Journal of Network and Computer Applications 94 (2017) 33–49

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Enabling network innovation in data center networks with software defined networking: A survey

Bin Dai[a,*], Guan Xu[a], Bengxiong Huang[a], Peng Qin[b], Yang Xu[c]

[a] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China
[b] China Academy of Electronics and Information Technology, Beijing, China
[c] School of Electrical and Computer Engineering, New York University, New York, USA

ARTICLE INFO

Keywords:
Software defined networking
Data centers
Automatic configuration
Network security
Resource allocation
Energy saving

ABSTRACT

With the rapid growth of cloud applications and users, Data Center Network (DCN) has become a critical component in the cloud ecosystem to sustain remarkable computation demand. It is facing great challenges in performance guarantee, security enforcement, and resource and energy management. Software Defined Networking (SDN), as an efficient way to collect network information and perform network management, has been adopted in DCN to enable automated network configuration and management. In this paper we aim at surveying the state-of-the-art techniques of using SDN in DCN and discuss the way SDN help DCN. We first present the introduction of SDN and DCN, then survey the SDN based DCN, and finally show some lessons and future trend.

1. Introduction

A data center is an industrial computing service infrastructure, with a facility to house computer systems and its associated components, such as storage devices, power supplies, communication devices, and security devices. It provides a cost-efficient way for companies and personal tenants to rent a slice of computation and communication resource to meet various requirements (Chen et al., 2013). A typical data center contains tens to hundreds of thousands of servers. For example, Google data center has been reported (Google architecture, 2008) to host more than 900,000 low-cost commodity servers in 2011, and a recent report (A. G., 2015) from IT consulting firm Anthesis Group infers that there are about 33 million physical servers deployed inside data centers around the world until 2015.

A fundamental challenge on Data Center Network (DCN) is to efficiently manage the huge number of network elements (Chen et al., 2013) to meet tenants' demands, and build DCN with low-cost commercial off-the-shelf hardware (Pries et al., 2012). According to literatures, there are many critical issues on DCN to achieve automatic configuration, security, resource allocation, energy saving and DCN virtualization (Dayarathna et al., 2016; Kachris and Tomkos, 2012). The DCN connection issues concern the bi-section bandwidth, which depends on the network structure. Innovated DCN structures, such as fatTree (Al-Fares et al., 2008) and VL2 (Albert et al., 2009), can achieve high bisection bandwidth by building rich connected topology with redundant switches and links (Chen et al., 2013). As the communication bandwidth demand of data center applications increases dramatically, the excessive power consumption is becoming another challenging issue in the design and deployment of a data center (Kachris and Tomkos, 2012). The traditional DCN architectures are not flexible enough to support DCN virtualization applications with QoS, deployability, manageability, and defense against security attacks (Bari et al., 2013).

To enable academic experiments on production networks without influencing the production traffic, Software Defined Networking (SDN) and OpenFlow are proposed as a novel network architecture (McKeown et al., 2008) with the network control decoupled from data forwarding. In the SDN architecture, the experiment rules are designed to handle experiment traffic and avoid influencing the production traffic on the same production network, as the rules on the flow tables of SDN switches can be instructed by the SDN controller. Besides OpenFlow, There are other SDN solutions including Cisco Open Network Environment (ONE) and Nicira Network Virtualization Platform (NVP). Cisco ONE is the DCN infrastructure solution that can solve challenges in both DCN computing and network management. For DCN network management, Cisco ONE can bring DCN greater agility and fast application delivery. For DCN infrastructures on Cisco Nexus Series devices, Cisco ONE provides foundation for Networking and advanced applications through Data Center Fabric. They focus on a scalable, resilient, high-performance physical and virtual network that

* Corresponding author.
E-mail address: nease.dai@gmail.com (B. Dai).

http://dx.doi.org/10.1016/j.jnca.2017.07.004
Received 4 April 2017; Received in revised form 2 June 2017; Accepted 4 July 2017
Available online 06 July 2017
1084-8045/

provides workload mobility across multiple data centers. They propose a business-relevant SDN policy model across networks with Cisco Application Centric Infrastructure (Cisco ACI) to reduce Total Cost of Ownership (TCO), automate IT tasks and accelerate DCN application deployments. Nicira NVP can create an intelligent abstraction layer between virtualized hosts and an existing physical network. NVP is managed by a distributed controller system.

There is an increasing requirement of innovated networking technologies to meet the dynamic computing and storage demand. Firstly, the DCN applications commonly access databases and servers distributed in both public and private clouds, which requires flexible traffic management and enough bandwidth to guarantee the performance of transmission. Secondly, more and more users are accessing the databases and servers by their own devices as well as the working IT devices, which requires flexible networking technology to keep mobile devices online, and also requires security technology to detect malicious users. Thirdly, the parallel processing in DCN requires bandwidth for synchronization between distributed servers. The traditional network is constrained to provide dynamic and complex network rules, the large-scale parallel processing algorithms, and a standard and open interface for network devices.

To meet these DCN requirements, SDN has been used to implement network appliances such as load balancer, traffic engineering, in-network caching, malicious identification, access control policies enforcement, and dynamic resource allocation. As a free and open-source software platform, OpenStack has been exploited by both public and private cloud. And with the increasing scale of high-density and multi-tenancy cloud environment, the cloud operator and the tenants need to move, add or remove workload between servers or even between different clouds on the fly for new requirements, such as creating private networks, controlling the IP addressing, and additional network services. Therefore the networking services are redesigned with SDN to meet these new requirements. With the northbound APIs of the SDN controller, the cloud operator and tenants can program flow tables of hardware and software switches for flexible network services with lower cost and higher efficiency, implementing functions such as creating and configuring private networks, compliance & audit mandates, QoS, monitoring and troubleshooting, firewall, intrusion detection and Virtual Private Network (VPN). VMware provides an industry-leading Software Defined Data Center architecture to implement a unified platform for public, private and hybrid clouds, where cloud operators and tenants can rapidly develop, automatically deliver and manage all network applications. Their best-in-class virtualization product, VMWare vSphere, has been significantly improving IT efficiency and performance of worldwide enterprises networks. However the growing field of mobile cloud poses new challenges to infrastructure services of providing IT and networking resources in DCN. For these challenges, they extend the virtualization concept to all DCN resources and services with SDN. The DCN resources virtualization services deliver abstraction, pooling and automation of the compute, network, and storage infrastructure. And the provisioning and ongoing management of all these virtual resources are enabled by the policy-driven automation, with the advantages in reducing both capital expenditures and operating costs, accelerating application deployment and expansion, increasing the granularity of security and compliance for every application, and increasing the flexibility of customized cloud platform on various hardware, hypervisors, and clouds.

For DCN architecture, the SDN can deploy the network appliances flexibly to provide the application performance and security policies can be enforced by the security services on SDN controller applications. Since the first Data Center with SDN (DCSDN) (Tavakoli et al.,) has been attempted early in 2009, more and more novel studies have been proposed to integrate the DCN and SDN structures. Firstly, automatic configuration with SDN can automatically configure and re-configure the network to fulfil tenants' traffic (Liu et al., 2013a, 2013b). Secondly, middleboxes with SDN can be designed for network security (Lara et al., 2014). Thirdly, SDN based scheduling of flow and link workload is promising for future networks (Sharkh et al., 2013). Fourthly, the energy consumption of data centers is non-trivial, including switching/transmitting data traffic, storage/computation in DCN servers, cooling systems, and power distribution loss (Li et al., 2014). In this paper, we focus on the DCN issues that have been solved with SDN, and we divide these issues into automatic configuration, network security, network resource allocation, and energy saving. To the best of our knowledge, we are the first to survey DCSDN comprehensively. The remainder of this paper is organized as follows. Section 2 reviews the up-to-date surveys on SDN and DCN. Section 3 looks into the main improvement SDN has brought to DCN. Section 4 presents details of resource allocations. Section 5 lists promising aspect of DCSDN for future works. At last, we present the conclusion in Section 6.

