 10 years. There was the addition of WiFi to the network and now the need for pervasive, reliable, high-capacity WiFi. Security and compliance requirements have exploded. PCI-DSS compliance alone can require one or more full-time networking staff to manage. Big Data science, exemplified by the Large Hadron Collider, is driving the need for massive increases in both local-area and wide-area bandwidth and the rapid growth in over-the-top video, such as Netflix, is placing significant new bandwidth demands on campuses. Cost reductions in networking equipment are not keeping pace with the increasing demands thereby causing considerable stress on university budgets. Therefore, it is imperative that we find step functions to reduce networking costs for campuses.

d. Shortage of “Masters of Complexity”

Another major challenge facing universities is a lack of well-trained network engineers. Modern campus networks are incredibly complex and universities often resort to using technologies and products in novel ways, along with “home-grown” management software, in order to meet their needs. In this environment, it is critically important to have network engineers with a deep, fundamental understanding of networking concepts. It is not sufficient for engineers to understand how to configure equipment; they have to understand the underlying technologies and how all the technologies interact. Mastering this complexity is extremely difficult, and often requires many years of on-the-job experience. Solving this problem will likely require both simpler networks and improved education for future network engineers.

III. Software Defined Networking and OpenFlow

This section briefly describes Software Defined Networking and OpenFlow, how it enables innovation to address networking challenges such as those described above, gives more control of the network to infrastructure owners, and creates more choices and competition.

a. Moving Towards Solutions

It is relatively easy to establish that modern network technology is extremely complex and slow to evolve, relative to computing technology. These two key factors lead to the challenges described above. It only takes a quick look at the activities of the various standards organizations to see that there are thousands of people working on hundreds of new network protocols and that this has been the case for many years. Network engineers already have to master 15-20 or more different protocols to manage a network. How will adding more protocols to the mix reduce

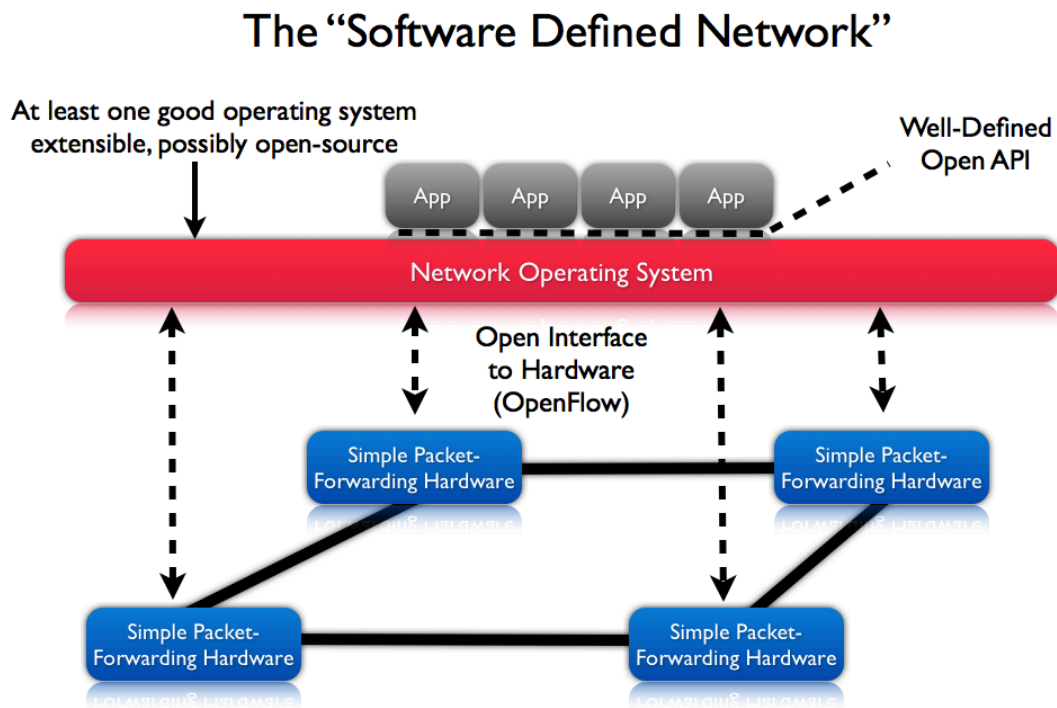
the complexity and cost of networking? Perhaps what we need is a more fundamental paradigm shift in how networks are built and managed.

b. Defining Software Defined Networking and OpenFlow

Software Defined Networking (SDN) is a term used to describe a new paradigm for networking designed to enable rapid innovation. A core concept of SDN is that by creating a few carefully chosen abstractions, a time-tested principle of computer science, you can hide much of the complexity of the underlying components and make it easier to develop and deploy new solutions.

