Electronics **2015**, *4*, 633-650

OPEN ACCESS

electronics

ISSN 2079-9292

www.mdpi.com/journal/electronics

Article

Reference Architecture for Multi-Layer Software Defined Optical Data Center Networks

Casimer DeCusatis

School of Computer Science and Mathematics, New York State Cloud Computing and Analytics Center, Marist College, Poughkeepsie, NY, 12601 USA; E-Mail: casimer.decusatis@marist.edu; Tel.: +1-845-575-3883; Fax: +1-845-575-3605.

Academic Editor: Lei Liu

Received: 24 July 2015 / Accepted: 8 September 2015 / Published:

Abstract: As cloud computing data centers grow larger and networking devices proliferate; many complex issues arise in the network management architecture. We propose a framework for multi-layer; multi-vendor optical network management using open standards-based software defined networking (SDN). Experimental results are demonstrated in a test bed consisting of three data centers interconnected by a 125 km metropolitan area network; running OpenStack with KVM and VMW are components. Use cases include inter-data center connectivity via a packet-optical metropolitan area network; intra-data center connectivity using an optical mesh network; and SDN coordination of networking equipment within and between multiple data centers. We create and demonstrate original software to implement virtual network slicing and affinity policy-as-a-service offerings. Enhancements to synchronous storage backup; cloud exchanges; and Fibre Channel over Ethernet topologies are also discussed.

Keywords: cloud; software defined network; network function virtualization; Fibre Channel over Ethernet; optical

1. Introduction

Traditional information technology (IT) infrastructures manage servers, storage, and networking as homogeneous silos, with extensive manual intervention by highly skilled system administrators. In particular, optical and other types of networks are statically configured, often with low bandwidth utilization. Virtual network resources are also statically configured, and often managed in the same manner as the hardware resources which they replace. This hardware-centric management approach is no longer sustainable, due to the emergence of dynamic, multi-tenant cloud services, growing applications such as mobile computing or big data analytics, and the need to reduce capital and operating expenses. A new approach to workload aware management is emerging across the cloud networking industry [1–15] in which servers, storage, and networking are treated as pools of fungible resources which can be assembled and managed as a single entity, using software virtualization and advanced automation tools. This leads to a more flexible, elastic service delivery model which radically improves time to value and availability for new service deployments. Higher utilization provides efficiency gains which lower the total operating cost. Automation improves the repeatability, consistency, and stability of the environment, and enables workload optimization.

In response to the requirements of this new environment, existing optical networks within and between data centers have been driven to adopt new characteristics Recent reports indicate that cloud data center architectures are now the largest volume driver for optical networks, surpassing conventional telecommunication systems [2]. The optical network environment is characterized by a programmable infrastructure using open application programming interfaces (APIs). Services are dynamically assigned to the best set of available resources based on characteristics of the application. Programmable APIs enable continuous optimization to quickly address infrastructure issues (such as providing self-healing or workload scaling) and make the environment more responsive to business needs. A key element of this approach is multi-tenant services, which leverage virtualization and abstraction technologies to provide different users with secure access over a shared optical infrastructure. For this reason, multi-tenancy has become an important part of the $130 B cloud computing market [1].

New cloud service offerings and revenue opportunities can be realized by creating a flexible, on-demand network infrastructure. While servers and storage can be abstracted and virtualized using software, similar tools have not been available for network equipment until very recently. The lack of network virtualization has created a significant bottleneck to the deployment of flexible, dynamic data centers. This issue is currently being addressed with the introduction of software defined networking (SDN) and network virtualization [6–9]. SDN network management involves separating the data and control/management planes within network equipment, and using a centralized controller to oversee the entire network. These features enable faster service provisioning and lower costs due to the resulting automation and efficiency gains. The network and control planes can now be developed and optimized independently, and the use of standardized management APIs with vendor-specific plug-ins means that equipment from different vendors can be accommodated in the same management plane. This helps avoid vendor lock-in since equipment from multiple vendors can be managed with a common set of control software. Network applications can be more easily deployed on top of a centralized controller, as opposed to traditional approaches which need to change the equipment firmware or control logic to implement new services. However, the use of SDN in multi-tenant optical data communication networks is a relatively new topic, and there is a need for a network management architecture which addresses the unique requirements of this environment.

In this paper, we present an SDN management architecture and experimental test bed demonstrating deployments of multi-tenant optical networks within and between cloud data centers. The test bed used in these experiments consists of three data centers running production level workloads and traffic patterns using commercially available network hardware; the test bed includes software which we created to enable new SDN features and functions. In some experiments, traffic generators were used in order to reproduce real world use case data, collected from a hybrid cloud production network. Although the management framework is hypervisor agnostic, the test bed illustrates the co-existence of VMware and KVM implementations with multiple open source network controllers. This is a practical issue for many deployments, which may use different hypervisor environments for different parts of their data centers. Further, this approach enables data centers which may need to merge environments running different hypervisors, without the need for extensive reconfiguration.

The paper is organized as follows. Section 2 describes related work in the area of dynamic network management, which provides background and context for the contributions of this research. Section 3 reviews our proposed four layer network management architecture with particular emphasis on the control of optical devices. This reference architecture will be deployed in a test bed to demonstrate important SDN features in the following sections. Sections 4–6 present the results of implementing this architecture to address different use cases. Section 4 discusses inter-data center applications, including dynamic provisioning of a packet-optical wide area network (WAN) or metropolitan area network (MAN), and virtual network slicing on various time scales. This demonstrates multi-tenancy in an optical MAN/WAN and significantly faster dynamic resource provisioning. Section 5 discusses intra-data center applications, including affinities for optical mesh SDN and use of the OpenStack Congress API for optical network control. This demonstrates new functionality within the data center enabled by SDN, and extends the dynamic re-provisioning developed in Section 4 beyond the MAN/WAN demarcation point, back into the data center optical network. Section 6 discusses multi-layer SDN (the simultaneous management of equipment within and between multiple data centers). This combines the results of Sections 4 and 5 to demonstrate an end-to-end SDN management application. We present results from implementing the proposed network management architecture using an SDN network test bed, including software we have created to enable SDN management of optical devices. This includes the first example of an OpenStack Congress driver for optical data center fabrics with affinities. The test bed demonstrates several use cases including bandwidth slicing of an optical network between multiple data centers, affinity policy enforcement for optical networks within a data center, and multi-layer optical transport across data center boundaries. These use cases, including the need for faster dynamic provisioning and the use of optical links within large, warehouse-scale cloud data centers, have been established previously as being of interest to cloud data center designers [1–6]. The test bed is implemented using commercially available optical networking equipment; other design tradeoffs, including the relative cost of optical connectivity within and between data centers, is beyond the scope of our current work. We discuss implications for dynamic optical network re-provisioning on different time scales. We also discuss changes in the storage network architecture enabled by our use of reliable iSCSI over optical transport.

