Virtual Firewall Performance as a Waypoint on a Software Defined Overlay Network

Casimer DeCusatis
IBM Corporation / Marist College
Poughkeepsie, NY

Peter Mueller
IBM Corporation
Zurich, Switzerland

***Abstract*— Cloud computing environments face many unique security challenges. Location-based firewalls with static policies require long provisioning times relative to other cloud data center components, and are not well suited to dynamic, virtualized workloads. In this paper, we discuss the use of virtual firewalls facilitated by software defined network overlays with forwarding graphs. Experimental results and performance measurements will be presented using a variety of workloads running over a virtual firewall deployment with an industry standard virtual overlay network.**

Keywords—Virtual, Firewall, SDN, NFV, DOVE

I. INTRODUCTION

Recently, the information technology industry has taken steps towards replacing capital intensive, manually provisioned Layer 4-7 network appliances with automated, highly virtualized software applications running on commodity servers. Originally pioneered by the telecommunications industry and the ETSI standard [1] for network function virtualization (NFV), this method has been adopted within data centers and cloud computing environments as well. This approach is motivated by factors including potential cost savings, faster service deployment, and new functionality such as the ability to dynamically provision multiple NFV appliances in function graphs (also known as service chains). It has been estimated [2] that some parts of the industry spend around $54 B annually for purpose-built, single-function hardware appliances; significant cost savings might be realized by using software appliances shared across multiple hosts to create a more agile, application aware network with higher levels of application orchestration and automation. While sever virtualization is relatively mature, and its benefits are well established [3], the virtualization of network devices and appliances such as firewalls remains a subject of ongoing research.

NFV is related to the concept of software defined networking (SDN), an enabling technology which abstracts the network control and management planes from the data plane [3, 4]. Potential benefits of SDN include a dynamic, flexible network infrastructure and significantly faster service deployments. SDN encompasses a variety of enabling technologies which can be implemented either together or separately, including physical flow control using the industry standard OpenFlow protocol [4] and virtual flow control using network overlays. Since overlay networks do not require replacement of existing switches and routers, they are being adopted in the short term as a first step towards full network virtualization. We will confine our attention to overlay networks for the remainder of this paper. In particular, the IETF industry standard NV03, known as Distributed Overlay Virtual Ethernet (DOVE) [5], is a network overlay which abstracts all Layer 2 and 3 addresses in the underlying physical network. DOVE also deploys a virtual switch in the server hypervisor to insure multi-tenant isolation in cloud computing environments. DOVE leverages the VXLAN [6] encapsulation and tunneling protocol, and one of its unique features is a network connectivity service which maps virtual machines (VMs) on the server to their corresponding overlay networks (analogous to an Internet Domain Name Server) [3, 5]. The connectivity service built into DOVE offers the advantage of scaling the network overlay in a more cost effective manner for large cloud deployments. Traditional network security is limited by physical firewall chassis deployment, and does not scale as easily as virtual firewall deployments. Since virtual firewalls and other virtual Layer 4-7 appliances can more easily be placed anywhere in a network, they can help prevent stranded resources in the data center. Traditional static security appliances impose fixed boundaries between trusted and un-trusted regions of the network. In an agile, highly virtualized environment, a different approach to security using location independent virtual firewalls and SDN dynamic traffic routing can potentially achieve much higher utilization of the data center resources.

A wide range of applications can potentially benefit from this combination of NFV and SDN, including public and private cloud environments. For example, over 70% of large enterprises have deployed some form of private cloud, and public cloud represents a $130 B market with 17% compound annual growth rate [7]. In 2012 alone there were 300 cloud service launches [8], and nearly all the major service providers now offer cloud services [9]. New service offerings and

revenue opportunities can be realized more quickly by leveraging a flexible, on-demand infrastructure for managed network services. According to some analysts, firewalls and virtual security management are among the leading near term NFV use cases [2]. Provisioning of security appliance policies is a good candidate for automation, since it is particularly complex and labor intensive, as well as being prone to human errors which can create vulnerabilities.