2. Related work

There are excellent surveys investigating the state-of-the-art in SDN (Lara et al., 2014; Nunes et al., 2014; Hakiri et al., 2014; Scott-Hayward et al., 2013) and DCN (Chen et al., 2011; Bilal et al., 2014; Dayarathna et al., 2016; Ranjana and Raja, 2013; Jain and Paul, 2013; Azodolmolky et al., 2013; Zhang et al., 2013). For example, Hakiri et al. (2014) discussed the value of SDN in cloud based networks. They stated that SDN brings complementary network technology to DCN, and promotes the intelligence in network virtualization, automating resource provisioning and creating new services on top of the provisioned network resources. Also, they summarized SDN based interactive solutions in DCN that enable cloud services and applications to retrieve network topology and network failures for an optimized initialization and adjustment of network connectivity/tunneling. Wang et al. (2015) surveyed technologies for cloud DCN, including topology design and network technique of DCN, virtualized network elements, and routing for physical and virtual network elements. SDN is promising in providing intelligence in the control plane and traffic flexibility in the data plane for cloud DCN. As use cases for cloud computing, SDN can offer flexible routing scheme, reduce switching consumption, improve the performance of VLAN-based communications in various traffic management issues. Zhang et al. (2013) surveyed the transport layer challenges, such as latency and malicious attack in virtual data centers over multiple DCNs. They focus on the problems of TCP incast for many to one traffic pattern, the large latency of online queries and TCP performance deterioration for Visualized DCNs, and malicious user defense in multi-tenant DCNs.

As the cloud services become increasingly important, security becomes another important research issue. Yan et al. (2016) studied the DDoS attack problem in distributed DCNs, and surveyed some SDN based solutions. Rahman and Choo (2015) surveyed existing incident handling and digital forensic literature to fill the knowledge gaps in handling incidents in the cloud DCNs. Juliadotter and Choo (2015) presented a conceptual cloud attack taxonomy that follows the nature flow of an attack on a cloud service, divided into 5 dimensions, i.e. source, vector, target, impact and defense. With this taxonomy, they quantified the risk by rating values from 1 (low rise) to 9 (high risk), that can be used to estimate the likelihood and impact of a given attack. Subashini and Kavitha (2011) surveyed security issues in the stacks of different cloud service delivery models, i.e. SaaS, PaaS and IaaS, where the security issues can be classified into data storage security, data transmission security, application security and security related to the third party resources.

SDN promotes network virtualization in cloud DCNs, which has been studied in several surveys. Azodolmolky et al. (2013) compared SDN-based virtual networking implementations in cloud DCNs. Jain and Paul (2013) presented network virtualization methods in cloud DCNs, and pointed out the way the SDN architecture can facilitate network virtualization. The survey (Bari et al., 2013) studied various methods of network virtualization to support QoS, deployability, manageability, and defense against security attacks in virtualized DCN from the metrics of management flexibility, cost, scalability, resources utilization, and energy efficiency. The survey in Belbekkouche et al. (2012) focus on the resource discovery and allocation in network virtualization. The resource allocation methods are surveyed in categories including central control and distributed control. Also, they surveyed resource allocations from perspective of Virtual Network (VN) update (reconfiguration), reliability (survivability), and wireless networks. Besides, they listed the architecture characters of network virtualization, including coexistence, recursion, inheritance, revisitation, flexibility, manageability, scalability, isolation, stability, and heterogeneity. Bist et al. (2013) surveyed existing open source IaaS with the category of public, private and hybrid cloud. Blenk et al. (2016) surveyed various hypervisors for network virtualization in cloud DCNs, including centralized hypervisors and distributed hypervisors. The centralized hypervisors are based on the seminal hypervisor, FlowVisor, and can be divided into hypervisors for general networks and special networks (including policy-based hypervisors).

Chen et al. (2011) surveyed the DCN routings from the aspects of DCN specified topologies, traffic patterns, and security. The DCN topology requires more customized routing with automatic and scalable assignment of DCN server addresses, rather than general internet routings, such as OSPF/ISIS or Border Gateway Protocol (BGP). The different traffic patterns in DCN for different type of DCN applications demand DCN Traffic Engineering(TE) for routing reliability, load-balancing, and energy-efficiency. Also they surveyed routings for the traffic patterns of group multicast, which is popular in DCN.

Energy efficiency is important for DCNs, Ranjana and Raja (2013) surveyed energy efficient VM placement methods in DCN, focusing on optimizing the server energy consumption. Bilal et al. (2014) presented a survey of DCN energy that focuses on the energy consumption of network devices rather than computational components. Dayarathna et al. (2016) presented an in-depth survey of the data center energy consumption with related literatures organized according to levels of a data center system architecture, i.e. the digital circuit level, server level, processor level, memory and storage level, data center level, software level, energy consumption modeling level. Hammadi and Mhamdi (2014) presented a similar survey on energy efficiency DCN, with more references and a more detailed catalog of energy efficient methods for DCNs. Cavdar and Alagoz (2012) surveyed energy efficient methods in DCN from aspect of computing and networking. They also refer to several works on virtualization and cloud that can consolidate resources to increase energy efficiency.

3. Existing applications in DCSDN

In this section we summarize the main contributions on DCSDN (shown in Fig. 1), from its benefits to SDN application in DCN to network optimizations. The SDN framework provides a uniform control plane to configure and re-configure network automatically and intelligently. The network security functions can be designed by deploying middleboxes in multiple SDN switches, which provides programmable interfaces in SDN architecture. Furthermore, TE in DCSDN can be formulated with objective of network resource optimization and sub-objectives, such as network energy saving.

SDN provides innovations of automatic and dynamic network configurations for data center applications in and across multiple DCNs. The SDN technology provides flexible and scalable functions for DCN specified networks (Tavakoli et al.,). The SDN framework supports non-SDN data center networking proposals, such as PortLand (Niranjan Mysore et al., 2009) and VL2 (Albert et al., 2009). As the DCN requires dynamic VM migration for a flexible tradeoff between cost and performance, the SDN based rerouting method can choose the ideal switches and deploy redirection rules (Mann et al., 2013) for online flows, as the SDN controller monitors flows and the global network view. To migrate flows without service interruption, network configuration primitives with SDN have been proposed with visibility and live orchestrate of DCN network resources (Liu et al., 2013a, 2013b; Khurshid et al., 2013).

For virtual data centers built over existing multiple DCNs, network configuration with cooperation between multiple DCNs can be implemented by SDN to reduce complexity (Luo and Chen, 2013). Large scale and geographically distributed services are always deployed in a set of DCNs with service performances optimized by offline or live VM migration between racks. Various SDN controller applications have been implemented for layer address mapping, ARP resolution and location updating in DCN (Hao et al., 2009; Pisa et al., 2010; Boughzala et al., 2011; Mann et al., 2012).

Network security in DCSDN includes topics such as traffic isolation between tenants' virtual DCN, traffic filtering and intrusion detection and access control. SDN switches can be programmed for special actions with tenants virtual networks, such as packet header rewriting for traffic hiding (Nunes et al., 2013) and MAC learning (Kawashima and Matsuo, 2013), multiple layers of MPLS labels (Kempf et al., 2013) for scalable security, bandwidth isolation (Shimonishi et al., 2013) for virtual network with VxLAN. With SDN switches supporting specific flow entries, general security middleboxes with network functions such as intrusion detection (Hu et al., 2013) and access control (Iyer et al., 2014) can be implemented by SDN.