2. Related Work

It has recently been established that reconfiguration of an end-to-end service within a single data center network can take 5–7 days or longer, while provisioning traffic between multiple data centers can take days, weeks, or more [9–13]. This is due to the lack of automated provisioning in current data center networks (both copper and optical). Because data network provisioning is static, these networks are commonly overprovisioned by 30%–50% or more to insure good performance [10,11], For example, conventional optical MAN or WAN networks statically provision bandwidth based on estimated bandwidth consumption over the next 6–12 months [9,10]. Not only does this contribute to stranded bandwidth issues, it is not a cost effective or energy efficient use of network resources. As noted in the introduction to this paper, the need for rapid, dynamic network provisioning is driven by emerging application such as data analytics and mobile computing [9,10]. SDN has been investigated as a mechanism for achieving faster provisioning, in the context of proprietary networking equipment management interfaces [9]. However, it is desirable to introduce an open, standards-based reference architecture for cloud data networking. While these changes apply to all types of cloud data center networks, they are particularly important to optical networks. This is because optical networks afford much greater bandwidth and distance, and are becoming more widely adopted within the data center as well as over extended distances. Recent proposals for an optical network management plane suggested the use of a single network management domain with multiple network controllers [7]. Prior work has shown that a three or four layer architecture yields improved performance and robustness compared with alternative designs [10]. For example, a proprietary three layer variant on this architecture has been proposed for telecommunication service providers [8] which defines application, policy orchestration, and network infrastructure layers.

Our approach differs from these prior proposals through the use of OpenStack middleware for the adaptation, services, and management layer. While it is possible to substitute other forms of cloud middleware, we have selected OpenStack because it provides a standards-based, open source solution which is commonly deployed within large cloud data centers [4] OpenStack APIs are readily integrated with the most commonly used cloud orchestration software, Among other differences, this allows us to use the OpenStack Congress policy-as-a-service API to manage optical mesh networks within a data center. We will demonstrate significantly faster provisioning times, as well as new functionality, using this approach. We also note that previously published architectures [6] provided only a theoretical comparison between different design tradeoffs; in Sections 3–6, we discuss our experimental test bed implementation of this approach. We also extend prior approaches by incorporating multiple hypervisors (our solution works with both KVM and VMWare), and support for direct attach Fibre Channel over Ethernet (FCoE) storage.

While optical networking is often associated with long distance networks between multiple data centers, there are several commercially available optical networking devices designed for use within data centers [16–18]. We will concentrate on devices which implement network affinities between the application management layer and forwarding layer. A study done by Microsoft in 2009 [19] demonstrated the extent and complexity of the relationships between applications running inside a data center. Many interactions between workloads are not random and unpredictable, but have known patterns that users, developers and operators can leverage to more efficiently utilize network resources and achieve desired quality of service (QoS) targets. Once these relationships are identified, they can form the basis for policies that enforce efficient communication, such as prioritizing some applications at the expense of others, or isolating communication between certain network tenants. Early research in this area included so-called “context aware” systems proposed for pervasive computing systems [20], which are made more efficient by collecting larger data sets describing applications and their inter-relationships (in more recent literature, these data sets are known as “affinities” [21]). Network affinities are enabled in the most recent versions of the Open Daylight controller [6], but there is currently no way to interface affinities with OpenStack middleware. We describe our implementation of a driver for affinity implementation with the OpenStack Congress "policy-as-a-service" engine, which to our knowledge is the first such driver created by a third party. Finally, we will combine the management planes within and between data centers in our test bed.

3. SDN Optical Network Management Architecture

In the following section, we present our proposed four layer network management architecture with particular emphasis on the control of optical devices. The proposed management architecture is illustrated in Figure 1, which shows how SDN may be incorporated into a data center cloud computing environment. Each layer is connected with the one above it using a management API; we have illustrated one example showing how OpenStack can be deployed in this environment.

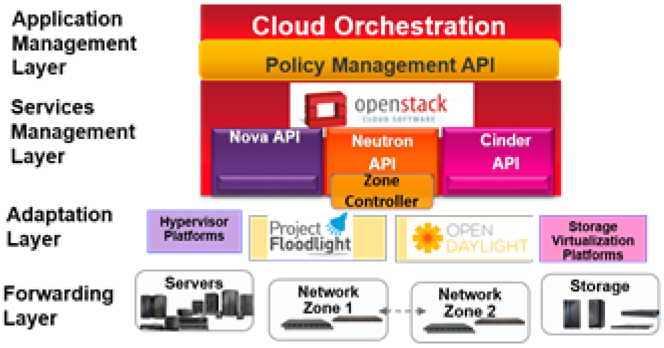

Figure 1. Software defined networking (SDN) optical network management architecture.

The architecture shown in Figure 1 uses four functional layers (forwarding, adaptation, services management, and application management, as well as multiple controllers (each with a partial view of the network).

Advantages of this architecture include the use of a single policy management API which governs security procedures and statistical data collection from the underlying physical layer. Further, our approach is based on open industry standards, including OpenFlow and the OpenStack APIs, which are more readily integrated with cloud orchestration and thus more suitable for data center applications. We also extend prior approaches by incorporating multiple hypervisors, a virtual overlay network, and support for FCoE storage. These features add significant versatility to the resulting architecture, and enable new functionality (for example, co-existence of multiple hypervisors during data center mergers or use of network overlays to enhance multi-tenant isolation). The use of open standards enables a multi-vendor, multi-layer environment; we demonstrate SDN management of optical transport equipment from vendors such as Ciena (Hanover, MD, USA) and Adva (Munich, Germany), and optical networks within the data center including Plexxi affinity switches (Nashua, NH, USA) and optical storage area networks using Brocade switches (San Jose, CA, USA). The test bed also accommodates virtual appliances from Vyatta (San Jose, CA, USA), as well as a number of different brand Ethernet routers, servers, and storage devices. Before presenting the test bed implementation, we describe each of the four layers of our framework in more detail.