Some of the key aspects of SDN include:

- Separation of forwarding and control planes
- A standard, vendor-agnostic interface to program network hardware
- Logically centralized control-plane or Network Operating System that constructs a logical map of the entire network and presents it via an API to application developers.
- Virtualization where multiple network operation systems can control “slices” of the same hardware device. This is a key differentiator of SDN and a focal point for many compelling use cases.



OpenFlow is an open API that provides a common instruction set for interacting with packet-forwarding hardware. OpenFlow is designed to allow the control-plane software to exist on a physically separate device from the packet-forwarding hardware. This makes it possible for a single control program to interact with many different switches. This control-plane software that controls multiple switches is commonly referred to as a “controller”, but is also called the “network operating system”.

One analogy for OpenFlow controllers are the controllers used in many wireless networks today. As with wireless controllers, a single controller interacts with many switches and there would typically be 2 or more OpenFlow controllers in a production network in order to provide redundancy and increase scale. However, unlike wireless controllers, OpenFlow controllers use an open API to interact with hardware and present a programmatic API that enables software developers to build new functionality as applications on top of the controller.

c. Key Abstractions Underlying SDN

As anyone who has programmed in both an assembly language and a high-level language like Python will tell you, abstractions are the key to dealing with complexity. This is why writing a program in Python does not require the programmer to understand all the details of the underlying hardware. Networking is complex because the appropriate abstractions have not yet been defined.

Software Defined Networking is based on the following set of abstractions:

1. **Forwarding Abstraction:** The networking industry does not yet have an equivalent to the x86 instruction set. Operating systems must be customized to each new chip technology from each vendor. OpenFlow is an open API that provides a common instruction set for interacting with any network chip technology. The term OpenFlow has become a buzzword recently, but in reality it is just one component of Software Defined Networking.
2. **State Distribution Abstraction:** Networking is fundamentally about distributing state (e.g. routing/switching tables) to every device in the network. Every group that develops a new protocol has to develop their own state distribution mechanism. Understanding the convergence properties of a single protocol is challenging, let alone the convergence of a network running many different protocols. The SDN approach is to create a single state distribution mechanism, with well-known convergence properties, that every new technology can utilize.
3. **Specification Abstraction:** One of the major challenges of managing networks is the fact that every device in the network has an individual configuration and there can be thousands or even tens of thousands of devices in a network. While there are tools today to help generate all those

configurations, the SDN approach is to provide an abstract view of the entire network so that network programmers can interact with the entire network as if it were essentially a single device.

d. SDN's Effect on Vertical Integration

SDN does not mandate a shift away from vertically integrated products. The SDN paradigm could be implemented with closed, proprietary software and APIs. However, OpenFlow, as the hardware abstraction for SDN, has been developed as an open, vendor-neutral interface. This makes it possible for a network operating system to interact with any switch that implements the OpenFlow standard and, like the x86 instruction set, this would allow vendors to compete independently in the hardware and network operating system markets.

IV. Use Cases

In this paper, we consider three key stakeholders who stand to benefit from the implementation of SDN in the University Campus environment:

1. **Campus Central IT:** The University central IT team that is responsible for network operations. They are typically looking to create new services for other organizations in the University and enhance their own operational efficiency
2. **Computer Science / Networking Research Community:** The community of researchers looking to leverage the programmability provided by an SDN architecture to enable advanced research using production (or near-production) network traffic and resources.
3. **Broad Science Research Community:** The community of researchers who frequently require specialized network considerations due to the sheer size of the data sets they use and the need to collaborate on these data sets with other research organizations (e.g. High Energy Physics Research, Bioinformatics / Genomics Research, Geological Research, etc).

