

Software-Defined Networking Overview and Implementation

Manal Algarni, Vinayak Nair, David Martin, Sayali Shirgaonkar
George Mason University
<malgarni, bnair, dmarti22, sshirgao>@gmu.edu

Abstract

This paper presents research in the area of software-defined networks. The goal of our research is to identify how software-defined networks (SDN) could reduce cost, and improve efficiency to provide significant business value to many enterprises and industries. These outcomes can lead to increased market potential for competitors which identify favorable solutions to existing networking constraints. We seek to identify solutions in this study to enable potential customers to assess that SDN may be preferable over existing IP networks.

I. Introduction

Software-Defined Networks have expanded beyond being a research topic at university campuses to a potentially new approach in designing, building and operating enterprise data networks. In this paper we explore the mechanics of software-defined networks, how they compare with the existing network architecture currently in place and the benefits of the new model. We also explain the details of control and forwarding plane and how SDN protocols such as OpenFlow decouple the two allowing more flexibility and granular control for network management. This is followed with diverse SDN deployments and how they compare with one another.

We will then determine the potential market opportunities and business value behind software-defined networks (SDN), which may enable significant network improvements, resulting in reduced costs and increased data transfer efficiency as compared to today's environment. A problem in today's environment is that businesses' which require significant bandwidth can improve their network efficiency by improved data routing based on control planes which they themselves can control and revise as opposed to reliance on IP protocols implemented in standard carrier networks.

II. Research Problem

Networks today are experiencing much larger consumption of data via an array of devices. With a continuous influx of data, and an increased demand for more resources, bandwidth, and accounts, an information overload is always a big concern in the networking industry. The ability to reduce complexity by automation is required to cope with real time changes at the application and user level, which the existing IP protocol network infrastructure is unable to provide. The current model is also not the most efficient use of resources, time, or capital to address tailored a specific business' needs. Rather IP Networks provide a general pool of resources setup with a predefined structure and limited configurations that confine users, which also leads to poor network efficiency.

In order to align to each business and optimize business process, redundant network resources need to be removed from the business process. A software-defined network can provide singular control over an overwhelming data load and multifaceted network infrastructure. Software-defined networks are a strategic user-friendly approach to resolving this problem and give customers more opportunity to craft their data streams today. There is significant growth opportunity and value in the software infrastructure model; however the networking community has not picked up the SDN solution as quickly as it should, considering the value this model could bring to both its providers and its users. The new software infrastructure model is reported as being more efficient while less likely to experience technical difficulties. Awareness of software-defined networking and its growth potential in the networking community is vital to pushing the optimized process to market.

Along with defining the elements, mechanics, deployment models, protocols, and advantages of software-defined networks in comparison to IP network infrastructure, we seek to define the market opportunities for implementation within a model large enterprise as well as a small business environment in today's operating environment. We are aware that software-defined networks are becoming available in the marketplace however our research project will enable us to identify cost and

productivity improvements within varying size businesses which would make this project and the associated implementation valuable to the business.

III. Internet Protocol vs. Software-defined Network

There is constant debate over which style of network is better; software-defined networking or Internet protocol (IP) networking. While they both have their advantages and disadvantages, overall we have found the software-defined network to be preferable. Key attributes of an SDN environment include its user friendliness, cost efficiency, and reduced complexity. However, while it is predominately better than IP networking, there are some cases in which IP networking can be more advantageous.

Software-defined networking can be directly related to simplicity, adaptability, and scalability in any network environment (Costanzo). IP networks are unable to match these qualities and because of this an increasing amount of Internet providers and business are beginning to rely more on SDNs. Not only is this style adaptable but it is also user friendly to systems administrators. Unlike IP networks, systems administrators no longer have to flip switches, go through manual configuration policies, or have direct access to the hardware. Instead, SDN systems administrators are able to have central programmable control over network traffic without the need to have direct access to the hardware (Costanzo).

Furthermore, SDNs provide singular control over the network infrastructure and in doing so reduce complexity of processes through automation (Costanzo). This is beneficial to companies who must be able to manage real-time changes at not only the application level but also the user level. Systems administrators at any point can make these essential changes in time regardless of their location. Remote access and changes to the network are made possible through the implementation of a role-based access system; this system is able to provide the security to keep hackers and other attackers from accessing the business's network (Koldhofe et al). Unfortunately, these on the fly and remote changes are not possible through the use of internet-protocol networks. In the IP networks systems administrators must have direct access to the control panel and go through manual configuration policies in order to make any changes. Any network policy change requires making hardware changes which makes the system rigid.