When comparing physical and virtual firewall deployment, we must bear in mind that even conventional physical firewalls use a number of "virtualized" features. For example, it is common practice to leverage VLAN trunking and tagging, and VLAG interface bundling, to secure boundaries between trusted and un-trusted portions of the network. Virtual firewalls may offer advantages in highly virtualized server environment, which currently lack an efficient solution for securing traffic between VM workloads. In such environments, a physical firewall requires traffic engineering to route data flows through the hypervisor, outside the physical server, through the physical firewall, then back into the same server and hypervisor. While this feature has been enabled on modern routers using the so-called "hairpin turn" specifications in the IEEE 802.Qxx Lossless Ethernet standard [10], it is far from an optimal approach to firewalling high volumes of inter-VM traffic. This issue can be avoided by deploying virtual firewalls as part of the server environment. This approach becomes potentially even more effective when combined with traffic filtering functions of software defined networks (SDN).

Despite the many potential benefits of implementing virtual firewalls on a virtual SDN overlay network, this approach has not been investigated previously. It is of interest to understand the performance implications of sharing virtual firewalls with multiple VMs, deploying firewalls closer to the VMs, and layering processor intensive features such as IPSec into these environments. In this paper, we explore the combination of virtual firewall appliances as a waypoint on a DOVE virtual overlay network with function graphs. We measure performance of this combination under different workload conditions (for the first time to our knowledge), including effects of the DOVE connectivity service (a novel feature which insures secure, multi-tenant isolation all the way back into the server hypervisor virtual switch). Use of an open source overlay controller (the Linux Foundation's Open Daylight [11]) is demonstrated in this environment. We present experimental results and performance measurements for a virtual firewall deployed in a DOVE overlay network, and make recommendations for future data center architectures.

II. Architecture for Virtual Firewall Waypoint

The reference architecture for integrating virtual network overlays and NFV appliances such as virtual firewalls is shown in figure 1. Physical servers may support many VMs, which may have different operating systems. Each VM operating system is installed and managed under a software hypervisor, which also serves to isolate individual VMs from each other. Each VM has virtual I/O connections which attach to a virtual switch in the hypervisor (analogous to physical network interfaces on a physical server). A virtual firewall appliance consists of software which can be hosted in a VM, rather than a separate physical appliance which occupies rack space and requires its own power supply, cooling, etc. Different VMs may host different workloads and applications (either on the same physical server or different physical servers). In a cloud computing environment, different cloud tenants can be hosted in different VMs, which may share the same physical server.

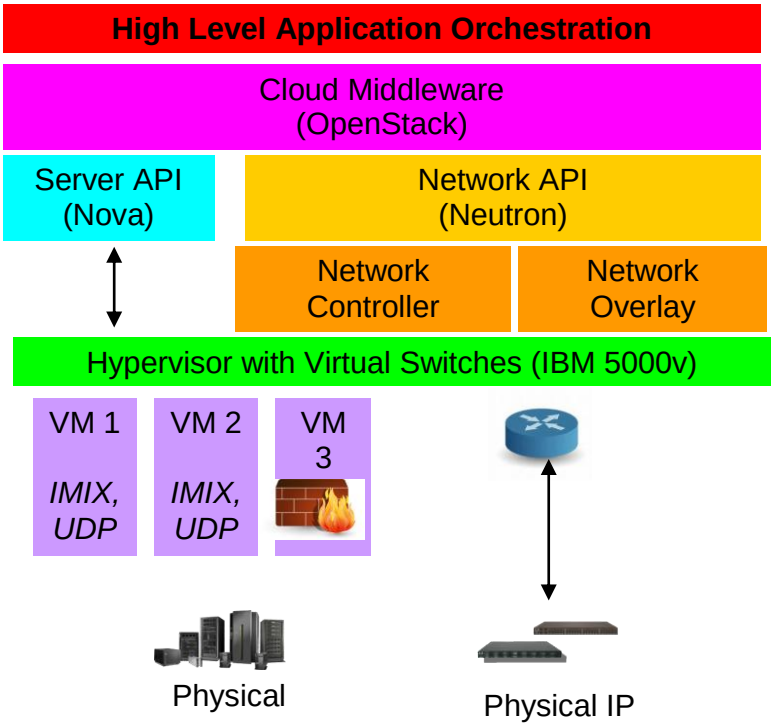

Figure 1 – Architecture for a combined virtual firewall and virtual network overlay