SDN architectures across multiple vendors show three consistent themes that emerge in answer to the question “what can I do with SDN that is impossible (or at least impractical) with traditional networking technologies?” These three themes help organize the business case resources below.

1. **Virtualization:** Ability to create and modify virtual network topologies that run independently of the underlying physical/L2/L3 network design.
2. **Advanced Forwarding:** SDN architectures deal well with non-tree topologies, as they use forwarding techniques that do not rely on spanning tree

protocols. The result is both resiliency against common L2 failure modes and a native capability to service topologies like fat tree designs used in Big Data applications.

3. Programmability: Programming hooks that allow for vastly simplified integration between network control and other applications in the IT environment.

There are use cases for SDN in many different areas. This section will attempt to cover some of them.

a. Mobility, Scalability and Flexibility in the Data Center

(i) Distributed Data Center / Private Cloud

SDN enables flexibility in design and equipment choice for distributed data centers and the private cloud. In private cloud settings, SDN facilitates a shift away from use of static VLANs, which present a number of management challenges. Static VLANs are inflexible, not easily scalable, and their configuration is complicated and error prone. SDN makes it possible to use centralized management to define and control VLAN configuration in the private cloud.

<u>Private cloud with SDN/OpenFlow</u>	<u>Private Cloud with static VLANs</u>
Centralized management	Error prone, time consuming, hard to modify
Backwards compatible w/ VLANs	Not easily compatible w/ other systems.
Cost effective (CapEx – cheaper switches OpEx – lower maintenance and service contracts)	More expensive.
Easily Scalable	Difficult to scale

SDN allows for true scalability in the data center because the dynamic provisioning of resources is no longer hamstrung by the network. This facilitates true scalability of applications as compute, storage and network resources can be scaled dynamically to meet application requirements.

Another benefit of SDN in the Data Center is non spanning-tree forwarding. This is especially useful in high-performance computing (HPC) applications where full bandwidth or low-latency is required.

(ii) VM Migration

Research-oriented campuses often experience constant churn in terms of collaboration on research projects. This leads to dynamic requirements that are difficult to support with traditional networking, and are a perfect application of the on-demand virtualization/multi-tenant capabilities of SDN.

Virtualizing the network gives administrators the ability to easily migrate virtual machines between blade enclosures, racks, or even Data Centers. SDN abstracts away complexities of the physical network and presents administrators with flexible virtual networks that facilitate these migrations. This flexibility allows for SDN applications that provide a versatile replacement for costly, single-purpose commercial migration solutions.

(iii) Disaster Recovery

In the event of a disaster at a primary site, SDN makes it possible to easily bring up virtual machine instances in a disaster recovery location. Relocating a large number of users is a difficult and complicated task. Centralized management and abstraction of the physical network make it easier to quickly spin up workstations in a new location with all required access controls in place.

(iv) Virtualization

It is fairly common to find non-virtualized data centers that run at only 20% utilization. Virtualization makes it possible to achieve much higher utilization of system resources by allocating multiple virtual machines to servers and by allowing movement of VMs from servers with heavy workloads to those with lower utilization. Dynamic movement of VMs and applications across racks to servers with unused capacity is made much more feasible with SDN. SDN allows a 'network slice' to be provisioned and allocated to an application. This slice can then be moved with the application from one rack (or Data Center) to another with no disruption.

b. Network Virtualization / Compliance (PCI-DSS Example)

SDN gives network administrators the ability to centrally define policy to manage security on the network. It allows for isolated network slices with tight control capabilities that improve security. Access to certain network slices can be provided on a need-to-know basis by business unit, or even as granularly as individual based access control. An OpenFlow controller can also be configured to evaluate and drop certain flows, effectively acting as a logical firewall anywhere on the network.

An example of how SDN can greatly simplify and improve network security is found where point-of-sale (POS) terminals are installed on the network. In order to protect user data, networks with payment terminals are required to meet Payment Card Industry Data Security Standard (PCI DSS) compliance requirements.