At the lowest or forwarding layer, the network is logically partitioned into zones, each of which has its own network controller. Redundant controllers may be present within a zone to provide high availability and avoid single points of failure, however only one controller is active at a given time (the OpenFlow standard discusses recommended procedures for failover between redundant controllers). It is desirable to choose multiple zones based on criteria such as improved latency and performance (avoiding controllers which are located far from some parts of the network). Zones may also be selected based on security partitioning, service level agreements, or other criteria. The proposed architecture distributes control functions across multiple controllers, each of which controls a local zone or segment of the network [9–15]. Dividing the network into zones should also reduce the management traffic load on each controller; this also facilitates performance-efficient scalability. Since the number of devices per zone can be kept fairly small, the controller can poll them more frequently; in this way, the controller can obtain the latest topology and configuration information even if the network configuration is changing.

The second or adaptation layer of the architecture includes zone management functions which accommodate different types of northbound and southbound APIs in the same network. This also enables the use of different brand controllers for each network segment. For use cases which require an end-to-end overview of the entire network, a higher level management API can provide direction to all the underlying controllers. This vertical hierarchy approach, which has been used by other SDN controller [9], should reduce latency between the network devices and controller in most situations, and simplifies network management rules for the higher level controller. For example, the higher level controller might be used to re-route elephant flows across the entire network, while an elephant flow between devices in the same network zone could be handled by the local zone controller. Since the architecture enables frequent polling from the zone controllers, aggregate configuration data can be forwarded from the zone controllers to the adaptation layer as required to support end-to-end decision making. The proposed approach may use a combination of inband and outband management networks; note that this approach is independent of the method used to communicate between controllers.

Above the adaptation layer, the third or service management layer can be implemented using open source middleware such as OpenStack [12]. The OpenStack network management API, known as Neutron, can interface with the upper level hierarchical controller, which in turn manages the zone controllers. Similar APIs (Nova for servers and Cinder for storage) allow for the orchestration of resources beyond the network. Note that while the Cinder API manages file and block storage devices, it does not include storage area network management or a Fibre Channel Forwarder (FCF) interface; these functions must be provided through the controllers under the Neutron API. The service management layer provides functions including topology management, security credential management (OpenStack Keystone), interpretation of the management API messages (OpenStack Congress), and aggregate statistical monitoring. In particular, security features can be implemented such as automated authentication when a new device attempts to connect to the network. OpenStack would maintain the public keys for controllers and network devices, insuring that only authenticated controllers can connect to the management system. Security between the controller and network devices is handled by the switch API protocol (OpenFlow, VMWare NSX, or something similar).

Finally, the fourth or an application management layer provides integration of other network management applications, including firewalls, intrusion detection and prevention systems, load balancers, and more. A RESTful management API provides loose coupling between the application and service management layers, and offers a wide range of features for the service management layer. A loosely coupled approach allows for stand-alone network appliances, such as physical and virtual firewalls, to use their native management interface features while also taking advantage of the latest network topology and configuration information propagated from lower layers of the management model. Application management can also be realized using tools which supervise and abstract many underlying network devices, such as IBM Cloud Orchestrator (ICO). In a later section, we will discuss implementation of ICO interfaces to other optical networking equipment.

The following sections consider several possible optical domains, including inter-data center connectivity via a packet-optical wide area network (WAN) or metropolitan area network (MAN), intra-data center connectivity using an optical mesh network, and SDN coordination of networking equipment within and between multiple data centers.

4. Inter-Data Center Optical SDN: Bandwidth Slicing

In this section, we discuss applications between multiple data centers interconnected by optical networks, including dynamic provisioning of a packet-optical wide area or metropolitan area network, and virtual network slicing on various time scales. We will briefly review previous results demonstrating dynamic re-provisioning of the optical network on a timescale of hours, and discuss new results which reduce the provisioning time to minutes or less. Using this approach, we demonstrate an optical network which can eliminate intermediate storage switches and support native iSCSI with line rate encryption up to 100 Gbit/s.

There are several different use cases for dynamic MAN/WAN provisioning, depending on the desired time scale. For example, there are many interconnected data centers near large metropolitan areas interconnected with optical links. During normal business hours, it may be desirable to operate multiple data centers in a load balanced configuration with uniform bandwidth between all sites. This network might be reconfigured a few times per day, such as during routine daily backups. It may also be desirable to reconfigure the MAN/WAN on fairly short notice for a period of hours or days, such as during a natural disaster, power failure, or other emergency situation. Even this fairly infrequent re-provisioning is not practical without SDN control of the optical network. We have previously demonstrated on-demand re-configuration of an Adva 125 km MAN between three regional data centers using the Open Daylight SDN controller, Adva network hypervisor, and original software created by our research team [16]. This novel approach makes it possible to dynamically re-configure multiple wavelength division multiplexing (WDM) platforms from a single controller (note that this class of application is not latency sensitive, so it is not necessary to use geographically distributed controllers).

On the other hand, some applications require dynamic reconfiguration on a timescale of minutes, equivalent to the time currently required to instantiate new instances of virtual servers or storage. One example is synchronous storage replication between enterprise data centers and public clouds. Many enterprise data centers are scaling out storage capacity by connecting their private data centers with public cloud computing environments, rather than purchasing new storage devices for their own use. This infrastructure-as-a-service (IaaS) approach offers lower total cost of ownership, since the enterprise does not incur fixed costs associated with owning and maintaining storage devices or variable costs associated with ongoing management; further, the enterprise does not need to depreciate hardware cost over time, and benefits from rapid, elastic scaling of storage resources. However, there may be performance concerns associated with the cloud access network, especially for applications which require time sensitive access to large volumes of data. Conventional storage networks accessing cloud environments are statically provisioned, based on estimates of peak bandwidth requirements rather than actual bandwidth consumption. The resulting networks are thus over-provisioned, and most of the allocated bandwidth is not used most of the time. Further, many synchronous storage applications can experience traffic bursts which exceed the pre-provisioned bandwidth. Under these conditions, the storage system can repeatedly fail to complete access requests, causing degraded performance and eventually halting the application altogether. A typical example of this use case is shown in Figure 2 [16], which depicts traffic monitoring data from a hybrid synchronous storage application collected over a seven day period (names of the enterprise and cloud provider have been removed at their request). Multiple traffic bursts are clearly visible, occurring several times per day, the largest of which exceeds typical bandwidth usage by over six times and lasts between 15–30 min. For comparison, static bandwidth provisioning levels are shown; even if these levels increase by 45% year to year, it is still not possible to accommodate the largest traffic bursts. In addition, higher over-provisioning becomes increasingly inefficient in terms of both cost and network resources.