SDN allows for unlimited policies and change to those policies for intrusion detection, firewalls and load balancing with changes to software, which makes managing networks much more flexible.

Another way in which software-defined networking outmatches and outperforms internet protocol networks is the fact that it allows administrators to indicate network services without conglomerating interfaces and specifications together (Costanzo). Not only does it allow administrators to choose specific services, it also permits them to control the two planes. Software-defined networking is able to separate the control plane and data plane. A detailed description of the control plane and data plane/forwarding plane and how they interact is provided in Section V and VI.

By separating these two planes, this allows the administrator to make decisions in regards to the path of data (Koldehofe et al). In decoupling the two entities many claim that networking is simplified, faster, less likely to be overloaded, and more user friendly (Koldehofe et al). Internet protocol networks are unable to do this due to the fact that the two planes are concreted and solidified into an almost single entity. IP networks cannot decouple them and therefore cannot allow the administrator to control the planes. This can result in an information overflow and network failure.

Software-defined networking is recognized for how advanced and user friendly it is but an advantage few seem to realize is how efficient it is and how less likely it is to experience technical difficulties. Due to the ability of systems administrators to interact directly with the software, they can make changes to data flow passages, which ensures that data packets do not get queued and degrade network performance. By ensuring the data does not block the pathways or overload them in anyway, it is less likely that the networks will malfunction or experience technical difficulties. Another key advantage to software-defined networking is the cost of it. It is cheaper than internet-protocol networks because it does not require as many people working on it (Costanzo). Companies could potentially cut out most of their system engineer costs and only have to rely on a few systems administrators rather than a whole squad of them.

There are many advantages to software-defined network in comparison to internet-protocol networks. Unfortunately for companies there are some drawbacks to using software-defined networks that are not present within internet-protocol networks. While it is advantageous for the system-administrator

to have remote access and control over the software-defined network, it raises a few security issues that are combated by internet-protocol networks.

The first main security issue is remote access, this means that regardless of firewalls put into place if the system is hacked anyone can have access to the settings and change them from anywhere at any time. They would also be able to access any secured files through the network. IP networks do not allow this because in order to access the network you must have access to the hardware itself (Koldehofe et al). Most companies only allow few individuals access to the hardware so their systems is more secure and less likely to be invaded by exterior forces.

An additional benefit of internet-protocol based networking is the availability of multiple layers. These layers are not able to be manipulated and are imbedded within the network devices (Costanzo). This leaves little room for malfunction in data flow due to operator error. While the software-driven network style is beneficial and allows for manipulation of data flow, this could potentially cause malfunctions within the network. This could greatly damage the company's ability to send data between hosts as well as conduct business.

While there are strict implementations for the protocol for the use of hardware and software in software-driven networks there are none present within internet-protocol networks (Costanzo). Due to this fact there is a multitude of different ways to use the software as well as the hardware in the Internet protocol networking system. Even though there is a multitude of ways to use hardware and software in the Internet protocol networking system, most computers today use transmission control protocol. Transmission control protocol is used mainly to access the internet, control the internet content that is seen by the user, seamlessly deliver email, and lastly send data from one place to another without any discrepancies (Koldehofe et al).

There is a multitude of benefits from the implementation of both networking systems; however, it is crucial one understands advantages brought forth by software-driven networking. This technology is not only ground breaking, it is also cost efficient and user friendly. It allows access to the network by authorized individuals and allows real-time changes to be made. These changes can keep the network from experiencing technical difficulties. Furthermore, this technology is cost efficient because it cuts out the necessity of a team of systems administrators and allows singular control over an

entire network, which is something Internet protocol networking is still unable to achieve.

IV. Network Planes and Virtualization

Networks contain a layered architecture, which play a vital role in transferring IP packets from source to destination, also making them fundamental within SDN environments. This layered architecture consists of a control plane, forwarding plane or data plane, and management plane (see Figure 1).