In order to configure data paths between the desired VMs and virtual firewalls, a virtual overlay network is required. The overlay network abstracts or virtualizes all the Layer 2/3 addressing from physical switches connected to the servers and virtual switches contained in the hypervisor. The network overlay is created and managed by a network controller, which contains all the flow tables and network topology information (obtained from a representational state transfer (RESTful) application programming interface (API) on the physical and virtual switches). Data packets created by the VM are encapsulated within a virtual network header (such as VXLAN) and tunneled through the network. Since the tunnels extend back into the hypervisor virtual switch, they provide secure isolation between tenants hosted in different VMs (either on the same physical server or different physical servers). In addition to provisioning end-to-end traffic flows between VMs and provisioning sufficient bandwidth for these flows, the network controller and virtual network overlay can also create function graphs, which instantiate NFV appliances

as waypoints on the virtual overlay network. The network controller includes rules for routing encapsulated packets, controlling virtual traffic flows, and handling packets with different types of virtual headers. It is not necessary for all packets to be handled by the network controller, since the controller can be programmed to handle common packet types (this significantly reduced management communication between the controller and the network). It is also possible to create virtual network profiles, or patterns, which define virtual traffic flows and security policies for a given application (such as online banking or retail systems). These standard profiles can then be saved to the equivalent of an application store in cloud environments, from which they can be downloaded and installed quickly on multiple instances of virtual servers and switches. This greatly simplifies repeated provisioning of resources, accelerates application deployment time, and helps avoid errors by automating the provisioning process.

Both the server hypervisor and network controller support northbound APIs which interface with cloud middleware to provide infrastructure as a Service (IaaS). The IaaS software may include other functions such as user authentication, storage or database access, and higher level application aware orchestration.

III. EXPERIMENTAL TEST BED

To demonstrate the configuration of virtual security appliances in NFV function graphs, we have configured a virtual firewall as a waypoint on a virtual overlay network using DOVE encapsulation and tunneling. Referring to the architecture shown in figure 1, our test bed is compatible with OpenStack IaaS using the Nova and Neutron APIs [12]. The network controller is Open Daylight Helium release (September 2013), an open source project sponsored by The Linux Foundation and available at no cost under an Eclipse Public License. The southbound API from the service layer of Open Daylight abstracts the data plane elements for virtual and physical switches. The Open Daylight controller includes an open source implementation of DOVE, specifically a version of IBM Software Defined Networks for Virtual Environments (SDN VE) version 2.0 (which supports VMware and KVM environments). The northbound API from Open Daylight is compatible with the OpenStack Neutron interface. Details of the APIs have been published previously [5, 11, 12] and will not be repeated here for the sake of brevity.

Our testing was conducted in a VMware 5.0 environment, using Intel e1000 vNICs running at 10 Gbit/s each. Performance measurements were made with the virtual firewall deployed on a Dell PowerEdge R910 server with 40 cores, 2.39 GHz CPUs, running VMware ESXi 5.0. Various industry standard test workloads were measured, running in dedicated virtual machines, including the IMIX TCP/IP traffic profile [13], representative of typical Internet traffic, and the UDP traffic profile [14], characteristic of workloads with large volumes of storage traffic and large data block sizes. Comparison tests were run using both IPv4 and IPv6. Other virtual machines hosted the virtual switch (IBM 5000v, which is included with SDN VE 2.0) and the virtual firewall under test (Juniper perimeter vSRX, also known as Firefly, February 2014 release). The virtual firewall used in this test setup runs the Linux-based 32 bit JunOS operating system on an Intel x86 architecture server. According to the virtual firewall manufacturer's specifications [15], the firewall requires a minimum of 2 GB vRAM per instance and 2 vCPUs per instance. The firewall supports up to 4K VLANs, 256K PAT sessions (using source network address translation (NAT) with port address translation (PAT)), and 256K concurrent firewall sessions.

The logical connections within our test bed are shown in figure 2. Two virtual networks are created which interconnect separate VMs on different physical hosts through a virtual firewall, to an external network such as the Internet. The physical network underlay includes open shortest path first (OSPF) configured across several different network devices, including Juniper Qfabric and IBM/Lenovo G8264 10/40G Ethernet switches.

The virtual firewall provides several functions, including dynamic host configuration protocol (DHCP) address assignment between two virtual networks (vNET1 and vNET2 in figure 2), stateful firewalling and NAT between a private trusted zone and a public untrusted zone, and secure access from outside the data center overlay network. Although the virtual firewalls can be managed as separate devices with static interfaces, using the DOVE overlay network we were able to implement dynamic configuration management through an OpenStack orchestrator. We confirmed that the same default gateway can be used across multiple VMs with a combination of DHCP and static addressing, as shown in figure 2. We also demonstrated that it was possible to save the resulting configuration as a network pattern for rapid, automated re-use in other parts of the network.