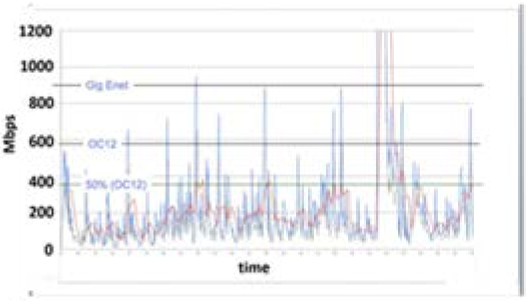

Figure 2. Bandwidth *vs.* time monitoring of hybrid cloud storage network, illustrating bandwidth spikes on the order of tens of minutes in duration.

Existing networks require days or even weeks to re-provision bandwidth, far too long for management of these traffic bursts. Since it is not practical to solve this problem with static bandwidth provisioning, we have demonstrated a dynamic provisioning use case which can respond to even the largest traffic bursts shown in Figure 2, while aligning network capacity with application demands at other times. We show experimentally that end-to-end re-provisioning of the optical network (including Ethernet switches within the data center and WDM transport between data centers) can be reduced to minutes or less using SDN control. This behavior is demonstrated using policies defined at the network orchestration layer and pushed down the software stack shown in Figure 1 to the physical layer. The test bed is shown in Figure 3, consisting of a 125 km single-mode fiber ring interconnecting three metropolitan area data centers. One data center represents the cloud provider, while the others are enterprise data centers sharing multi-tenancy in the cloud. The sites are interconnected with a dense wavelength division multiplexing (WDM) platform (Adva FSP3000, Munich, Germany), including excess, discretionary wavelength pools which can be applied to the enterprise data center connections. Each site also contains inexpensive demarcation point hardware (Adva XG210) which serves as a traffic monitor and can also inject traffic patterns so that we can reproduce the traffic profile collected in Figure 2 under controlled conditions.

Data center iSCSI storage resources are connected via 1/10 Gbit/s Ethernet switches (Lenovo G8264, Morrisville, NC, USA and Beijing, China). Servers at each location host virtual machines (VMWare environment), which contain software defined network (SDN) controllers for the Ethernet switches and a network hypervisor for the WDM optical equipment.

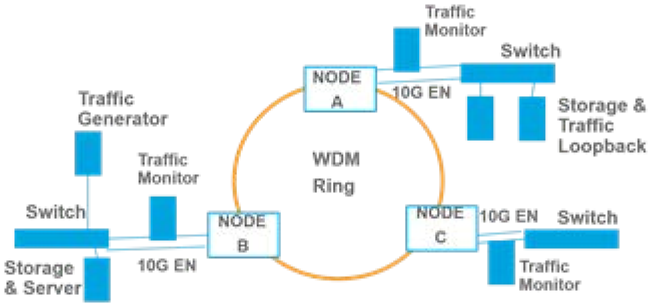

Figure 3. Packet over optical wavelength division multiplexing (WDM) test bed using SDN and network hypervisor control.

The software management plane for this test bed is illustrated in Figure 4. We have created an extension for open source SDN controllers and the WDM network hypervisor, known as “Advalanche”, which performs dynamic bandwidth provisioning based on the industry standard protocol OpenFlow 1.3.1. Furthermore, this approach offers the additional functionality of slicing the physical network into two or more virtual network segments, each of which can be assigned to a different tenant. Each tenant is then able to use an open standards-based SDN controller of their choice to optimize bandwidth allocation within their slice of the network. There are a number of different controllers which have been proposed (including Open Daylight, Floodlight, and several vendor proprietary versions) as well as different server hypervisor environments including KVM, VMWare, and others. These controllers lack a common northbound or application level API, making it difficult for more than one type of controller to exist within the same network. This has forced many users to choose between controllers, each of which has its own strengths and weaknesses, and let to fragmentation of open source controller development efforts. Our approach addresses this issue by allowing each tenant to use an SDN controller of their choice, which may be different from the controller used by the cloud service provider (CSP) to slice the network.

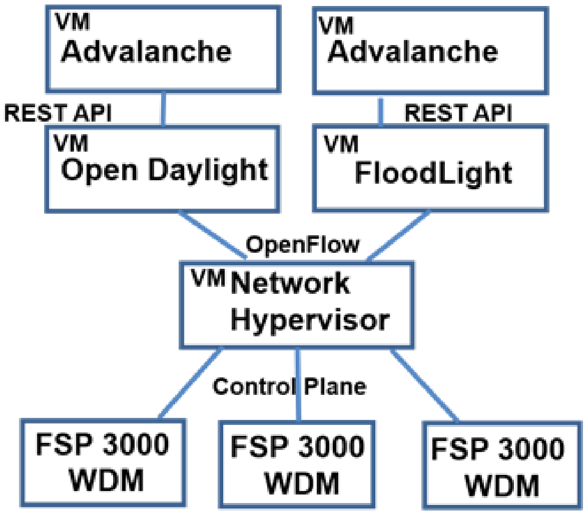

Figure 4. Software control plane architecture for cloud service provider with two tenants.

Large, highly virtualized data centers (including public and private clouds) need to share data center resources across multiple tenants in this manner. In a private cloud, for example, different departments in the same organization (research, sales, marketing, *etc.*) may need to share resources but remain logically isolated from each other for security and performance reasons. Similarly, in a public cloud multiple companies may need to share resources while maintaining separation of their network traffic. This requires a mechanism to isolate network data flows from each other, preferably end-to-end across the entire network (including virtual switches in the server hypervisor). In an SDN network, multi-tenant isolation can be performed by the network controller. The controller may dynamically isolate physical flows using OpenFlow. Network management functions can be automated and saved as a profile or pattern, which may be reused on other parts of the network or customized for other applications.