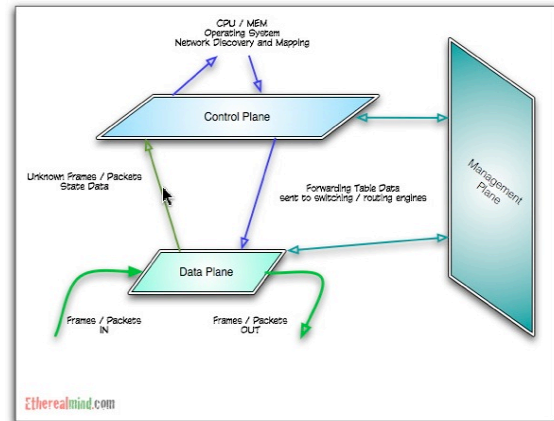


Figure 1: Control, Forwarding (Data), and Management Plane

The control plane essentially controls how routers interact with other hosts; it takes into account system configuration, management, and exchange of routing information contained in an Information Base (see Figure 2) and Label Information Base. These databases will contain tables of various routing scenarios based on the router's vendor's priorities and preferences, and update forwarding tables, essentially defining and redefining the router's topological outlook.

Usage	Family	Junos Name
Unicast routing	IPv4	inet.0
IPv6 Next-hop resolution	IPv4	inet.3
IPv6 Multicast routing	IPv4	inet.1
IPv4 Multicast RPF	IPv4	inet.2
IPv4 MPLS labels MPLS	MPLS	mpls.0
inet-vpn	INET-VPN	bgp.i3vpn.0
I2vpn L2VPN	L2vpn	bgp.i2vpn.0
IPv4 Unicast routing	IPv6	inet6.0
IPv4 Next-hop resolution	IPv6	inet6.3
CLNS routing ISO	ISO	iso.0

Figure 2: Juniper's Routing Information Base tables. Juniper is a networking gear provider that will be adopting the SDN strategy and model.

Since forwarding data updates slowly in large networks, control planes are considered the legacy path over packet switching architectures.

The forwarding plane can be also called the data plane. The data plane parses packets headers, manages encapsulations, queuing, and policing; it deals with user traffic. When packets are destined to or originating from a router, they do not go through its data plane, only its control plane. Only when packets are being sent through an intermediary router is when the intermediary's data plane is in use. The management plane deals with administrative traffic in order to manage network traffic.

Network virtualization is the concept of carving several logical paths out of a physical network and multiplexing infrastructure. This would entail direction over packet manipulation and forwarding in the data plane by a network virtualization supervisor, who would produce one or more logical forwarding elements. The control plane would then use this to explain the required network functionality. However, network virtualization by a network supervisor would allow mainly static configuration. To have constant changes in configuration done through the logical interface would require network management software. This is where software define networking comes in.

V. Software-defined Network Controller

The SDN controller is the interface between the application layer and the network devices. The control plane is removed from the switch and is now contained in the SDN controller. The SDN controller can be now programmed for making routing decisions, instead of having those algorithms built into the switch. It allows control and enablement of intelligent networking platforms to operate a variety of technology components.

The SDN controller will then relay the decision to all the devices in the network, based on a communications protocol such as OpenFlow (see Section VI). Technology components are set with certain protocols to synergize with OpenFlow in order to allow services to transfer data to switches and secure any designated packets within the network. OpenFlow updates the flow table in the switches, which is used by the network device to direct data packets. This allows the SDN Controller to manage flow control in the network and choose the optimal path depending on network conditions.

VI. OpenFlow Protocol

OpenFlow is a communications protocol used in Software Defined Networking (SDN) that decouples the control plane and data plane. This allows for control of a network's layout and traffic flow from a singular point. The control plane refers to the capabilities of the Routing Engine. The creation of routing and forwarding tables, maintenance of adjacencies, filtering, policies, and system monitoring are handled by the control plane. In contrast, the forwarding plane of the router consists of the interfaces, the Packet Forwarding Engines, and the switch fabric (see Figure 3).

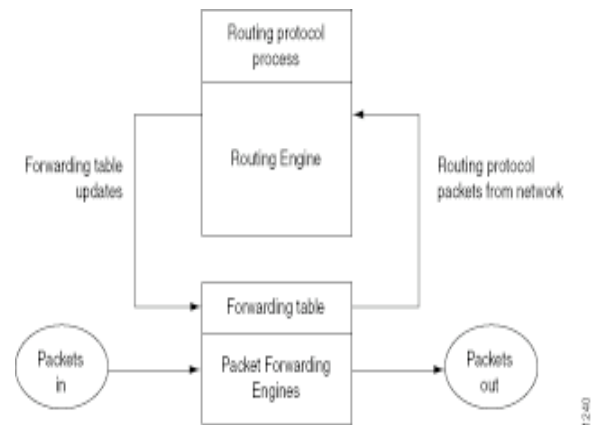


Figure 3: Forwarding plane of a Router

In the current architecture the control plane populates the forwarding table, which is used by the forwarding plane/data plane to forward packets to their next destination. This architecture is rigid since it implies that all data flow between two end hosts will follow the same path even if their requirements are different say one is a video data packet versus regular page content.