Network resource allocation in DCSDN can fully utilize link resources with network global visibility. Considering the network resource controlled components, existing resource allocation methods in DCSDN can be divided into host-based control, network based control, and hybrid control. In host-based control methods, end hosts in DCN are controlled according to the network resource awareness provided by SDN. There are host-based control methods in optimized TCP (Ghobadi et al., 2012; Jouet and Pezaros, 2013), max-min fairness (Popa et al., 2011, 2013), and pre-served bandwidth barging between VMs (Guo et al., 2013; Guo and Liu, 2014). Considering the characters of DCN services, VM can be migrated to a better server with more network, storage and computing resources (Benson et al., 2011; Liu and Jin, 2013). In network based control methods, SDN controller describes heterogonous network resources in a uniform way, making it possible to optimize and schedule the connectivity, bandwidth and latency for tenants networks. Generally network resources can be dynamically and actively orchestrated by Breadth First Search (BFS) resource bookkeeping algorithms (Rosa et al., 2014) and VPN based virtual link re-configuring methods (Baucke et al., 2013). Besides, there

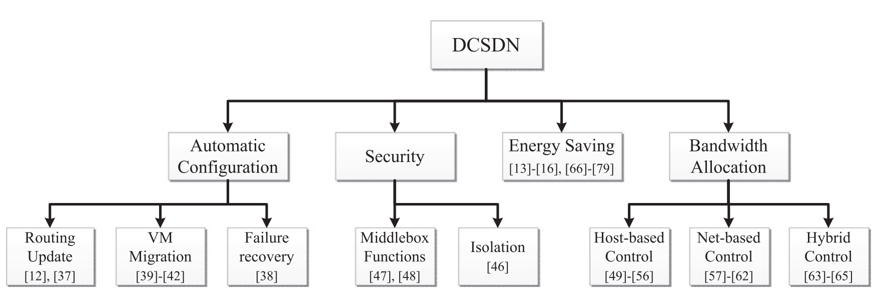


Fig. 1. Taxonomy of SDN Applications in DCSDN.

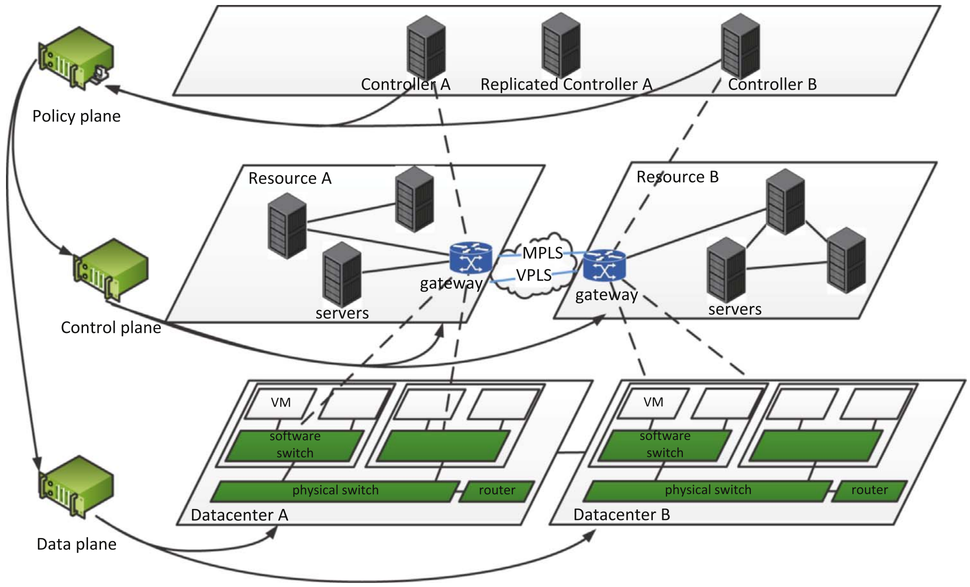


Fig. 2. A network abstraction framework: Police plane abstracts the multiple SDN controllers of each DC, control plane abstracts the inter-DC connections, and data plane abstracts intra-DC links of each DC. Virtual networks across multiple DCs can be abstracted as a single DC and routed according to existing schemes within single DCN.

are some simplification on the controller schedule (Al-Fares et al., 2010; Bi et al., 2013; Curtis et al., 2011), and accelerations with high performance hardware (Ke et al., 2013). In end-network control methods, flows can be controlled on both end hosts and NE to meet the demands of end hosts. With NEs monitoring the traffic, the SDN controller can optimally schedule and route all DCN flows by TE methods (Jain et al., 2013; Hong et al., 2013; Zhao et al., 2013).

As the DCN are designed with scalable pooled resources, more and more literatures are considering the energy efficiency of the pooled, network aware methods (Biran et al., 2012; Meng et al., 2010; Dias and Costa, 2012; Georgiou et al., 2013; Lara et al., 2014; Sharkh et al., 2013; Li et al., 2014; Nunes et al., 2014) energy aware methods (Beloglazov et al., 2012; Lago, 2011; Srikantaiah et al., 2008; Ebrahimirad et al., 2015) and network and energy aware methods (Kliazovich et al., 2010, 2013; Pascual et al., 2015; Wang et al., 2012; da Silva and da Fonseca, 2016; Mann et al., 2011). For network aware methods, VM traffic and VMs are placed on a small set of switches and servers to minimize the network workload in the top-level switches. For energy aware methods, the network is simplified with no constraints, optimizes VM placement to minimize the server energy. For energy and network aware methods, the network aware character favors grouping VMs with higher inter traffic into nearby servers, and the energy awareness determines the server selection, that is to consolidate VMs into a small set of servers for higher energy efficiency. Besides, the VM placement methods can be further enhanced with flow scheduling and aggregation. The flow scheduling will control both the application and network DCN, determining the flow transferring time of each application (Li et al., 2014). The flow aggregation methods consolidate flows into a minimal subnet (Heller et al., 2010) to minimize the energy consumption of switches.

3.1. Automatic configuration in DCSDN

3.1.1. Problem

With the increasing portion of hybrid cloud of public and private clouds, there are challenges in network configuration for a large number of DCN services such as VM migration, network load balance, and the increasing requirement of network security and reliability such as switch failure repair, switch firmware upgrade and new switch onboarding. In VM migration operations, the DCN operators will choose a new address for the migrated VM from the pooled network addresses. This raises challenges in dynamic network configurations for lossless migration of both VMs and VM flows. In hybrid clouds, the migrated VMs with new IP addresses will bring challenges to address mapping and redirecting routing rule updating for lossless migrations. The network load balancer needs to orchestrate flows with dynamic and intelligent configuration and reconfigurations, especially for the unpredictable DCN traffic of different tenants. In typical DCNs, the operators need to update the network devices frequently for switch firmware upgrade, new switches onboarding and link failure recovery. The significant growth of cloud tenants and their huge demands for the pooled network resources raise a number of concerns with respect to the disruptions to the applications.

With SDN architecture embedded into DCN architectures, the SDN can facilitate the automatic configuration for variable, vast volume, and unpredictable traffic in DCN. The logically centralized controller can manage flow entries on each SDN switch directly, and deploy application-level polices with consistent configurations efficiently. Even for unpredictable DCN traffic, the SDN architecture can be programmed to design and update seamless flow policies dynamically and efficiently.

3.1.2. SDN based solutions

Seamless Virtual Network. The DCN provisioning has to maintain a seamless addressing system to migrate VMs across different networks. The SDN controller can abstract physical and virtual NEs across multiple DCNs, and treat them as a single virtual core switch with virtual components of ports, links and policies (Luo and Chen, 2013). The virtual ports are NE ports connecting to DCN, with the NEs treated as linecards for the virtual core switch connected by inter-NE connections. The virtual links are connections between servers and NEs inside a physical DCN. The virtual policies are applied to virtual networks, supporting mobile NEs. Fig. 2 shows the network abstraction of the virtual networks across multiple data centers with SDN. The police plane is a logically centralized control framework with multiple controllers that computes route on gateway switches, installs flows into switches, and enforces policies throughout the system. The control and data planes are designed to abstract connections between DCN gateways and links between physical and virtual switches inside DCNs. With this framework, DCSDN can automatically configure and reconfigure switches with rules for VM migrations, routing updating and failure recovery routing. For inter virtual DC VM live migration, the VM flows are also migrated to the new location online by congestion-free update of flow entries. The Virtually Clustered Open Router (VICTOR) (Hao et al., 2009) is a big virtual router abstracted from the SDN supported NEs distributed in different DCNs, enabling the inter-DC VM migration without modifying user applications or TCP/IP stack, as the traffic between DCs can be forwarded in an IP-in-IP tunnel. In VICTOR, a SDN controller is programmed to handle signaling and routing for intra and inter-DC VM migration. However, when VMs are migrated between heterogeneous DCN architectures, such as VL2 and Portland that have different addressing and routing schemes, it is not easy to maintain connections online, as they locate VMs by pseudo IP and pseudo MAC address separately. To address this problem, Boughzala et al. (2011) was proposed to handle VM migrations between DCNs with different network addressing rules. It consists of two-level rules, the global level for the network structure and the special level for the address mapping in different DCNs, such as in VL2 and Portland. The prototype CrossRoads (Mann et al., 2012) was proposed for the seamless online and offline inter-DC VM migration with consideration of coordinating multiple SDN controllers in each DCN. The locations of migrated VMs are denoted by both pseudo MAC and pseudo IP address, and used to configure the connections of migrated VMs across DCs.