To demonstrate multi-tenant slicing, our test bed divides the network into two virtual slices, then runs the Floodlight controller in one slice, and the Open Daylight controller at the same time in another, logically independent slice. As we will show in Section 6, the Ethernet switches within a data center (which contain optical links) and the WDM equipment between data centers are all controlled from a common management plane. The physical topology is shown in Figure 4, and the logical topology in Figure 5. Policies at the application layer can be used to create an end-to-end path from a server in one data center, through the first data center network, across the WDM network to a second data center, and then through the second data center network to a destination server. Each tenant is assigned a virtual network slice, and they can optimize their individual WDM bandwidth using Advalanche, which may be called from either Floodlight or Open Daylight; each tenant's controller can also provision Ethernet switches within the tenant's location. The XG210 hardware is integrated in the FSP3000 package, and managed from the network hypervisor using SNMP 3.0.

The XG210 monitors traffic, and is able to detect an impending traffic burst like those shown in Figure 2. In this way, traffic monitoring is non-invasive to the data center, as opposed to previous work that required monitoring servers within the client's data center [18]. This triggers automated re-provisioning of Layer 0–3 connections to temporarily provide increased bandwidth for the duration of the traffic burst. We have experimentally achieved end-to-end re-provisioning in under a minute, more than fast enough for responding to measured traffic bursts. Additional wavelengths for the affected application may be provisioned from an available wavelength pool, or by re-allocation of discretionary wavelengths from other applications on the ring (this will depend on existing service level agreements). Once traffic returns to nominal levels, both the networks within and between data centers are restored to their previous configurations automatically. Throughout the automated re-provisioning process, the controller management user interface is updated dynamically as changes are made to the network equipment.

To our knowledge, this is the first implementation of a standards-based, dynamically reconfigurable data center and optical transport network. Since OpenFlow is a relatively new standard, realistically it will take a long time for most network equipment to support this interface. We have used open standards based software wherever possible in our test bed (including OpenStack, Open Daylight, and OpenFlow). Vendor specific APIs have been accommodated through software we have created at the adaptation layer of Figure 1, and which we have released as open source. To demonstrate the new functionality in our test bed, we have generated traffic bursts between 5–20 Gbit/s with durations of 5–10 min each. In all cases, our automation software was able to dynamically implement a policy to accommodate these traffic bursts (*i.e.*, adding additional wavelengths) in less than a minute. In some cases, the network re- provisioning time were actually faster than the response time for provisioning new virtual machines.

The ability to reliably delivery native iSCSI storage traffic over an optical WDM network has further implications for the storage network architecture inside the data center. Specifically, native storage over SDN controlled optical networks offers advantages over conventional approaches using Fibre Channel over IP gateways or SONET, as shown in Figure 5. Using SDN to provide dynamic bandwidth on demand for a packet optical network running over native wavelengths and dark fiber, we have shown that it is possible to eliminate hardware required by other solutions (such as Fiber Channel directors, FC/IP gateways, and SONET MSPPs). Further, benefits of this approach include simplified migration to higher data rates (the iSCSI architecture is 100 Gbit/s ready today, while native Fibre Channel currently supports a maximum of 32 Gbit/s). We have demonstrated line rate encryption at 100 Gbit/s over the FSP 3000 platform; this data rate is significantly higher than currently enabled by Fibre Channel. Future research will include hosting virtual network functions on the XG210, including firewalls and intrusion detection/prevention.

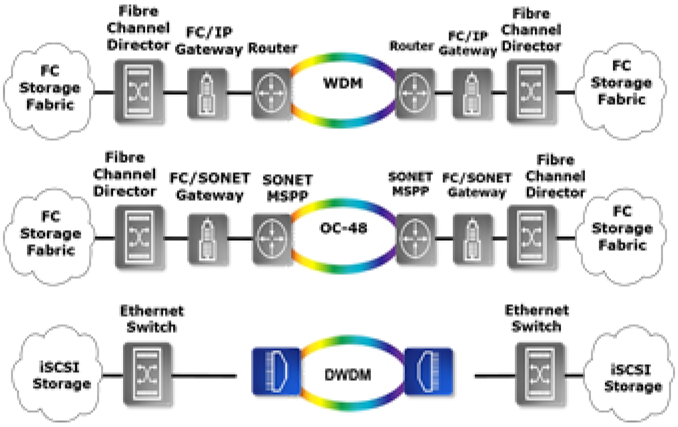

Figure 5. Comparison of network architectures for Fibre Channel over Internet Protocol FC-IP (**top**); Fibre Channel over Synchronous Optical Network FC-SONET (**middle**); and native iSCSI (**bottom**).

5. Intra-Data Center Optical SDN: Affinities

In this section, we discuss applications of SDN to optical networks within a single data center, including affinities for optical mesh SDN and use of the OpenStack Congress API for optical network control. There are many potential sources of affinity data sets, including policies for cloud orchestration and service assurance, as well as data collected from monitoring network traffic and management interfaces. In principle, this data can be analyzed to yield application aware network analytics. However, traditional networks are unable to offer proper support for context-based management, because the related management operations have to be performed manually by network administrators. In fact, conventional networks rely on highly skilled administrators who are able to translate high level policies and application requirements into low level network provisioning and configuration commands while simultaneously adapting to changing network context as rapidly as possible. Limitations in the network management interface often result in only approximate implementations of the desired application requirements. The resulting manual intensive approach has been relatively slow and error prone, prior to the adoption of centralized SDN automation of network control systems. In this section, we will discuss how affinities for optical networks may be realized through an automated SDN reference architecture.

Affinities define linkages or “conversations” between servers, networking, and storage elements, and then outline important features of those linkages. Put another way, affinities provide an open model for expressing application workload requirements in a standard language for data networks, servers, and storage. Affinities differ from other policy constructs such as VLANs and access control lists (ACLs) in that they employ higher level abstractions at the application level. This application level focus is intended to break down traditional management silos between servers, networking, and storage. While traditional data center networks are often based on over-subscribed, hierarchical designs (edge, access, and core layers), this approach is not well suited to distributed workload environments. In such cases, SDN applications may leverage parallel processing and subdivide workloads into smaller tasks (such as done by the Hadoop or MapReduce algorithms [22–24]), running on highly virtualized data center infrastructure. In this environment, most data traffic flows east-west between virtual servers, making it beneficial to flatten the network hierarchy in order to reduce latency and improve performance. Affinities enable a flatter architecture by combining virtual data center resources into “affinity groups”, and dynamically prioritizing connections between these groups depending on their needs (as opposed to providing static, equally weighted connections to all resources, regardless of their application requirements). The resulting optical network performance and scalability are now based on servicing the needs of the affinity groups as a first configuration principle. While the concept of affinities is not new, in order to fully implement affinities we need the ability to dynamically re-provision optical networking equipment in response to application requirements. Conventional decentralized network architectures, with each router providing an embedded, low level management interface, are not well suited to affinity implementations [21]. However, with the advent of SDN optical networks, affinities are enabled as a practical approach to automated network management.