OpenFlow is a protocol to program the flow table in different switches and routers. This allows the next destination of data packets to be determined by the program as opposed to the control plane deciding. This separation of the data plane and control plane allows a program to define the network path provided the software is installed on multiple routers or switches. The network devices now simply can work of a single set of SDN instructions as opposed to thousands of protocol standards.

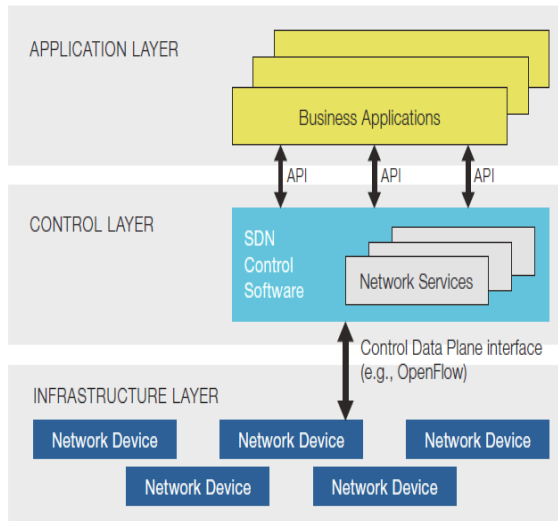


Figure 4: OpenFlow on the Control Layer

The Open Flow software is installed on both the control layer (i.e. the control software) and the network devices (*see Figure 4*). Open Flow allows identification of network traffic based on pre-defined match rules and allows for control based on parameters such as usage patterns allowing for it to respond to real time changes at the application user and session levels.

An entry in the flow table has three fields: (1) A packet header that defines the flow, (2) The action, which defines how the packets should be processed and (3) Statistics, which keep track of the number of packets and bytes for each flow. Three basic actions that can be performed on a packet are:

- (1) Forward this flow's packets to a given port
- (2) Encapsulate and forward this flow's packets to a controller
- (3) Drop this flow's packets, use for security (*see Figure 5*)

This can be compared to an instruction set given to a CPU. An example of how a Flow Table would look is given below:

FIGURE 2
Example of OpenFlow
Instruction Set

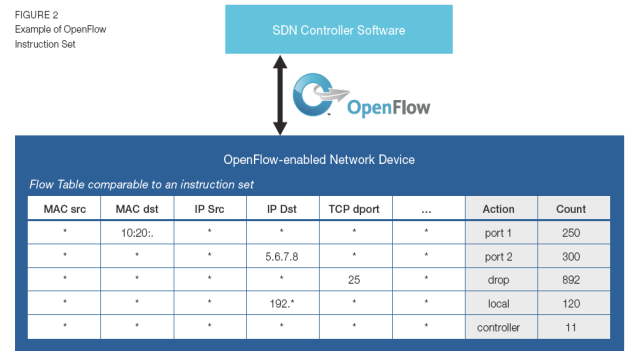


Figure 5: Example of Open Flow Instruction Set

While OpenFlow was initially tested on a campus Ethernet network, it can now be deployed on both physical and virtual networks. Network devices can also support the traditional forwarding from the forwarding table as well as SDN defined forwarding with OpenFlow protocol allowing for a gradual upgrade to SDN technology across multiple vendors.

VII. SDN Deployment Models

SDN utilize policies implemented in an SDN controller to provide the services and applications to the data plane for data delivery. The typical SDN architecture is illustrated in the figure below:

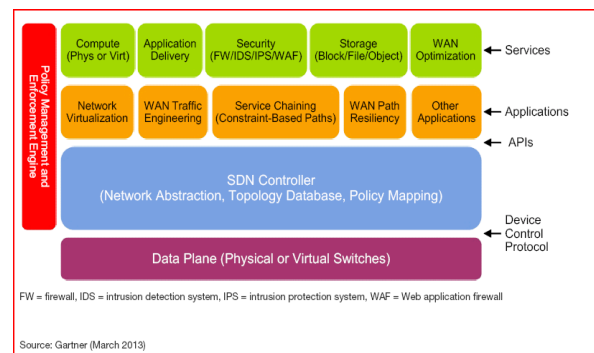


Figure 6: SDN Architecture

Currently there are three predominant approaches to deploying a SDN, switch based, overlay, and a combination of the two, which referred to as a hybrid deployment strategy. These deployments involve the methods of data control and configuration below the SDN Controller (example Open Flow) level as illustrate above.