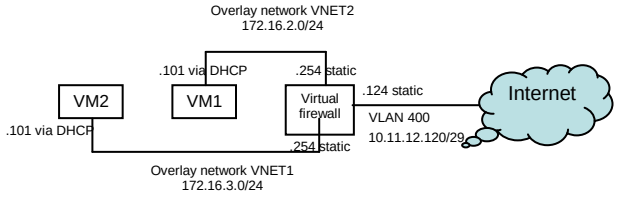

Figure 2 – Logical network with virtual firewall acting as default Internet gateway for two VMs (also performing NAT and stateful failover). VM1 and VM2 are hosted on different ESXi servers located in different physical data centers, and the virtual firewall is hosted on another ESXi server located in data center 1.

IV. EXPERIMENTAL RESULTS

An evaluation of the virtual firewall perimeter performance (throughput in Gbit/second under different workloads) is summarized in table 1. We note that performance may vary slightly with more recent versions of VMware, and may vary significantly in a KVM environment. As expected, performance can vary depending on workload conditions. When using the standard IMIX TCP/IP traffic profile in an IPv4 environment, firewall throughput was around 1 Gbit/s. This is representative of results obtained from evaluating other vendor firewalls under similar conditions (see table 2). Similar performance was measured when enabling NAT on the virtual firewall. Changing from IPv4 to IPv6 resulted in a slight improvement in throughput (1.5 Gbit/s). The ability to maintain or increase throughput with IPv6 is important, since the larger IPv6 address space is expected to play an important role in the emerging Internet of Things, which some analysts project will result in billions of embedded sensors, each with its own IP address, deployed within the next 5 years.

We also note that performance improved significantly (slightly over 4 Gbit/s) for both regular and NAT enabled virtual firewalls when a UDP traffic pattern was employed. This traffic profile is characteristic of workloads with large volumes of storage traffic and large data block sizes, and suggests that this firewall configuration would be a good candidate for scaling to such workloads. Measurements on similar virtual firewall products from other vendors are shown for comparison in table 2 (vendor names have been removed to preserve anonymity). It is apparent that some current generation virtual firewall products do not scale as well to high volume workloads, providing only about 1 Gbit/s throughput.

Average latency introduced by the virtual firewall was measured to be 102 microseconds. Although the latency results given in table 1 represent UDP data, no appreciable difference in latency was measured when using IMIX or IPv6. This latency is adequate for the vast majority of practical applications, although care should be taken in latency-sensitive environments such as real time stock trading.

Table 1 – virtual firewall performance testing with different workloads

Performance Test	**Results**
Firewall with IMIX, IPv4	1.0 Gbit/s
Firewall with UDP (1500 B packets), IPv4	4.1 Gbit/s
Firewall with UDP (512B packets) IPv6	1.5 Gbit/s
Firewall with NAT (1500B packets), IPv4	4.0 Gbit/s
Latency (512B UDP)	101 microseconds

Table 2 – Relative throughput comparison of three different virtual firewalls (vendor names kept anonymous)

Vendor A	1.1 Gbit/s
Vendor B	1.3 Gbit/s
Vendor C	1.0 Gbit/s

The impact of enabling optional security features on the virtual firewall, such as IPSec (triple DES plus SHA), is shown in table 3. As expected, the virtual firewall throughput decreased significantly from the performance shown in table 1 due to the overhead associated with processing IPSec protocols. However, the throughputs shown here are sufficient for many practical use cases, including perimeter security for cloud computing data centers. While running IPSec for secure tunnel creation, the virtual firewall was able to process about 80 tunnels per second, which is comparable to a physical firewall under similar conditions.

Table 3 – Performance impact of enabling IPSec on virtual firewall

Performance with IPSec (triple DES, SHA-1)	
IMIX	66 Mbit/s
UDP (64 B packets)	70 Kbit/s

Since the features and performance of virtual and physical firewalls were comparable, we also compared provisioning time for a virtual firewall instance and a physical firewall from the same vendor. In a typical installation, provisioning of the physical firewall requires making changes to the cable infrastructure, and re-provisioning multiple firewall instances one at a time. Depending on the complexity of the installation, this process can be quite time consuming. For example, end-to-end provisioning of traffic flows from another vendor's routers, firewalls, and load balancers can take up to 5 days [16], and we have previously measured firewall provisioning times on the order of several days for cloud computing environments involving less than five virtual machines. Since the virtual firewall does not require any changes to the cable configuration, it can be re-provisioned in a few minutes. Using optional software applications running external to the virtual firewall [15], it becomes possible to re-provision multiple firewall instances without appreciably increasing the overall provisioning time; this is an advantage over physical firewalls.