There are many potential benefits of affinities in SDN data center networks. Affinities allow the orchestration layer at the top of Figure 1 to describe workload needs in terms of service level, rather than device specific configurations. The REST APIs shown in Figure 1 allow affinity information to be collected from optical network equipment and incorporated into the orchestration layer, which in turn is programmed to implement different network policies based on the affinity data. In this way, affinities facilitate a self-service, on-demand network resource consumption model which is well suited to cloud service provider traffic patterns. The network infrastructure adapts to application requirements, rather than the other way around. Once optical network resource utilization is automated with SDN, over time resource usage can be optimized around the requirements of different workloads. Architecturally, affinity routing using SDN has potential applications which replace conventional approaches to vSwitch routing as used in vCenter and KVM. Many existing networks deploy some form of source-based MAC/port ID pinning (in which switches pin all traffic from a particular source MAC or vSwitch port to a particular TOR or any northbound uplink). Another common option is Load-Based Teaming (in which traffic is distributed across server NIC uplinks to the TOR based on load, typically updating every few seconds. Affinities can be programmed to supplement or replace either of these functions.

The affinity concept is based on a layered architectural model as depicted in Figure 1, in which SDN controllers dynamically provision data center network infrastructure to satisfy workload affinities, as directed by external applications. The affinity service provides an API to allow controller and higher-level applications to create and share an abstract topology and implementation independent description of the infrastructure needs, preferences and behaviors of workloads that use the network to communicate with each other. This abstraction shields programmers from the complexity of the optical mesh routing algorithms; in other words, developers do not require detailed information about the underlying optical infrastructure. Since there are so many paths between any two nodes on a full mesh optical network, sophisticated proprietary algorithms are used to assign traffic flows to different paths. The affinity routing algorithms have been described elsewhere [20,21], although we will highlight a few key features. The routing algorithms maintain boundary conditions such as loop free topologies with dual redundant path switching for high availability. They also take into account real time state data learned over time from the physical network (as opposed to conventional approaches such as ECMP and hashing). SDN Controller services and controller applications can use the affinity state information to decide how to program data plane devices and can provide troubleshooting and monitoring outputs.

Since SDN is a relatively new concept for optical networks, there are a limited number of network devices which support affinity creation and management. More devices are expected to emerge in the near future, since affinities are supported by open, standards-based SDN controllers such as the Open Daylight API [25]. In our cloud SDN test bed, we implement affinities using four Plexxi 10G Ethernet switches at the forwarding layer. Most data center Ethernet switches employ variations of a spine-leaf architecture. By contrast, Plexxi is one of the few Ethernet switching platforms which implements an optical WDM mesh over relatively short distances (hundreds of meters) within a cloud data center. In our test bed, the four switches are interconnected on a physical ring using 40 Gbit/s WDM inter-switch links to realize a virtual mesh network topology. Affinities to the adaptation layer and above are created by the Plexxi data services engine (DSE), a layer 2.5 network controller appliance. Network policies based on affinities can automatically follow VM live migrations across the network. The Plexxi control plane uses an application resource broker to conduct communications between the SDN network controllers and switches. The private optical WDM network between Plexxi switches enables Layer 1 routing between switches, in addition to traditional Layer 2–3 routed connections. The controller implements dynamic flow-based topologies; every time a new affinity is implemented, the Plexxi controller automatically re-provisions traffic flows to incorporate the new policy without disrupting existing affinities.

The SDN architecture shown in Figure 1 does not show all available OpenStack APIs for the sake of simplicity, however these APIs are still enabled in our SDN solution. For example, OpenStack implements "policy as a service" through the Congress API. We have implemented the first Congress drivers for third party event monitoring and affinity control. As shown in Figure 6, Congress monitors and collects data from other OpenStack API interfaces, including Nova, Cinder, Neutron, and Keystone. Data is stored in tables (analogous to other relational databases) and is accessible through a REST API. We have created code (now released as open source) to provide Congress drivers for switch control planes, capable of extracting network topology information and building linkages or affinities with virtual machine applications. The Congress driver communicates with Plexxi Core, which is the software used to manage Plexxi switches. Affinities between network topologies and VMs can be created by linking the Plexxi and Nova data tables within Congress; a repeated names table is used to identify common features in the switch and server control plane interfaces.

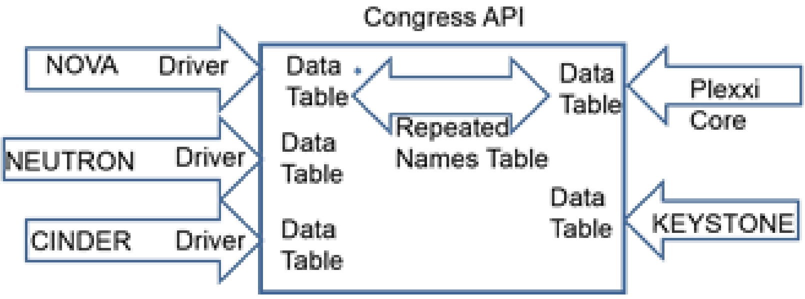

Figure 6. OpenStack Congress driver for Plexxi Core.

The concept of affinities is available in the Open Daylight SDN controller, making it possible to provision network connectivity based on policy definitions in Congress. By implementing Congress drivers which interact with the SDN core network switches, we have demonstrated that the architecture shown in Figure 1 can realize affinities for optical mesh networks within a cloud data center through the Congress API in OpenStack.