- 1) **Switch Based:** In this model, the SDN control protocols are issued directly from the SDN controller (virtual machine) directly to the data control plane within SDN enabled

switches and network equipment (*see Figure 7*).

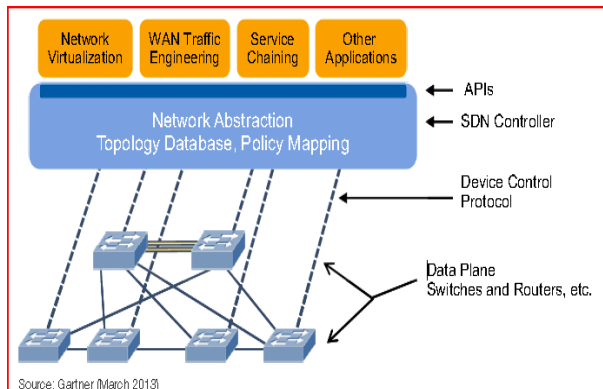


Figure 7: Switch-Based SDN

When a packet arrives at a switch in a conventional network, rules built into the switch's proprietary firmware tell the switch where to forward the packet. The switch sends every packet going to the same destination along the same path -- and treats all the packets the exact same way. In the enterprise, smart switches designed with application-specific integrated circuits or "ASIC" are sophisticated enough to recognize different types of packets and treat them differently. These ASIC enabled switches are more expensive than existing commodity IP network switches.

In a software-defined network, a network administrator can manage traffic from a centralized control console without having to touch individual switches. The administrator can change any network switch's rules when necessary -- prioritizing, de-prioritizing or even blocking specific types of packets with a very granular level of control. This is especially helpful in cloud architecture because it allows the administrator to manage traffic loads in a flexible and more efficient manner. Essentially, this allows the administrator to use less expensive, commodity switches and have more control over network traffic flow than ever before.

SDN allows network engineers to support a switching fabric across multi-vendor hardware and application-specific integrated circuits. Currently, the most popular specification for creating a software-defined network is an open standard called OpenFlow. OpenFlow lets network administrators remotely control routing tables.

The biggest limitation to this approach is that is currently does not leverage existing L2/3 network equipment.

2) Overlay Network: This deployment approach can be used to accelerate deployments in enterprises with an existing IP network using a tunnel based overlay approach which can be implemented by a server virtualization team. The data source and end host maintain virtual devices that are part a "hypervisor" environment. In this model, the SDN control protocols are issued directly from the SDN controller (virtual machine) directly to the SDN hypervisor switches that are in control of an enterprise's existing IP network equipment. Special SDN switches are not required to implement software-defined networks using the overlay model (*see Figure 8*).

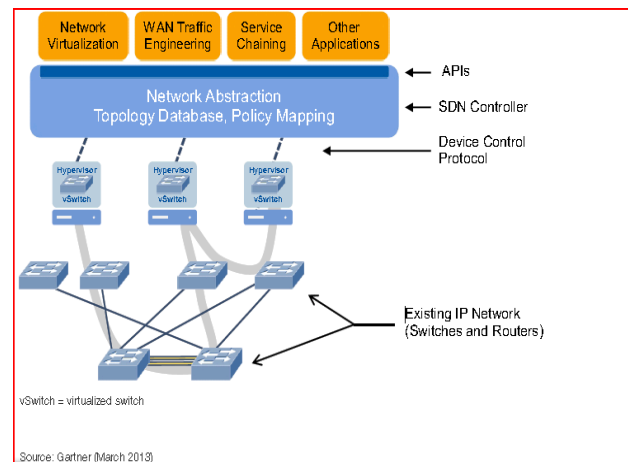


Figure 8: Overlay Network SDN

The overlay model requires the use of Hypervisor virtual switch equipment that is responsible for providing instructions to the existing IP network that run under the virtual switch. The virtual switch is a virtual machine responsible for performing the network edge responsibilities, which would interface with the SDN network applications. Overlay model virtual switches have two primary responsibilities to include layer two network delivery functions via a "virtual Ethernet module" and adherence to supervisory policy instructions.