Centralized network management as enabled by the SDN architecture greatly increases flexibility and opens up new possibilities for how these payment systems are managed. For example, one of the PCI DSS requirements is that each payment system should be on its own segment, separate from the rest of the network. If the MAC address of a POS terminal were known, it would not matter where on the network the terminal gets plugged in. The terminal would be identified by its MAC address and an isolated logical network could be dynamically created. This would greatly simplify network configuration when POS terminals are moved within the network. In the same way, unknown devices that plug into the network could be automatically isolated, reducing risk of data theft.

Campuses are often extremely heterogeneous in their user base. It is often the case that Campus Central IT has to manage systems as disparate as guest user wireless access, PCI compliance for payment, high performance computing, accounting applications, dorm rooms, faculty housing, etc., each with its own set of network requirements. The programmability and central management provided by OpenFlow and SDN would play a great role in simplifying management of these systems.

c. Dynamic Provisioning (GLIF/DYNES/ION/OSCARS)

There are currently a number of initiatives aimed at providing pooled network and compute resources to the academic and commercial community. Such initiatives include the Global Lambda Integration Facility (GLIF), Dynamic Network System (DYNES), Interoperable On-demand Network (ION), and On-demand Secure Circuits and Advance Reservation System (OSCARS). These systems are implemented in different ways on different underlying network infrastructure, but are generally available to the research community.

Access to these systems is currently provided through various resource allocation systems (e.g. Expedient and Opt-in manager, which are resource allocation interfaces written at Stanford). Although these resource allocation systems are open to the general research community, they are not necessarily designed to be compatible with the other resource pools.

Deployment of OpenFlow could be seen as a competitive research advantage that allows for dynamic and more efficient use of resources. OpenFlow's programmability allows it to facilitate interoperability between these systems. OpenFlow plug-ins for the different allocation systems would both greatly expand the total pool of resources available to research facilities and at the same time simplify the management of these resources. OpenFlow has that potential to become

a unifying protocol that will allow for the development of a common interface for scheduling and reservation across different systems, resulting in the truly dynamic allocation of resources in a flexible and scalable way.

V. Overview of SDN Status

In this section we outline the state of SDN standards and products. A more formal and lightweight process is being defined to standardize SDN interfaces and protocols with the participation of the largest network operators and a broad range of industry vendors. A few vendors have already announced products and we expect to see more product announcements during the coming year. In short the timing is right for the universities to plan and lead OpenFlow/SDN deployments keeping in mind that they would be early adopters of this technology.

a. Vendor Support / Product Availability

SDN enables an ecosystem with the potential for products and solutions at multiple layers in the SDN architecture. There are switches that support OpenFlow, network operating systems to control the switches, and applications that run on top of those operating systems. There will likely be both commercial and open-source alternatives at each layer as well as vendors that will bundle and support multiple components to create comprehensive solutions. With the success of SDN, it is reasonable to expect the networking marketplace to evolve to eventually resemble the modern PC marketplace.

At Interop 2011, 14 vendors demonstrated prototype switches supporting OpenFlow including HP, NEC, Dell, Netgear, Juniper, Brocade, Extreme, IBM/Blade, Pronto, Broadcom, Marvell and Fulcrum. Of these, NEC, Netgear and Pronto have “productized” versions of their OpenFlow-enabled switches that can be purchased through their normal sales channels and HP is offering support for their prototype firmware that supports OpenFlow. As more customers include OpenFlow in their purchasing requirements, we anticipate more of these companies will turn their prototypes into products. In particular, we are hopeful that the following vendors will release OpenFlow enabled switches in the near future: Brocade, Dell, Extreme, HP, Juniper, and IBM/Blade. Initially switch vendors are adding OpenFlow support to their existing hardware, but as they design new chipsets with SDN in mind, the capabilities and performance are likely to increase significantly.