6. Multi-Layer Optical SDN

In this section, we discuss multi-layer SDN (the simultaneous management of equipment within and between multiple data centers). We will combine the implementations from Sections 4 and 5 into a common SDN management plane for end-to-end management of optical equipment within the data center, and between multiple data centers. Consider the optical network configuration shown in Figure 3, which provisions network resources on the MAN and within the data center network from a common cloud orchestrator. The orchestrator contains application aware network policies, such as security provisioning. For example, applications such as real time stock trading will have different security requirements than consumer video streaming or a cloud exchange service. The application level security policy includes provisioning, configuration, and management considerations for networking equipment (this policy can also be audited to verify compliance with standards used by the telecommunications industry for exchange services, or standards in other industries such as HIPPA for health care or Sarbanes-Oxley for financial services). These policy requirements are pushed down through the reference architecture stack shown in Figure 1, through the OpenStack Neutron API to the Open Daylight SDN controller, which in turn manages and provisions the underlying networking equipment through their device-specific APIs. In our testbed, it is possible to implement code which interfaces to optical network equipment within the data center, and the optical MAN/WAN, from a common controller. As an example, in addition to the equipment discussed in Sections 4 and 5, we have used the Brocade/Vyatta virtual router/firewall API, within the data center, and the Ciena metro Ethernet equipment between multiple data centers (via the Ciena V-WAN and OneControl APIs). This allows us to provision both devices from a common interface, and to derive requirements from other software appliances. Continuing the security policy example, we have shown that the ACLs which block selected traffic sources for the Vyatta virtual firewall can be derived from data collected on an SSH honeypot configured elsewhere on the network. These security attributes are provisioned on the virtual firewall from the same network controller which manages routers within the data center and the optical MAN/WAN. We could also control the Plexxi and Adva equipment discussed in Sections 4 and 5 from this interface. This does not significantly increase end-to-end provisioning time, which is still on the order of minutes (rather than hours or days, as with traditional management systems). Our management software also enforces flow isolation between multiple tenants, from the server hypervisor across the network within the first data center, across the MAN/WAN, and across the network within the second data center to the target server hypervisor. Within the data center, changes to the forwarding layer will generate monitoring alerts which are propagated back up to the OpenStack Congress interface described in Section 5. Note that at this time, we have only implemented the Plexxi affinity driver for OpenStack Congress; additional Congress drivers will be considered for future work, depending on user demand. Between multiple data centers, multi-tenant virtual slicing can be implemented using the Adva interface described in Section 4; extension of this capability to other vendors will also be considered for future work. Our testbed is capable of running end-to-end network application provisioning, decision making, and flow isolation using the equipment discussed in Sections 4–6, at the same time that we can manage other vendor equipment including the Vyatta and Ciena devices noted previously. This includes the streamlined attachment of storage devices shown in the bottom portion of Figure 5. We have demonstrated re-provisioning of the network policy in a testbed using all of this equipment on a timescale of minutes or less, which is significantly faster than conventional solutions. The entire architecture was exercised, including all layers of the reference design shown in Figure 1.

In one possible use case, multi-vendor, multi-layer network interoperability can form the basis for a cloud exchange service (similar to the telecom exchanges used today). Our work on vendor agnostic virtual network slicing represents a first step in this direction. The WDM platforms used in our test bed are compatible with legacy telecom interfaces such as MPLS providing a migration path from traditional systems to our reference architecture. In the future, the XG210 demarcation point hardware integrated within our metro WDM platforms could potentially host various NFV applications such as firewalls or load balancers, if it could be loosely coupled to the OpenStack management plane. This is facilitated by our work to control both virtual firewalls and optical MAN/WAN equipment from a common control point. The CSP could simplify their service offering and reduce costs by eliminating storage gateways and dedicated SAN switches in favor of the SDN managed iSCSI storage network proposed in Section 4 A multi-CSP cloud exchange would also benefit from a dynamic packet-optical network, which we have demonstrated using the Adva equipment. Cloud exchanges will need to deal with different network traffic flow profiles, including traffic bursts as shown in Figure 2 or high bandwidth, long duration flows between cloud data centers. It may be cost effective to isolate certain traffic flows using SDN provisioning of end-to-end optical links, thus providing additional flexibility in traffic engineering. SDN management of an optical WAN may require dynamic monitoring and provisioning of other network resources, such as optical amplifiers, dispersion compensators, and optical cross-connects; additional research in this area is ongoing [26].

7. Conclusions

We have demonstrated a hypervisor agnostic multi-tenancy optical SDN control framework in a large data centers. Our approach is based on the OpenStack Neutron interface as well as elements of VMWare and KVM environments. We have developed original code to enable dynamic provisioning and virtual slicing of an inter-data center optical WDM network, on a timescale of minutes or less (as opposed to conventional approaches which require days or weeks). Each network slice is able to use a different brand SDN controller to optimize its portion of the network bandwidth. This addresses use cases such as traffic bursts resulting from synchronous storage replication. The reliable delivery of iSCSI storage traffic over an SDN provisioned optical network can potentially eliminate some types of networking equipment. We have also developed a third party driver for OpenStack Congress, enabling policy-based management of intra-data center optical mesh fabrics with affinities. We have also investigated multi-layer SDN control of optical networks from different vendors.

There are several proposed extensions to this interface currently under consideration which would improve our management capabilities. For example, industry standard certification programs for OpenStack are still under development (a recent proposal from the Ubuntu OpenStack Interoperability Lab may be a first step towards addressing this concern). Presently, the orchestration API implementations differ by vendor for each OpenStack release; features which work on one release, for example, may not work properly with a subsequent release, or the same feature from two different vendors may not work the same way even if both are implemented under the same release. Neutron plugins such as Open vSwitch 1.4 use only one core in a multi-core processor to match flows in a flow table; since this does not take advantage of multi-core processing potential, it can result in a potential communication bottleneck. For converged IT and CSP infrastructures, OpenStack management faces challenges including resource scheduling across multiple data centers and traffic bifurcation issues (*i.e.*, management of network traffic which is allowed to use available sub-rate channel capacity, rather than filling the entire available channel bandwidth). Future work will also include cybersecurity concerns for SDN enabled optical networks, including the development of automated network security policies and application aware security provisioning.