Function of the Virtual Ethernet Modules - The Virtual Ethernet module provides configuration information and Layer 2 switching and advanced networking functions such as configuration for port channels, quality of service, security to include port, (VLAN) and access control. Additionally in the

event of loss of communication with the virtual switch, the VEM has Nonstop Forwarding (NSF) capability to continue to switch traffic based on the last known configuration. Thus, the VEM provides advanced switching with data center reliability for the server virtualization environment.

Function of the Virtual Supervisor Modules -
The virtual switch supervisory module controls multiple VEMs as one logical modular switch. Instead of physical line-card modules, the VSM supports multiple VEMs running in software inside the physical servers. Configuration is performed through the VSM and is automatically propagated to the VEMs. Instead of configuring soft switches inside the hypervisor on a host-by-host basis, administrators can define configurations for immediate use on all VEMs being managed by the VSM from a single interface. The virtual supervisor provides port configuration via software, system failover instructions to increase availability, and can be implemented and managed via existing protocols such as SNMP, API and command line interfaces.

This approach has the disadvantage such that the network team will be required to maintain both the historical network equipment and the task of debugging routing issues needs to evaluate both the SDN and the historical network to resolve issues.

- 3) Hybrid: This deployment is a combination of the switch based and the overlay (tunneling approach), which can be used to gradually migrate existing equipment to a new switch based model. This allows for an enterprise to control the speed of its SDN deployment and control the rate of equipment investment. One potential disadvantage of this approach is that certain gateway links may not always support the tunneling methods employed (see Figure 9).

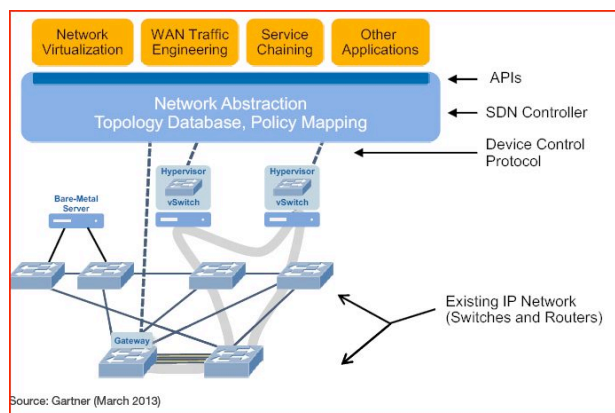


Figure 9: Overlay Network SDN

VII. Related Research Works

Hardware and software providers are considering software-defined networks already. Vendors such as Cisco, Juniper, Big Switch, and others are manufacturing hardware to support SDNs. Currently software protocols, such as OF and others, are being developed and refined to enable incremental SDN capability to be deployed. We expect broad based adoption of SDNs in a wide variety of business over the next decade.

VIII. Solutions and Analysis

IP vs. SDN Cost Analysis

We have compared the cost of existing IP network equipment with that of the SDN network devices. We found the cost of mature market (i.e. commodity based) IP network gear to be substantially cheaper at the current time due to the relative immaturity of the SDN equipment design life cycle. We noted that all major network equipment manufacturers to include Juniper, Cisco, NEC, Arista Networks, Brocade, Big Switch, HP, and IBM are designing equipment to be used to support various SDN protocols and network operations. This competition will drive down the SDN equipment costs over time.

Further we noted that the virtualization encourages potential reduction in other standalone network hardware such as firewalls, spam filters, and intrusion detection hardware, as these functions can be performed by a properly configured SDN through software. The SDN controller pricing from IBM illustrated below is applied to the first and second software licenses granted (note: most networks require redundancy in design hence two licenses would be required) with all additional incremental licenses to be priced at \$1,700 each. The chart in **Appendix A** illustrates the current disparity in pricing between IP and SDN network gear based on commercially available retail sources and excludes any consideration of vendor discounts.

Business Case Assessments

As mentioned above, any network policy change requires making hardware changes which makes the system rigid. SDN allows for unlimited policies and change to those policies for intrusion detection, firewalls and load balancing with changes to software, which makes managing networks much more flexible.

1. SDNs can be implemented within large-scale data centers environments such as public cloud providers.

These are customers that are hyper scale public clouds vendors and can include Amazon.com, Google, Facebook, MSN, Yahoo, Badu, IBM, AT&T, Verizon, and Rackspace.