There is also significant SDN activity in the area of network operating systems and applications. There are several open-source projects including NOX, Beacon, Open vSwitch, Maestro, Trema, and RouteFlow. NEC has recently released a commercial OpenFlow-based product called Programmable Flow that is targeted at data centers. At the 9th GENI Engineering Conference, Ericsson Labs demonstrated a prototype system that used OpenFlow to signal MPLS paths across a network. Additionally, a few start-ups have grown around OpenFlow. Notable among these are Big Switch

Networks and Nicira, whose founders include senior members of the Stanford research group that pioneered development of the OpenFlow technology.

b. Standards Process (ONF)

The Open Networking Foundation (ONF) is a nonprofit organization dedicated to promoting Software Defined Networking. The first task of the ONF is to standardize the OpenFlow protocol. However, the ONF also intends to define standards in other areas of the SDN architecture. It is these common standards with interoperable components at each layer of the architecture that will help create an open marketplace where users have a variety of choices at each layer.

ONF's Board of Directors brings together six major network operators, Deutsche Telekom, Facebook, Google, Microsoft, Verizon, and Yahoo!. Members of ONF include established companies (e.g. Google, HP, IBM) that seek to understand the potential Software Defined Networking presents in redefining their networks, as well as a number of start-ups that have built businesses around commercialization of OpenFlow through value-add applications (e.g. Big Switch and Nicira).

There is a clear and definite excitement in the industry about SDN and OpenFlow, and the possibilities that it presents. This buzz has continued to grow as more people begin to understand the technology and recognize ways in which this will enable them to better leverage their network. ONF has the opportunity to take advantage of this new technology and the support it has garnered to redefine networking as we know it.

c. IPv6 Support

The current OpenFlow specification does not include support for IPv6 although it is possible to create an extension to the specification for IPv6. It is broadly recognized that IPv6 support is an absolute requirement for OpenFlow and facilitating support for IPv6 is one of ONF's highest priority items.

VI. Strategy for Expansion of Campus Deployments

The process of building a program to deploy a cutting-edge networking technology such as SDN to a large number of university campuses is well understood. In the early days of Internet2, similar programs were created to encourage the deployment of IP Multicast and IPv6 on university campuses that were very successful. There are many similarities between those technologies and SDN so many of the lessons learned from those efforts that can be used to build an even more successful program to deploy SDN on many campuses.

a. Encourage the SDN/OpenFlow Marketplace

A core concept of the GENI project is that it will be possible to build a very large-scale networking test-bed because campuses will simply be able to turn-on the OpenFlow feature in their Ethernet switches. This highlights the perhaps obvious but important fact that the commercial success of OpenFlow is important to the GENI project.

At Interop 2011, 14 vendors demonstrated switches that support OpenFlow and 3 of those vendors have productized their OpenFlow implementations, which is extremely promising. This gives a strong indication of the level of interest in OpenFlow-enabled switches in the market. This is an ideal time to encourage the market by releasing additional RFPs for OpenFlow-enabled switches. Therefore, a successful program to deploy OpenFlow on campus networks should include competitive bids for a significant quantity of production OpenFlow-enabled switches – not prototypes.

b. Education Effort

Similar to Inter-Domain IP Multicast in the late 1990's, there is an amazing buzz around SDN and OpenFlow. During the last 3 months alone, Network World has published 7 articles on SDN & OpenFlow and the number of blog posts, podcasts, and tweets has been incredible. But while many people have heard about SDN & OpenFlow and are excited by the concept, relatively few people really understand it and some people may misunderstand it.

Therefore, a successful program to expand OpenFlow deployments to more campuses should include a plan to provide training. For IP Multicast and IPv6, a series of hands-on workshops targeted at network engineers has been very successful. The focus of the workshops is to train campus network engineers so they both understand the use cases for using the technologies and have the knowledge they need to start deploying the technology on their campus. The workshops are typically hosted at campuses around the country to encourage local participation and the instructors are network engineers who have already deployed the technology on their own campuses, therefore they can share their real-world experience.

In the case of SDN & OpenFlow, there may need to be multiple workshops. There may need to be some focused towards network engineers deploying the technology and other focused to towards developers. The developer workshop would be useful for both developers in the IT department and the faculty and students in the computer science department.

c. Ongoing Support

One important lesson learned from previous experience is that after network engineers participate in a workshop and go back to their campus to start working with the technology, they typically come up with lots of questions. A hands-on workshop is a great first step, but ongoing support is needed after the workshop.