When deploying virtual firewalls, the number of firewall instances per server under different conditions and the number of administrators required per instance are important considerations which affect total cost of ownership. The number of firewall instances which can be deployed on a single physical server is a function of the server processing speed, number of cores/server, memory, and network I/O. We measured two configurations to provide a range for this value.

First, an 8 core Intel x86 architecture server with a clock rate of 2.66 GHz, 64 GB RAM, and two Ethernet connections at 10 Gbit/s each, allowed us to achieve the performance shown in tables 1-3 with 20 virtual firewall instances installed on the server. Second, a 40 core server with a clock rate of 2.39 GHz, 256 GB RAM, and four Ethernet connections at 10 Gbit/s each, allowed us to install 100 virtual firewall instances per server.

Given that we have shown acceptable performance of the virtual firewall over a wide range of install instances, we next consider the architectural implications for virtual firewall deployment. At one extreme, we could simply replace each physical firewall in an existing data center with a server running a dedicated virtual firewall. This approach does not result in appreciable hardware, energy, or space savings, and may result in the same type of issues facing static physical firewalls (for example, creating traffic trombones within the network architecture as a result of firewalls being in a fixed location while workloads migrate to different VMs). This approach does have the advantage of simplicity and ease of migration from a legacy physical firewall environment. Since each firewall handles a limited amount of traffic, the definition and deployment of firewall rules is straightforward and requires minimal administrative effort. At the other extreme, in order to provide more cost effective scalability and reduce energy consumption, it is desirable to deploy one virtual firewall for multiple servers (perhaps one firewall per rack), with the firewalls placed closer to the servers they are intended to protect. This placement results in simpler rules for each firewall, but more firewall instances to manage and thus higher administrative effort, so there is a tradeoff in overall cost. A quantitative evaluation of the actual costs would involve details of the virtual firewall licensing approach (per VM, per CPU, etc), and the labor/burden rate to estimate cost per administrative instance. This is beyond the scope of our current work, but may be considered as part of future research in this area.

For high availability, the virtual firewall supports stateful failover and either active-active or active-passive operation. Two virtual firewall instances interconnect to form a cluster, using a control link (for kernel state replication, configuration, and heartbeat functions) and a fabric link (for data traffic forwarding). Thus, there is strict separation between the control and data planes in the virtual firewall, just as in other aspects of an SDN network. When two virtual firewalls are connected in this manner, only one virtual instance is shown by the management interface (there is a global management view and naming space for interfaces on all virtual firewall instances). In an active-passive configuration, the master instance is selected by a priority level in the redundant group defined for the virtual firewall management interface. Redundant physical interfaces are aggregated to simulate virtual interfaces as part of the stateful failover approach. This results in stateful failover of redundant virtual firewall instances, so the virtual firewalls can be effectively deployed in pairs (or on clusters of physical servers) for high availability applications. The redundancy and failover features of the virtual firewalls tested were comparable to physical firewalls from the same vendor.

V. Discussion and Conclusions

We have demonstrated (for the first time to our knowledge) performance resulting from the combination of a virtual firewall as a waypoint on a data center virtual overlay network (using DOVE industry standard protocols). This approach is compatible with OpenStack IaaS, since the open source network controller (Open Daylight) has a RESTful API compatible with OpenStack Neutron. There are many other Layer 4-7 applications which can be virtualized and run as waypoints on a virtual network overlay, including load balancers/application delivery controllers, deep packet inspection, and intrusion detection/prevention [17].

Performance, latency, and high availability measurements indicate that both typical and high storage volume workloads can comfortably share a single virtual firewall instance across multiple VMs. Thus, practical deployments can replace a small number of physical firewalls placed relatively far from the servers they protect with more instances of virtual firewalls placed closer to the servers. Tradeoffs between the increased flexibility and lower device cost must be weighted against potentially higher administrative costs associated with more complex firewall rule configurations and more virtual firewall instances. Based on our test results, potential advantages of virtual firewalls include the replacement of physical firewalls with savings on rack space, power, and cooling, without sacrificing features or performance available on physical firewalls. Virtual firewalls were also significantly faster to deploy than physical firewalls in our test bed, and faster to reconfigure into service chains (minutes vs days). Potential disadvantages of virtual firewalls include higher complexity in firewall rules and the associated impact on operating costs, depending on the number of firewalls deployed and their proximity to the servers they protect (generally speaking, firewalls placed closer to the physical servers have simpler security policies).