As a scalable addressing architecture, the Locator/ID Separation Protocol (LISP) (Farinacci et al., 2015) proposed by Cisco reorganizes a network address by two separate address spaces: the EID address in virtual address space, and the Routing Locator (RLOC) address in physical network address space. With this architecture, a virtual node in DCN can migrate to new locations with different RLOC addresses, as the LISP mapping system services will maintain the mapping between EID address and current RLOC address. LISP can be implemented as a southbound API in SDN architecture, and the relation between LISP architecture and the SDN paradigm is studied in Rodriguez-Natal et al. (2015). LISP can act as a southbound API in SDN for the underlay LISP network elements, i.e. LISP Tunnel Routers (LISPTRs), as the Mapping System in LISP can update the LISPTR state from a global view and response to mapping queries of the Tunnel Routers. The internal function of the Mapping System can be designed on demand with scalabilities (Jakab et al., 2010). The SDN policies can be deployed on LISPTRs for two reasons, first the LISPTRs can act as intermediate routers for other LISPTRs, decapsulating, re-encapsulating and forwarding traffic to the next hop LISPTR, according to the instructions from the Mapping System. Second, an EID maps to different LISPTRs with priority/weight attributes, which supports programmable deployment in the Mapping System.

Conflict-Free Updating of Flow Rules. To update network traffic without congestions, routing loops, or routing black hole, network rule updating procedure can be scheduled by the central controller. The zUpdate (Liu et al., 2013a, 2013b) strategy is proposed to avoid traffic congestions in updating configurations of traffic distribution, as the configuration update may lead to congestion in the transient scenarios as shown in Fig. 3. The zUpdate controller can compute a lossless transition plan, breaking all updates into sets of intermediate updates without congestion. The Veriflow (Khurshid et al., 2013) is implemented as an agent abstraction of switches, modeling network behavior and checking routing invariants, such as routing loops and routing black holes. The network behavior can be built by classifying existing rules by prefix trees denotation and generating a forwarding graph. Then the invariants of flow rules can be detected by basic reachability algorithms, which would traverse every forwarding graph using Depth First Searching (DFS). After the detection, the violated invariant rules would be dropped or installed with alarms.

Alternative Routing Rules deployment. In order to recover from network failures, recovery schemes have been proposed to shift flows into valid paths. With the programmable data plane, SlickFlow (Ramos et al., 2013) prepares recovery paths by computing alternative paths for each link and shifting to alternative paths automatically upon a link failure. The alternative paths are embedded into the header of packets, so that the SDN switches detecting link failure of the primary path can switch packets to the alternative path. For each link of the primary path, an alternative is determined proactively by the controller. The prototype built on Open vSwitch can be effectively used in Mininet with various typical topologies, such as fat-tree, Bcube, and Dcell topologies for DCN. Although this method can make use of the DCN character of rich path diversity, there are two problems. Firstly the load of flow table size would be twice as before, and secondly alternative paths are embedded into packet header, which would increase the network traffic volume and power consumption. The Network Copy (NCP) (Mann et al., 2013) purely handle the flow replication between multiple DCNs, with OpenFlow controller and middlebox suppliant. The controller runs a simple lowest common ancestor (LCA) algorithm to calculate the minimal path from the primary path to a given NCP middlebox server, which will redirect the flow to the replica host. However, NCP middleboxes are needed to capture and order redirected packets to reconstruct the session from the target of replication. Since OpenFlow switch version 1.4 (Foundation, 2013), OpenFlow switches support quick and local reaction to failures without the need to resort to a central controller, named the OpenFlow fast-failover mechanism. However, only selected OpenFlow switches of a flow's path can reroute the flow on failure detection, for a global optimization. In order to bring less modification to existing networks, Capone et al. (2015) proposed an MPLS tag based method to reroute packets to reroute switches. Once an OpenFlow switch detects node or link failure, it will tag the packet and send it back along the primary path, and then the first OpenFlow switch that can reroute the packet will reroute the packet and change the flow tables to reroute the following packets from the source node.

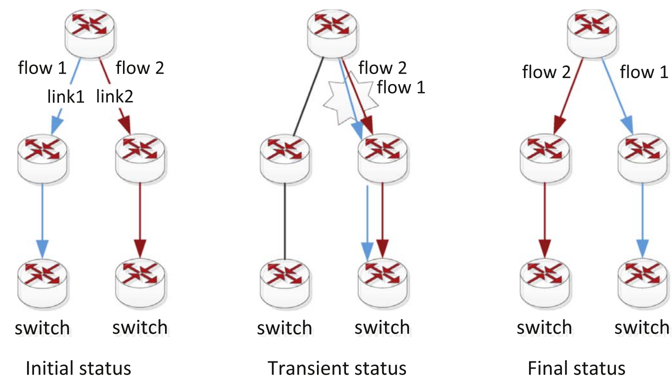


Fig. 3. An example of link load burst in network configuration. In this scenario, flows $f_1$and f_2 are initially routed with output links link 1 and link 2, respectively. With an update, $f_1$and f_2 exchange their output links. However, f_1 is firstly updated in the transient stage, increasing the load of link 2 by f_1.

3.1.3. Discussion

For the above DCN issues on automatic configuration, the innovative SDN paradigm makes it easy to track and maintain the configurations of dynamic applications, flows, and NEs. So the SDN based automatic configuration can build a stable infrastructure for DCN. We compare them in Table 1, which shows different aspects of automatic configuration, including the application scenario, the input and output of the innovations. They are all built on the basic characters of dynamic update, programmable central control plane, and global network topology of SDN.

3.2. Network security in DCSDN

3.2.1. Problem

The traditional network security devices of firewall, Intrusion Detection System, and Access Admission Control are challenged in

Table 1
Comparison of automatic configurations for DCSDN.

Publication	Applications	Input	Output	Centralized
zUpdate Liu et al. (2013a, 2013b)	Lossless update	Traffic matrix, traffic distribution	Rule order for transient update	Yes
Veriflow Khurshid et al. (2013)	Lossless update	New rules, existing forwarding graphs	Detected rules collision	Yes
Luo and Chen (2013)	Standard abstraction virtual network	Primitive of policies	Controller functions	Yes
VICTOR Hao et al. (2009)	VM migration	Inner-DC, inter-DC destination	Selected OF switches	No
Boughzala et al. (2011)	VM migration	Required resources description	Specific rules	No
CrossRoads Mann et al. (2012)	VM migration	New location of VM	Reconfiguration for live migration	No
Mann et al. (2013)	VM replication	Traffic distribution, place of middleboxes	Selected redirection switch	Yes
SlickFlow Ramos et al. (2013)	Redundant source routing	Feasible paths	Extend packet header	Yes

the cloud tenants' virtual networks, where the tenants decide their services and virtual networks with dynamic virtual connections and boundaries. In DCN, the data provenance problem (Buneman et al., 2001) has been solved by network accountability and forensic analysis, which detects the faulty nodes that may be compromised or simply malfunctioning. Typical tools are designed to monitor network activities by analyzing the event logs on each DCN node. However, the faulty nodes may coordinate their lies to avoid detection, which is a challenge to data provenance (Bates et al., 2014).

With the rapid rise of services on top of virtual machines and virtual network rented from Cloud DCN, DC clients (tenants) bring new requirements and challenges to DCN. To prevent tenants' traffic from accidental or malicious actions from other tenants, existing isolation methods, such as VLANs and Layer 3 routing can be exploited, however they both limit the network scalability and the network operability (Nunes et al., 2013). For example, the VLAN protocol supports no more than 4094 VLAN tenants as the length of the VLAN ID space is 12 bit according to IEEE 802.1Q protocol. The L2-in-L3 tunneling protocols, such as VxLAN (Mahalingam et al., 2015), NVGRE (Wang and Garg, 2015), and STT (Davie and Gross, 2016) can be used to increase the tenants ID space to 24 bit. However, when a Layer 2 frame is encapsuled into a Layer 3 packet, the packet length will easily extend the traditional path MTU limit and drive IP fragmentation operations.