Business Needs

The operations of certain large-scale public cloud providers currently are centered on content delivery network model and website hosting. These enterprises have data demand, availability, and latency requirements. Scaled online retailer providers such as Amazon.com as well as enterprise business data storage networks are prime candidates for SDN implementations. These public cloud providers must contend with service and server mobility which requires a per device configuration approach within an IP based network model. The SDN can provide customizable equipment provisioning and configuration that can be automated and centralized to flexibly enable existing assets to be configured by policy to respond to peak (synaptic) data flow that accompany the business cycles such as holiday sales seasonality, data consumption patterns that do not occur ratably.

1.1 SDN Benefits

- An SDN can direct traffic through the network from an originating point to a terminating point based on real-time status of all network elements and policies defined for each endpoint.
- A single high-level program (API) can be used to separate and control the data plane as well as the control plane via the network controller device as desired by the requirements of the underlying business.
- The SDN provide a means of addressing hyper scale growth in the utilization and scalability of their data center network equipment.
- An SDN can be reconfigured faster than current network architectures to respond to new business needs.
- With enterprises of this scale the ability to manage data based on internal service policies promotes the ability to create competitive advantages over networks that rely on IP protocols only.
- These cloud providers have significant operating costs from data transport, storage,

and network administration. Each of these cost drivers can be reduced by the use of SDN as virtual machines replace manual device-by-device configuration formerly performed by a large team of network administration personnel.

- The ability to route and manage data over preferred network channels and slot information can enable the business to flatten bandwidth consumption to reduce the quantity and costs of dedicated circuit from global network providers.
- OpenFlow controllers enable administrators to set policy to drop packets which increase network security and vulnerabilities from distributed denial of service attacks.
- A single high-level program (API) can be used to separate and control the data plane as well as the control plane via the network controller device as desired by the requirements of the underlying business.

2. SDNs can be implemented within small businesses and campus environments.

These are customers that are local campus network environments such as George Mason University “GMU” as well as small business with multiple distributed office locations over a wide geographic area that require a higher level of network security.

2.1 SDN Benefits

- SDNs enable enterprise security in businesses with high throughput yet low latency requirements by domain isolation within a single data center.
- SDNs allow for central control plane instructions over multiple end-devices, which can strengthen network firewall capabilities.
- The SDN controller can be configured to act as a proxy on behalf of applications to prevent and control network device accesses
- SDNs may utilize “service chaining” as a way of inserting services into the flow of network traffic as it moves among network devices.
- SDNs can implement virtual LAN “VLAN” instructions and provide access control lists (ACLs) as a means of enforcing network security.
- Campuses can benefit from the ability to consolidate many network equipment types

onto industry-standard servers, switches and storage.

- Distributed offices will benefit from the SDN automates equipment configuration in common events such as install/adds/moves/changes/ ("IMACs") and client device roaming.

3. SDN Common Consideration Points (Large and Small Implementations)

- Depending on the implementation model (switch vs. overlay), new IT hardware to perform the layer 2/3 routing will be required for switch base deployments. The equipment costs (i.e. new switches or virtual switch/virtual machines) are more expensive than traditional IP switches that have been on the market for decades.
- SDNs as implement via virtual machines may be perceived as a threat to the Company's existing network engineering team as the implementation of the SDN may reduce the headcount requirements of the business' network management team.

VIII. Summary and Future Work

We have identified the advantages and capabilities of SDNs vs. IP models, network planes, protocols, SDN deployment models and resulting application to today's business environment. We believe there is significant growth opportunity in the market.

References


"48 Port Gigabit Router." *Google Shopping*. Google. Web. 21 Apr. 2013.

Amazon. "iwNetworks." *Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & More*. Amazon. Web. 21 Apr. 2013.

Cisco. "Products & Services." *Cisco*. Cisco, Web. 07 Apr. 2013.

<http://www.cisco.com/en/US/prod/collateral/iosswrel/content/white_paper_c11-707978.html>.

Costanzo, Salvatore, et al. "Software Defined Wireless Networks (SDWN): Unbridling SDNs." (2012).

 Cost Central. "Switches." *Networking*. Cost Central. Web. 21 Apr. 2013.

Ending the Confusion About Software-Defined Networking: A Taxonomy Published: 12 March 2013 By Joe Skorupa, Mark Fabbi, Akshay K. Sharma. Gartner Research.

"EtherealMind." Web log post. *EtherealMind*. Web. 07 Apr. 2013.