Potential future research topics may include repeating these performance measurements in a KVM environment, and integrating firewall and VM management with a higher level application orchestration engine. Potential enhancements to the virtual firewall include leveraging Intel encryption libraries and data plane development kit (DPDK) for improved VM acceleration, and support for a 64 bit operating system which would allow the virtual firewall to exploit a larger amount of memory.

References

[1] ETSI General Specification NFV 001 - 004 ver 1.1.1 (October 2013) http://www.etsi.org/technologies-clusters/technologies/nfv (last accessed May 10, 2014 3:05 PM EST)

[2] C. Mendler, "Navigating the telecom cloud", Informa industry report, 20 pp., June 2012 www.informatandm.com (last accessed May 10, 2014 3:05 PM EST)

[3] C. Dixon, D. Olshefski, V. Jain, C. DeCusatis, W. Felter, J. Carter, M. Banikazemi, V. Mann, J. Tracey, R. Recio, J. Kidambi, A. Biswas, and U. Shankar Nagaraj, "Software defined networking to support the software defined environment", IBM Journal of Research & Development, p. 204-216 (April/May issue 2014)

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks", ACM SiGCOMM vol 38, no 2, p. 69-74 (April 2008)

[5] L. Dunbar and D. Eastlake, draft IETF NV03 working group standard reference, Sept. 20, 2013 https://ietf.org/meeting/88/agenda/nvo3-drafts.pdf (last accessed May 10, 2014 3:05 PM EST)

[6] M. Mahalingam, "VXLAN: a framework for overlaying virtualized Layer 2 networks over Layer 3 networks", IETF active internet draft standard ver 9 (April 10, 2014) https://datatracker.ietf.org/doc/draft-mahalingam-dutt-dcops-vxlan/ (last accessed May 10, 2014 3:05 PM EST)

[7] Gartner Group reports, "Public cloud forecast" (June 2013) and "Private cloud matures" (September 2013), www.gartner.com (last accessed May 10, 2014 3:05 PM EST)

[8] K. Taga, S. Best, D. Levy, and N. Racz, "Cloud from telcos; business distraction or key to growth?", Arthur D. Little Research Report, 27 pp, July 2013 www.adl.com/cloud_from_telcos (last accessed May 10, 2014 3:05 PM EST)

[9] Infonetics report, "SDN and NFV Strategies Survey Highlights", 27 pp, June 15, 2013 www.infonetics.com (last accessed May 10, 2014 3:05 PM EST)

[10] IEEE 802.1Qxx, "Data center bridging", May 2012, http://www.ieee802.org/1/pages/802.1az.html (last accessed May 10, 2014 3:05 PM EST)

[11] Linux Foundation Open Daylight, Helium Release (December 2013), open source download available at www.opendaylight.org (last accessed May 10, 2014 3:05 PM EST)

[12] OpenStack open source cloud middleware, documentation and free downloads from https://www.openstack.org/ (last accessed May 10, 2014 3:05 PM EST)

[13] IETF RFC 6948, "IMIX Genome", July 2013 http://tools.ietf.org/html/rfc6985 (last accessed May 10, 2014 3:05 PM EST)

[14] IETF RFC 768, "UDP Protocol", August 1980 https://www.ietf.org/rfc/rfc768.txt (last accessed May 10, 2014 3:05 PM EST)

[15] Juniper Networks data sheet, Junos Security Director, February 2013 ,http://imtech.com/Content/ImtechICT%20UK/Documents/PDF/Infosec%20datasheets/junosspacesecurity.pdf (last accessed May 10, 2014 3:05 PM EST)

[16] J. Manville, "The power of a programmable cloud", OFC 2012 annual meeting, Anaheim, CA, paper OM2D.2 (March 18-22, 2013)

[17] C.J.S. DeCusatis, A. Carranza, and C. DeCusatis, "Communication within clouds: open standards and proprietary protocols for data center networking", IEEE Communications Magazine, p. 44-51 (October 2012)