3.2.2. SDN based solutions

SDN has been stimulating innovations in security utilities (Lara et al., 2014), such as traffic isolation, covert channel protection, source address validation and DDoS attack detection. In this section, we show the solutions of network security issues in DCSDN, which focus on traffic isolation, security middlebox substitution, and flow table protection respectively. We classify the SDN solutions by the DCN security requirements, including traffic isolation, security middleboxes for firewall, Intrusion Detection System, Access Admission Control, and DDoS. For traffic isolation, tenant's virtual networks need to be invisible to other tenants. With security middlebox substituted by SDN switches, network functions can be flexibly deployed. Flow table protection can detect the DDoS attack and preserve the limited capacity of flow entries in each SDN switch.

Traffic isolation. For traffic isolations, customizable SDN switches have been designed to substitute middleboxes in DCSDN to rewrite MAC addresses and insert VLAN ID and MPLS labels. In DCPortals (Nunes et al., 2013) system (shown in Fig. 4), VMs can send packets to each other through the core network, with MAC address transformed at the OVS inside the host. For example, when VM1 hosted in host1 sends a packet to VM2 on host2, it will change the packet fields by replacing the source and destination MAC address by the MAC address of host1 and host2, and the OVS of host2 will turn the packet to the original one. The rewriting flow entries are only installed in OVS of the physical servers, without bringing extra pressure to the physical switches. To achieve address isolation and scalability of multi-tenant virtual DCN networks, Kawashima et al. utilized packet header rewriting on SDN virtual switches for tenant networks (Kawashima and Matsuo, 2013) with the real MAC address of destination VM replaced

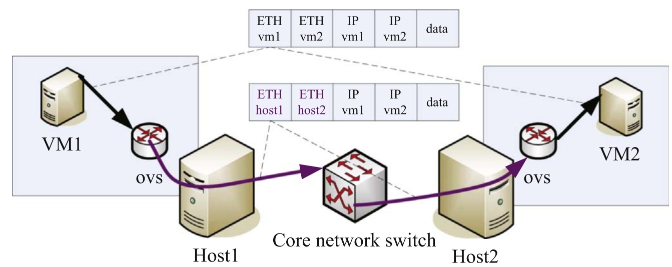


Fig. 4. DCPortals with packet header rewriting.

by a two-tuple of MAC address of the destination server and a host-based VLAN ID (VID). The SDN controller maintains mappings between the two-tuple addresses and the real MAC addresses of VMs, adding a new mapping once a new VM instance starts up. As each host has its own VID space, the VID space limitation (4094) will not affect the number of tenant networks. In the experiment, the isolation method is shown to reduce the CPU usage and increase the throughput for DCN flows, compared with general tunneling protocols such as GRE and VXLAN. Apart from tenant network isolation, the tenant bandwidth isolation can be achieved by maintaining path cost tables on the SDN controller. For example, Shimonishi et al. (2013) proposed an adaptive routing to calculate flow paths with the link cost matrix of the network in real-time. As the tenant networks are identified by a 24-bit VID and constructed over the physical substrate, the adaptive routing fabric only need to focus on meeting the tenants' requirement of bandwidth. Besides, MPLS has been used for tenants isolation as well in large scale DCN. For example, MPLS labels are used to denote the location and routing paths of tenant virtual networks in Zeppelin (Kempf et al., 2013), with the SDN controller identifying isolated links between virtual switches and top of rack switches (TORSs) as well as the isolated paths between TORSs.

Security Middleboxes. Middleboxes with security functions, such as firewall, Intrusion detection system (IDS) and access admission control, have been implemented based on SDN in DCN for lower cost and better performance. Firewall has been implemented with packet filtering rules distributed over SDN substrate in oftables (Koerner and Kao, 2014), an innovative firewall for the security requirement of data center and enterprise networks. With SDN switches tracking flows statistics in DCN, an event-based IDS has been proposed in Hu et al. (2013). When an SDN switch detects anomaly events, it will inform the sub-controller, which will inform the Event Processing Engine to detect network attacks. However, the work is not introduced in details yet. Exploiting the global visibility of DCSDN, an SDN based security system, named Avalanche (Iyer et al., 2014), is proposed to enforce admission control for DCN tenants. With pre-defined security policies on SDN switches, Avalanche can decide whether a new member should be admitted to a multicast group. In StEERING (Zhang et al., 2013), a framework of load balanced middleboxes, security policies can be implemented by dynamically assigning a sequence of middleboxes rather than a single middlebox for flows of each subscriber. The selection of middlebox placements with the global network visibility can be solved by a heuristic algorithm and achieve fine-grained traffic

Table 2
An overview of network security in DCSDN.

Publication	Application	Innovation	Method	Advantage
Nunes et al. (2013)	Traffic isolation	Dummy MAC address	Packet header rewriting	No need hardware switch
Kawashima and Matsuo (2013)	Traffic isolation	Dummy MAC address	Packet header rewriting	No need hardware switch
Kempf et al. (2013)	Traffic isolation	MPLS labeled	Packet header rewriting	High scalability
Shimonishi et al. (2013)	Traffic isolation	24-bit VLAN ID	Packet header rewriting	Open-source prototype
Hu et al. (2013)	IDS	Event-based detection	Controller based detection	Reduce response time
Iyer et al. (2014)	Access admission control	Central control	Controller based detection	Alleviate security concerns
Koerner and Kao (2014)	Firewall	Distributed tables	Distributed filtering	Reduce filter rule chains

steering. Chen et al. (2016) investigates a new type of threat: DDoS attack on the control plane of the SDN architecture. The slow software layer of commodity SDN devices and centralized SDN controller can easily become potential control plane bottlenecks, and in turn be interesting targets of DDoS attacks, which cause massive collateral damage to all network users. Chen et al. (2016) proposes SDNShield, a defense framework towards more comprehensive protection against DDoS attacks on SDN. The framework leverages emerging network function virtualization (NFV) technology to mitigate stress on SDN devices. Meanwhile, it implements a two-stage filtering scheme to ensure that the centralized controller is not overwhelmed. Prototype tests and real data-driven evaluation results show that SDNShield maintains high resilience against intense attacks and keeps performance overhead low.

3.2.3. Discussion

From these SDN based methods of DCN security, SDN can be used to implement a security application, or a pre-filter to the real security servers and middleboxes. Whats more the SDN based security solutions can be implemented on all distributed NEs, which brings balanced load to security services. In Table 2, we compare those network security schemes in DCN in the aspects of application, innovation, method, and advantage.

Besides the benefit brought to DCN, SDN also brings threats. Yan et al. (2016) study the Distributed Denial of Service (DDoS) attacks in SDN based cloud computing environments. Except for the security functions with SDN, there are protections on flow tables, avoiding DDoS from the effect of mice flow flooding. Lee et al. (2013) proposed a differential flow cache layer between the switch and the controller, where the indices and buckets are for elephant and mice flows separately with individual cache replacement strategies. So the flooding of mice flows will not directly evict the elephant flows' flow entries in the flow tables.

3.3. Network resource allocation in DCSDN

For DCN tenant services, the requirements on network resources and computation resources can be fulfilled by SDN based Resources Allocation (RA) methods (Sharkh et al., 2013). We focus on existing methods of RA in DCSDN, which focus on the allocation of network resources rather than computing resources. As the resource allocation methods are implemented with flow rate control that is implemented on hosts or network devices, we classified these methods into host-based control, network-based control, and hybrid control. In host-based control methods, only end hosts control the flow rate using schemes. The DCN application controller allocates server resources to existing DCN services based on network awareness; In network-based control methods, network resources are optimally scheduled and allocated to meet the demand of DCN services; In hybrid control methods, cross-layer resource scheduling with both network and service information is considered. The detailed work on RA will be presented in Section 4, as there are plenty of work on RAs.