Gartner Research. Joe Skorupa, Mark Fabbi, Akshay K. Sharma – "Ending the Confusion About Software-Defined Networking: A Taxonomy Published:" 12 March 2013

Greene, Kate. "TR10: Software Defined Networking." *MIT Technology Review*. MIT Technology Review, Apr. 2009. Web. 07 Apr. 2013.

Google Shopping. "D-Link Data Center 10GbE Top-of-Rack Switch DXS-3600 Switch - 24 Ports." *Google Shopping*. Google. Web.

"Electronics: Network Switch." *Amazon.com*. Amazon. Web. 21 Apr. 2013.

Juniper. "Technical Documentation." *Juniper Networks*. Juniper Networks. Web. 07 Apr. 2013.

Koldehofe, Boris, Frank Durr, Muhammad Adnan Tariq, and Kurt Rothermel. "The Power of Software-defined Networking: Line-rate Content-based Routing Using OpenFlow." *Artikel in Tagungsband INPROC-2012-41*. University of Stuttgart, 2012. Web. 29 Mar. 2013.

McKeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. "OpenFlow." *ACM SIGCOMM Computer Communication Review* 38.2 (2008): 69. Print.

Mullins, Robert J. "Brocade Adds to Router, Switch Lines to Advance SDN, Fabric Networking." *Enterprise Networking*. EWeek. Web. 21 Apr. 2013.

Nicira, Martin. *Virtualizing the Network Forwarding Plane*. Stanford. Web. 7 Apr. 2013.

Open Networking Foundation. "Software-Defined Networking: The New Norm for Networks." Open Networking Foundation, 13 Apr. 2012. Web.

Rouse, Margaret. "Software-defined Networking (SDN)." Tech Target. Web. 07 Apr. 2013. <<http://searchsdn.techtarget.com/definition/software-defined-networking-SDN>>

Rubens, Paul. "Using Floodlight to Explain SDN Controllers and OpenFlow." *Software Defined Networking Controllers Explained*. Enterprise Networking Planet, 14 Jan. 2013. Web. 21 Apr. 2013.

Salisbury, Brent. "The Control Plane, Data Plane and Forwarding Plane in Networks." Web log post. NetworkStatic Brent Salisburys Blog. Web. 07 Apr. 2013.

"Switches and Bridges." *Netgear, Inc - Netgear ProSafe 24-Port, 10 Gigabit Stackable L2+ Managed Switch*. Computech International. Web.

"Supermicro 1/10-Gigabit Ethernet Switching Module." *Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & More*. Amazon. Web. 21 Apr. 2013.

"Telecommunications Network." *Wikipedia*. Wikimedia Foundation, 04 June 2013. Web. 07 Apr. 2013.

APPENDIX A

ILLUSTRATIVE TABLE OF COMPARATIVE SDN /IP NETWORK EQUIPMENT PRICES

Network Type	Price	Ref	Vendor	Product / Part #
DEVICE: NETWORK CONTROLLER				
SDN	\$ 92,000	A	IBM	Programmable Network Controller software license
IP	\$ -		N/A	Standard IP Network does not require a SDN Controller
DEVICE: TOP OF RACK SWITCH				
SDN	\$ 33,835	B	Brocade Communications	BR-MLXE-8-MR2-M-AC - Mlxe-8 Ac Sys W/ 1mr2 M Mgmt Mod 2high S
SDN	\$ 15,000	C	SDN	SDN 8952S Series 10Gb Top-of-rack Network Switch
IP	\$ 7,162	D	D-LINK	D-Link Data Center 10GbE Top-of-Rack Switch DXS-3600 Switch - 24 ports - managed - stackable
IP	\$ 4,679	E	Netgear	Netgear, Inc - Netgear ProSafe 24-Port, 10 Gigabit Stackable L2+ Managed Switch Part Number #: XSM7224-100NES CTI #: 1023877618
DEVICE: SWITCHES				
SDN	\$ 3,300	F	SDN	SDN 8254S series 1G/10G Enterprise-class Ethernet Switch (Redundant PSU)
IP	\$ 1,853	G	SuperMicro Computer	Supermicro 1/10-Gigabit Ethernet Switching Module Part Number #: SBM-GEM-X2C CTI #: 1013032657
IP	\$ 155	H	DLINK	D-Link 24-Port Rackmountable Gigabit Switch DGS-1024D
DEVICE: ROUTERS				
SDN	\$65,000	I	Brocade Communications	MlxeV DX 8770 SDN enabled router 48 ports gigabit
IP	\$4,599	J	Cisco	Cisco 3560 48 port gigabit