Acknowledgement

The authors gratefully acknowledge the support of Marist College and the New York State Cloud Computing and Analytics Center, as well as its industry partners who supported this work, including Adva, Brocade, Ciena, IBM, Lenovo, NEC, Plexxi, and Vyatta.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Gartner Group Reports, "Public Cloud Forecast" (June 2013) and "Private Cloud Matures" (September 2013). Available online: http://www.gartner.com (accessed on 9 October 2014).
2. Shimano, K. "Research Activities for SDN/NFV Technologies on the Photonic Network", Keynote Presentation, Majorca at MIT Workshop, Cambridge, MA, USA, 27–29 July 2015. Available online: http://majorca-mit.org/ (accessed on 27 July 2015).
3. Faw, D. "Intel rack Scale Architecture". In *OIDA Workshop at OFC 2015 & OSA Industry Development Associates (OIDA) Roadmap Report: Photonics for Disaggregated Data Centers*; OIDA and the Optical Society of America: Washington, D.C. USA, 2015
4. Bryce, G.; Aubuchon, G.; Fainberg, M. "Cloud unlocked: Connecting the cloud to provide user value". In Proceedings of the OpenStack Summit, Vancouver, BC, Canada, 18–22 May 2015.
5. Bachar, Y. "Facebook's Next Generation Mega (and Micro) Data Center Technology", Majorca at MIT Workshop, Cambridge, MA, USA, 27–29 July 2015. Available online: http://majorca-mit.org/ (accessed on 27 July 2015).
6. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parklkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. "OpenFlow: Enabling innovation in campus networks". *ACM SIGGCOM Comput. Commun. Rev.* **2008**, *38*, 69–74.
7. Ahmed, R.; Boutaba, R. "Design considerations for managing wide area software defined networks". *IEEE Commun. Mag.* **2014**, *52*, 116–123.
8. Fernando-Palacias, J.P. "Multi-layer, multi-domain SDN at Telefonica". In Proceedings of the OFC 2015 Annual Meeting, Los Angeles, CA, USA, 19–24 July 2015.
9. Yeganeh, S.H.; Ganjali, Y. Kandoo "A framework for efficient and scalable offloading of control applications". *Proc. HotSDN* **2012**, *2012*, 19–24.
10. Koponen, T.; Casado, M.; Gude, N.; Stribling, J.; Poutievski, L.; Zhu, M.; Ramanathan, R.; Iwata, Y.; Inoue, H.; Hama, T.; *et.al.* "Onix: A distributed control platform for large scale production networks". In Proceeding of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010), Vancouver, BC, Canada, 4–6 October 2010; pp. 1–6.
11. Tootoonchian, A.; Ganjali, Y. Hyperflow: "A districuted control plane for OpenFlow". In Proceedings of the 2010 Internet Network Management Conference Research on Enterprise Networking, Ser INM/WREN 2010, Berkeley, CA, USA, 27 April 2010; pp. 3–13.
12. OpenStack: Open Source Software for Building Public and Private Clouds. Available online: http://www.openstack.org (accessed on 9 October 2014).
13. Manville, J. "The power of a programmable cloud". In Proceedings of the OFC 2012 Annual Meeting, Anaheim, CA, USA, 18–22 March 2013.
14. DeCusatis, R.C.; Hazard, L. "Managing multi-tenant services for software-defined cloud data center networks". In Proceedings of the 6th Annual IEEE International Conference on Adaptive Science & Technology (ICAST 2014), Covenant University, Covenant, Nigeria, 29–31 October 2014.

15. DeCusatis, C. "Value and cost of multi-layer SDN". In Proceedings of the OFC Service Provider Summit, Los Angeles, CA, USA, 22–26 March 2015.
16. DeCusaits, C.; Sher-DeCusatis, C.J. "Dynamic Software Defined Network Provisioning for Resilient Cloud Service Provider Optical Network". In Proceedings of the International Conference on Computer and Information Science and Technology, Ottawa, ON, Canada, 11–12 May 2015.
17. DeCusatis, I.; Marty, R.; Cannistra, T.; Bundy, Carolyn J. Sher-DeCusatis, "Software defined networking test bed for dynamic telco environments". In Proceedings of the SDN & OpenFlow World Congress, Frankfurt, Germany, 22–24 October 2013.
18. Cannistra, B.; Carle, M.; Johnson, J.; Kapadia, Z.; Meath, M.; Miller, D.; Young, C.; DeCusatis, T.; Bundy, G.; Zussman, K.; *et al.* "Enabling autonomic provisioning in SDN cloud networks with NFV service chaining". In Proceedings of the OFC Annual Meeting, San Francisco, CA, USA, 10–14 March 2014.
19. Kandula, S.; Segupta, A.; Greenberg, P.P.; Chakin, R. "The Nature of Data Center Traffic: Measurements and Analysis". In Proceedings of the IMC 2009, Chicago, IL, USA, 4–6 November 2009. Available online: http://research.microsoft.com/pubs/112580/imc09_dcTraffic.pdf (accessed on 5 June 2015).
20. Mathews, M. "The Road to the Open Network", IEEE ComSoc SCV (January 2014). Available online: http://www.comsocscv.org/docs/20140108-Mathews-Plexxi.pdf (accessed on 5 June 2015).
21. Plexxi White Paper, "Affinity Networking in an SDN World", SDX Central. Available online: http://www.sdxcentral.com/wp-content/uploads/2013/04/Affinity-Networking-in-an-SDN-World-FINAL.pdf (accessed on 5 June 2015).
22. "A Benchmarking Case Study of Virtualized Hadoop Performance on VMWare VSphere 5". 2011; pp. 1–17. Available online: http://www.vmware.com/files/pdf/VMW-Hadoop-Performance-vSphere5.pdf (accessed on 21 July 2015).
23. MacDougall, R.; Radia, S. "Hadoop in Virtual Machines", Proc. Hadoop Summit, June 2012 Available online: http://www.slideshare.net/rjmcdougall/hadoop-on-virtual-machines (accessed on 22 July 2015).
24. "Virtual Hadoop", Hadoop Wiki. Available online: http://wiki.apache.org/hadoop/Virtual%20Hadoop (accessed on 21 July 2015).
25. The Linux Foundation, Open DayLight Collaborative Project. Available online: http://www.opendaylight.org/ (accessed on 4 September 2014).
26. Birand, B.; Wang, H.; Bergman, K.; Kilper, D.; Nandagopal, T.; Zussman, G. "Real-time power control for dynamic optical networks—Algorithms and experimentation." In Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP'13), Göttingen, Germany, 7–10 October 2013.