Besides the link bandwidth, other resources, such flow table space, also play important roles in performance provisioning in DCN. In OpenFlow switches, flow table stores flow entries that define how to process the received flows. However, current OpenFlow switches are usually equipped with limited flow table space, which may become insufficient when a large number of flow entries are demanded by the network. Yan et al. (2014) proposed an innovative reactive wildcard rule caching scheme that can save the precious flow table space at SDN switches. Guo et al. (2015) proposed a new forwarding scheme named JumpFlow that attains low and balanced flow table usage in DCNs by allowing packets to use VLAN identifier (VID) in their headers to supply routing information.

3.4. Energy saving in DCSDN

3.4.1. Problem

Network devices normally consume 20–30% of energy in a DCN (Heller et al., 2010), Many works have been proposed recently to reduce the power consumption of network devices (Cavdar and Alagoz, 2012). Typically, the DCN architectures are built with rich connections and multipath routings to achieve high network performance. However, such design also causes energy waste due to the following reasons: First network devices are in low utilization as the DCN traffic is typically lower than the peak rate. Second, the power consumed by network devices is usually constant, not scale to traffic load.

3.4.2. SDN based solutions

The main strategies of reducing power consumption of network devices are divided into two categories: traffic aggregation methods and VM placement methods. The traffic aggregation methods are designed to choose a minimum set of active NEs and turn off idle NEs to reduce power consumption.

Traffic aggregation. Traffic aggregation can be implemented on SDN controller, which provides capability of real-time monitoring and managing SDN supported NEs in a centralized way. For example, ElasticTree (Heller et al., 2010) is designed to choose a minimum power subnet of active NEs for any given DCN traffic matrix. In the prototype implementation (shown in Fig. 5), OpenFlow switches provide port statistics (traffic data)to the optimizer, then the optimizer calculates the switch/link subgraph, and sends it to the routing module. Finally the routing module applies it to the network (path control). With SDN monitoring all network resources and abstracting the traffic patterns, the energy minimizing problem can be modeled as a multi-commodity flow (MCF) problem and solved by various optimizers, such as CPLEX, greedy-bin packer, and topology-aware heuristic (TAH). For real world DCN, they showed that the complexity of TAH algorithm is linear to the number of hosts. With switches in discrete power,

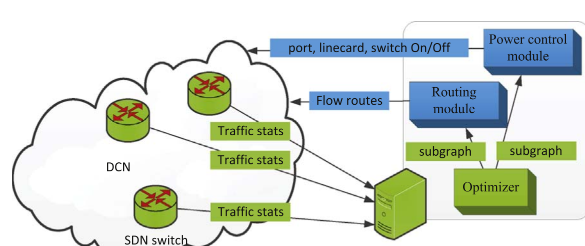


Fig. 5. ElasticTree architecture.

ECODANE (Thanh et al., 2013) is proposed with energy-aware functionalities implemented in SDN controller. ECODANE (Thanh et al., 2013) extended the algorithm of ElasticTree and proposed a new rate-adaptive topology-aware heuristic (RA-TAH) algorithm. The algorithm turns off the right NEs to avoid overload, and improves the power saving level by combining power scaling and smart sleeping.

VM migration for energy efficiency. On top of the traffic aggregation methods, energy savings can also be achieved by VM placements strategies, which consolidate VMs into a small set of pods and servers and potentially reduce the amount of network traffic. For example, VMFlow (Mann et al., 2011) applies an intelligent VM placement heuristic for higher energy efficiencies when they transform the energy efficiency problem into a network-aware VM placement (NAVP) problem subject to the constraints on the topology and VM bandwidth requirements. For the VM placement and VM migration, they proposed a fast heuristic algorithm that can obtain additional energy saving of 15% and support 5 – 6 times more network traffic compared to ElasticTree approach in the simulation with a CLOS topology of 50 ToR switches and 1000 servers. However, the VM placement assumption in VMFlow is not practical, as they assume that a VM or a group of VMs can be only mapped to a server. For practical scenarios with unlimited VM placement, VMPlanner (Fang et al., 2013) is proposed to save energy by solving the NP-complete problems of VM placement as well as the traffic flow routing. The VMPlanner divides the energy saving problem into 3 sub-problems, a Balanced Minimum K-cut problem for the traffic-aware VM grouping, a Quadratic Assignment problem for the Distance-aware VM grouping on server-rack mapping, and a variant of multi-commodity flow problem for the power-aware inter-VM traffic flow routing. Similar to VMPlanner, a unified platform is proposed in Adami et al. (2013) to schedule the network subgraph and VM placement with traffic statistics from SDN switches. There is a Resource Selection Manager that can select computing servers and feasible paths, according to traffic statistics estimated per flow on the connected link. This framework can guarantee the initial resource of VM flows, however, the assignment would become complex in the scenarios of dynamic traffics in DCN.

Exclusive Routing. From perspective of power characters, there are fixed power NEs and speed-scaling power NEs, and discrete power NEs. The NEs with fixed power consumes constant power even when the NEs are idle. So in the ideal case all traffic are transmitted by a minimum set of active NEs with full speed. With a central controller fully controlling DCN traffic, a flow scheduling method (Li et al., 2014) exploits the exclusive routing (EXR), which guarantees the full utilization of active links. In this methods, flows are routed with EXR and scheduled directly by the priority in terms of flow completion time or flow deadline. Each flow will route on a path exclusively and make it possible to fully utilize links of the path. In the implementation, the SDN controller will control the flow status, shifting between send and suspend mode. The preemptive flow scheduling is easy and efficient for online flows. There are two limitations on this work. First they assume all links are in identical capacity and ignore the network links with different capacities. Second the EXR would not guarantee that all active switches are in full utilization anytime or on the whole.

Fine-tune NEs. For speed-scaling NEs, such as the optical switches and OpenFlow switch with NetFPGA, the NE link capacity can be changed with different power and the network energy will be scaling to the size of the traffic. With optical switches selectively multiplexing the streams of differential quadrature PSK (DQPSK) and on-off keying (OOK), optical links with a range of several capacity levels and the proportional power can be exploited to save network power (Malacarne et al., 2014). The network power can be cost-effective when NEs adapt the link capacities to the traffic, which can be pro-actively predicted by the predictable traffic pattern or reactively estimated from the statistic reports of the SDN switches. However, the scheduling schemes turn switches according to the traffic per port without network visibility, which lacks a fine tradeoff between link cost and allowed traffic peak. As the OpenFlow NetFPGA gigabit router can choose discrete port power from 4 alternative power modes, i.e. off, 10Mbps, 100Mbps and 1 Gbps, Thanh et al. (2012) built a platform for in-depth analysis of the performance factors such as energy proportionality, QoS and complexity.

Joint inter-and intra-datacenter work load management. Guo et al. (2014), Guo et al. (2013) presented a joint inter- and intra-datacenter workload management scheme on the SDN controller called JET to determine the best workload distribution among data centers in a short processing time. JET jointly considers an array of pertinent factors, such as the variation of local electricity prices, the number and location of active servers in a data center, and the status of a cooling system, that influence the electricity cost of data centers. JET provides the benefit of trading off the electricity cost of active servers and cooling systems by selectively selecting the electricity prices or the efficiency of a cooling system as the main factor for determining the electricity cost of geographically distributed data centers. Comprehensive simulation demonstrates that JET is considerably more efficient than state-of-the-art systems and can save up to 50% of the electricity cost of distributed data centers.

3.4.3. Discussion

The SDN techniques provide a centralized controller that can estimate the energy consumption of each NE, and then optimize the overall energy consumption by traffic engineering or inform energy information to the DCN applications for cross-layer optimization. In Table 3 we list the state of the art of Green DCSDN, with different concentrates on power elements, solution, and constraints. The power model of switches can be divided into fixed and fine-granular. The power is the same in both idle and working status for general physical switches, while the power can be adaptive to the number of active ports for the switches with fine-granular power. The objective of minimizing the network energy cost can be transformed to the problem of maximizing the network energy efficiency that can be solved by TAH algorithms based on SDN. Considering the practical delay in the switch mode shifting, parallel traffic flows in the Time Division Multiplex

Table 3
An overview of energy saving in DCSDN.