There are a number of possible components to provide ongoing support that have proven successful in the past:

1. **The Buddy System:** 8 campuses already have experience with SDN. If each campus helped to support 2 additional campuses, the number of SDN/OpenFlow campuses would be scaled to 24 without adding significant central support resources.
2. **Coordination:** The buddy system is a useful mechanism, but the effort must be well coordinated. Tools such as a central information repository (e.g. knowledge-base or wiki), forums, and mailing lists could all be useful in coordinating the effort and disseminating knowledge. Also, it's important to have leadership to provide overall direction for the project.
3. **Measurement/Reporting:** It's important to quantify the benefits of the project at each campus and to report them to the rest of university community in order to encourage interest at additional campuses.

d. Funding

The initial NSF GENI funding to 8 campuses proved to be an extremely worthwhile investment to bootstrap SDN activities on those campuses. Once the central IT organizations saw the incredible potential SDN has to transform the way they manage their networks, they started investing their own resources to expand and increase the pace of the project. For example, some campuses have already created internship programs and other initiatives to build collaborations between their IT staff, network researchers and students around SDN because they have seen first-hand how this not only benefits the IT department, but also the research and academic missions of the university. This funding model can serve as a useful template for the expansion to additional campuses.

VII. Summary

Universities are facing unprecedented challenges in funding and managing networks that support the rapid pace of innovation in computing technologies. Their networks are becoming increasingly complex and costly while not meeting the needs of their users. Software Defined Networking offers a paradigm shift, based on

the time-tested principle of using abstractions to address complexity, that has the potential to revolutionize networking and solve many of these challenges.

The NSF has shown incredible vision in funding 8 campuses to spearhead the deployment of SDN on university campuses. There are numerous examples as to the success of this project, but perhaps the most telling is that several of these campuses are now investing their own resources to bolster their SDN efforts, and they plan to include OpenFlow as a “must have” requirement in upcoming RFPs for switches. This clearly shows that these campuses understand the impact that SDN holds for their campuses and are committed to its success.

Now is the ideal time to invest in taking the next step forward. There are universities ready to champion the idea with their colleagues and to help them get started. The market for commercial SDN products is growing rapidly and the university community has the experience to draw from in building a successful program to expand SDN deployments to many more campuses.

The NSF's seed funding has demonstrated the potential of SDN and now, as it is ready to take off, it is important that the NSF continue the funding to scale deployment of OpenFlow/SDN on the original 8 campuses and also grow the number of new campuses with OpenFlow/SDN deployments. This will provide state-of-the-art networking infrastructure to universities and they would again become a place where many innovations happen to define computing and networking.

APPENDIX

i) Data Center Networking Costs

Data Center Size	100,000 square feet
Cost per square foot	\$2000
Total D.C. Cost	\$200,000,000
Number of Servers	30,000
Number of Switches	3500 (3000 ToR, 500 EoR)
Useful Life of Network Equipment	5 years
Network Infrastructure cost	15% of total D.C. cost (=\$30M amortized over 5 years)
Cost of network service contracts (over useful life of 5 years)	50% of total network equipment cost (=\$15M over 5 years)
Network Administrator Salary	\$50/hour
Network Administration Cost	0.5-1hr/switch/week (= 26-52hrs/switch/year = \$1300-\$2600/switch/year)

ii) Potential Savings with SDN and OpenFlow Controller Software

Capital Expenditure

Network Infrastructure approximately 15% of total Data Center costs.

- Typical build out cost: \$2000 per square foot.
- Typical Size: 100,000 square feet.
- Total Cost: \$200M.
- Networking Cost: \$30M.

SDN would reduce required hardware by half.

- Savings of approx. \$15M.

Operational Expenditure

Better charge-back visibility.

- Allow groups to pay for actual usage.

Human Resources - redeploy to highest and best use.

- 0.5-1hr/switch/week spent on manual configuration and maintenance.
- 52 hrs/switch/year X \$50/hr = \$2600/switch/year.
- SDN reduces manual configuration time by >50% (\$1300/switch p.a. savings).

Service Contracts

- Current total service contract cost. (~50% of total CapEx over product life).
- Expected reduction with SDN. (75% total service contract cost reduction).

Registration costs.

- Fewer hardware and software installs.
- Expected reduction in registration cost with SDN.