Publication	Power element	Approach	Constraint	Algorithm
ElasticTree Heller et al. (2010)	Switch	MPNS[a]	Unsplittable	TAH[b], GBP[c]
ECODANE Thanh et al. (2013)	Switch, port	MPNS	Discrete-powered NE	Rate adaptive TAH
Malacarne et al. (2014)	Optical port	MPNS	Tunable port	–
VMFlow Mann et al. (2011)	Switch	VM placement	Unsplittable	Greedy heuristic
VMPlanner Fang et al. (2013)	Switch, Port	MUNE[d]	Elastic, fixed-sized and unsplittable	GBP
EXR Li et al. (2014)	Switch, port	MUNE	Fixed-sized and unsplittable	Greedy

[a] MPNS: minimal power network subset.
[b] TAH: topology-aware heuristic.
[c] GBP: Greedy Bin-Packing.
[d] MUNE: maximal utilization of active network elements.

Access (TDMA) links should be scheduled to make full use of the time slices. This work proposes the path-level collision avoidance as an extreme case. However, the DCN application controller would have to schedule existing traffic dynamically and comprehensively, which may be a high cost of control.

There are other methods for DCN energy efficiency. For example, packet length can be reduced by introducing the Flow-ID for route and trimming header (Kannan and Banerjee, 2012), which can decrease the switch fabric power consumption by a factor of 2.5 compared to existing L2 switch. The power controller with SNMP (Mahadevan et al., 2011) can control switches to adapt the current flow rate in DCN. Recently, traffic grooming in underlying network is regarded as an effective method for energy saving by using fewer transponders and, by avoiding the need of additional guard bands (Sankaran and Sivalingam, 2016; Zhang et al., 2014, 2015). Widjaja et al. (2013), Widjaja et al. (2014b) propose a new approach by exploring the design stage of a datacenter network, and focus on how to choose the right switch size that can potentially save the most power during the expected operation of the network. Widjaja et al. (2013), Widjaja et al. (2014a) also consider speed scaling where the power of a switch can be varied by adjusting its processing rate according to its traffic demand. They reveal that deploying a large number of small switches is more power-efficient than a small number of large switches when the traffic demand is relatively moderate or when servers exchanging traffic are in close proximity. With speed scaling, the reverse is generally true.

4. Network resource allocation in DCSDN

In this section, we show detailed survey of RA in DCSDN, which is divided into three aspects by the controllable elements for RA. In host-based control, end hosts will proactive or reactively adapt the flow rates. In network based control, the flow rate is allocated on switches and software switches only, and in hybrid control, both end hosts and network are controlled for flow allocation.

4.1. Host-based Control

4.1.1. Problem

In end hosts, the rate of a flow is constrained by the traffic generating rate of applications and the available bandwidth of allocated network resource slice. The former depends on the application's implementation while the latter is determined by the characteristics of networks, such as network capacity and congestion situation. As the underlying network resources are transparent in traditional networks, end hosts can only reactively measure the network status and make heuristic adaption of flow rate. In traditional DCN, end hosts can adapt their transfer rate to saturate bottleneck links using by existing protocols such as TCP without knowing details of network status (e.g., topology and link utilization). However, the unawareness of the network links and traffic flows would cause lower utilization of link bandwidth, leading to resource waste, especially for the typical DCNs built with rich connections.

4.1.2. SDN based solutions

For DCN applications, end hosts would be a set of VMs assigned to each job. The VMs are allocated to servers with available VM slots and network resources. Unawareness of structural resources in servers would cause poor network resource utilization during VM-to-server assignment. When SDN is applied to DCN, novel methods of host-based control have been proposed in optimizing end rate adaption by considering network and traffic character. SDN provides network visibility such as network topology and link utilization to facilitate the flow rate control, for which we list relevant schemes proposed in literature in Table 4. End hosts can be scheduled to change their rates or migrate to alternative DCN servers. The network visibility can be exploited in designing the scheduling strategies of DCN application, such as VM placement in Hadoop job scheduling. SDN can also be used to reproduce existing protocols with simplified implementation.

TCP-like. To maximize the network resource utilization, end hosts can adapt their sending rates based on feedback of network congestion using TCP-like schemes. Network congestion can be monitored lively with SDN switches programmed to count link utilization and traffic matrix. OpenTCP is proposed to use a controller application to maintain the underlying network topology and traffic, detect the congestions status, and change the TCP behaviors of end-hosts (Ghobadi et al., 2012). In implementation, OpenTCP requires installation of an agent on each end-host for TCP sessions, which is feasible in data center environment. However, the study in Ghobadi et al. (2012) did not consider the detailed relation between the TCP parameter and network measurement result. The detailed relation has been studied in Jouet and Pezaros (2013), where an SDN based TCP control loop is designed to solve the slow-start problem of DCN mice flows. In DCN, the bandwidth delay product (BDP) is significantly low, so the TCP parameters such as the initial window and the retransmission timer can be tuned based on the temporal measurement at the SDN controller. In this design, a central controller calculates the minimal bandwidth delay product in real-time and monitors network-wide view of the topology, bandwidth, latency and switch buffer. Based on these dynamic congestion parameters, the end hosts' networking stack can turn TCP's initial windows. Simulation results show the benefit of network aware per-flow tuning with latency decreasing of factor 8, flow start and completion times improved by a factor of 2 and 5. However, the benchmark on the number of packets is not presented, as not only the elephant flows but also the mice flows are adjusted in real-time.

Google argued that the reliable transport protocol, TCP, requires more than one round trip times (RTTs) in connection establishment. And they proposed Quick UDP Internet Connection (QUIC) (Hamilton et al., 2016) for their DCN in June 2015 atop UDP to functionally replace the suit of TCP+TLS+HTTP/2. The QUIC multiplexing and flow control replaces HTTP/2, the QUIC security replaces TLS, and the connection semantics, reliability, and the congestion control of QUIC replace TCP. For a connection, the QUIC client maintains multiple streams and each stream will have its own flow control. A slower stream won't completely starve the connection as it can shrink to a smaller window or be blocked from receiving more data. However, the QUIC protocol only works in the end host, and is unaware of the global network topology. An SDN controller can be used to optimize multi-path routings for QUIC connections to avoid bandwidth competition between streams of the same QUIC connection.

Max-Min fairness. For Cloud network resource sharing, the fairness weight can be defined in different levels, ranging from VM, flow to the tenant. In the VM level, VMs on each server fair share the server's bandwidth; in the flow level, all flow has the same weight; in the tenant level, each tenant's weight is defined by the number of VM-pair flows. Faircloud (Popa et al., 2011) is proposed to apply max-min fairness in bandwidth allocation between tenants with different tenant weights. As each VM's weight is determined by the tenants' payment, the tenant weight can be determined by three different allocation policies at different levels: Network-level(PS-N), Link-level(PS-L), and Proximate Link-level (PS-P). The PS-L weight of a tenant is determined by the sum weight of VMs communicating through the given link. The PS-N weight of a tenant is determined by the sum weight of VMs in the network. The PS-P splits the PS-L weight for the weight for each link of VM pairs according to the distance between the link and the two VMs, which is suitable for the tree-based topologies. These weight models can be used to calculate the resource reservation per link for existing flows in flow routing, where PS-L achieves link proportionality and provides utilization incentives, PS-N achieves better proportionality at the network level, and PS-P achieves guarantees and incentives for high utilization. In ElasticSwitch (Popa et al., 2013), the SDN supported virtual switch on each server can turn the flow rate of each VM in real time according to the utility of virtual links. The DCN is modeled by the

Table 4
An overview of Host-based control of RA in DCSDN.

Publication	Network Aware model	Rate Adaption model	Centralized	Network Control method	Implementation
Ghobadi et al. (2012)	Proactive	Controller	Yes	TCP-like	Congestion detection
Jouet and Pezaros (2013)	Reactive	Controller	Yes	TCP-like	TCP initial window
Faricloud Popa et al. (2011)	Reactive	Controller	Yes	Weight proportional	Setting switch queue
Popa et al. (2013)	Reactive	Virtual switch	No	TCP-like	Guarantee bandwidth per flow
Guo et al. (2013)	Reactive	VM-pair	No	Game-theoretical	Heuristic without dual variables delay
Guo and Liu (2014)	Reactive	VM-pair	No	Logistical model	Heuristic

Table 5
An overview of network control of RA in DCSDN.

Publication	Application aware	Method	RA objective	DCN specific	Algorithm
LiveCloud Wang et al. (2012)	Tenants specify	Orchestration	Multi-objective	Tenant QoS, path diversity	–
FPTAS Agarwal et al. (2013)	NE measure	Dynamic Routing	Maximum link utilization	Loop free, minimal congestion	Primal dual algorithm
Baatdaat Tso and Pezaros (2013)	NE measure	Greedy routing	Maximum Link utilization	LB for inter- and intra-Rack	Depth-first search
TASDN Zhao et al. (2013)	AC notification	Time-aware SDN	End-to-end QoS guarantee	Delay demands	–
PANE Ferguson et al. (2013)	User notification	Hierarchical Flow Tables	Minimal conflict demands	User requires may conflict	–
Ji et al. (2013)	Racks TOR	Subcarrier assignment	Fine granularity	OFDM based optical DC	–
LABERIO Long et al. (2013)	NE measure	Rerouting biggest flow	Traffic LB	Large scale, dynamic flows	Path switching
Kanagavelu et al. (2013)	NE measure	Least loaded routing	LB, resident capacity balanced	Mix of large and small flows	Greedy
SiBF Macapuna et al. (2010)	–	Random iBF switching	LB	L2, failure prone, multipath.	–
MSRI Yang et al. (2013)	Service-aware flow estimation	Optical path	QoS (bandwidth, latency)	–	–
Rosa et al. (2014)	Tenant specify	Mapping & Bookkeeping	Resilience, bandwidth guarantee	Load balanced virtual network	BFS
Nejabati et al. (2013)	User specify	Mapping	Maximum mapping ratio	Virtual DC	–
Channegowda et al. (2013)	User specify	Mapping	Service aware LB	Various applications pattern	–

hose-model (Popa et al., 2011), where servers connect to a switch without bottleneck. Since flows can fairly use the server bandwidth, the flow rate can be upper bounded by the minimal average bandwidth of its VMs. According to the TCP feedback, the flow rate can be calculated by the rate function of existing weighted TCP-like algorithms, such as Seawall (Shieh et al., 2011). Alternatively, the rate function can also exploit TCP variant latency rather than bandwidth for congestion, as the compatibility of latency and bandwidth variants for network congestion is unknown yet.

Bargaining between VMs in Hose model DCN. With Hose model, Falloc (Guo et al., 2013) formulates the RA problem of VMs into a bargaining problem. In Hose model, servers are connecting to a non-block virtual switch, and their VMs will share the server I/O bandwidth competitively. So Falloc formulates the fair bandwidth allocation as a bargaining game problem, and solves it by dual-based decomposition, which consists of 2 steps. Firstly the primary problem can be converted into a dual problem, and then iteration can be designed to optimize the step-size by the sub-gradient method. Inspired by the smooth and fast-growing function of Logistic model, Jian et al. proposed a distributed rate control algorithm with the VM-flow rate changing according to the rate control mode (Guo and Liu, 2014) for traffic requirement between VMs as a bargaining game problem. With the characters of slow start, exponential convergence, and stable equilibrium, the rate control mode of VM-flows is formulated by the rate of flow, the factor for the bottleneck link, the VM weight, the link utilization, and the number of flows on each link. Then the rate factor from a flow's bottleneck link controls the rate at the source. They verified the algorithm with prototype for static requirements and evaluated the performance by trace-driven simulations with dynamic requirement.

Network aware VM placement and migration. The VM placement is to select available physical host for each VM, which can be optimized to minimize the number of network devices between communication VMs. The VM placement problem has been solved by a bin-packing heuristic algorithm in the CloudNaas framework (Benson et al., 2011), which arranges communications VMs into virtual network segments and then sorts them by number of VMs, and finally allocates them into the nearest hosts. The location of VMs can be the same host, the same rack, the same aggregation device, any available physical host. Based on the VM information maintained by SDN technology, the VM migration procedure (Liu and Jin, 2013) is designed to expand sub-net seamlessly with minimized migration time and service downtime. The migration procedure specifies the detailed actions of DC controller, network controller, and the global orchestrator. The SDN controller provides predictable performance with the network visibility and the VMs' migration bandwidth request. However, the VM migration procedure lacks of bandwidth reservation strategies in this work.

4.2. Network based Control

4.2.1. Problems and SDN based solutions

From the aspect of network based control, flow demands are one of the input to the network resource allocation, which can be directly informed from the DC application controller or indirectly from estimation of SDN NEs. With SDN architecture, Network RA can be designed with network visibility to allocate network resources in high efficiency, especially for multicast routing trees. What's more, network RA can schedule network resources with strategies of reservation, dynamic allocation, and load balance, which guarantees the performance of network resource allocation. The network control methods has been designed with optimization in Load Balance (LB), Link Utilization, QoS of resilience and bandwidth guarantee, summarized in Table 5. there are SDN based DCN architectures that can allocate network resources with programmed NEs. And there are network based resource allocations with network services to provide tree routing, bandwidth resource reservation, and load balance.

SDN based DCN architecture. There are various SDN based DCN architectures that support resource allocation. The LiveCloud (Wang et al., 2012) is a pure network control RA architecture that can meet the tenants' demands of connectivity, bandwidth and latency. With SDN NEs tracking the network topology, LiveCloud maintains a consistent view of the global network, and provides a uniform resource description to abstract and quantify network resources, including the physical NEs performance, capacity of memory and storage, and the supporting functions. Calculating the placement of tenants' workload, resource allocation algorithms can provide service-level agreements and speed up service delivery in heterogeneous physical networks. With service controller assigning the delay requirements, the end-to-end light-path can be calculated in a time-aware SDN architecture, called TASDN (Zhao et al., 2013). The time-aware service scheduling (TASS) strategy can be designed to determine the path and schedule for service migration, which is a cross stratum optimization of application and network resource. As a standard framework for DCN network virtualization, Mirage (Rotsos et al., 2012) provides backward compatibility for Mirage servers with different stack layers. With a high level source code compiled for various servers, Mirage can provide network aware extension for network resource virtualization and inform running applications the global view of traffic matrices effectively and accurately. However, the integrated controller is too simple to provide suitable higher level event representing, such as link discovery or routing protocol specific events.

Shortest Routing Tree. The multicast in DCN has been a prominent topic with a particular attention to the resource allocation problem. Fully utilizing the rich path diversity of DCN, the network multicast traffic can be optimized with SDN. Iyer proposed AvRA (Iyer et al., 2014), a bandwidth efficient multicast routing algorithm that leverages SDN's global network visibility to minimize the size of the routing tree for each multicast group. In traditional multicast routing protocols, Routing Tree Formation finds the shortest path for each member individually. Based on the global visibility, AvRA designs the multicast tree by adding members one by one with the shortest path to the existing tree. Designing minimum routing tree for each multicast group, AvRA can reduce the solution to a Steiner Tree problem for typical topologies such as FatTree and Tree. The emulation on Mininet showed that AvRA achieves 12% increment of application data rate, and 51% decrement of packet loss on average compared to IP multicast.

Resource Reservation. The flowVisor (Sherwood et al.,) works in SDN networks, splitting network resources into slices. M2cloud (Liu et al., 2013a, 2013b) extends flowVisor to enforce cross-site performance isolation for tenants. The global controller in M2cloud architecture can build tunnels to local controllers of each DC to exchange local information. In OpenNaaS, an open-source framework of network resources management for cloud DCs, a network connectivity service (Escalona et al., 2013) is designed to reserve VM-to-VM connectivity, and resources across DC sites. For hybrid NE networks of SDN and non-SDN NEs, SDN-TE services (Agarwal et al., 2013) on SDN controller has been designed by considering the fixed shortest paths of non-SDN NEs. However, this SDN-TE only handles traffic on packet granularity, and there is no QoS consideration. For DCN across different domains, such as optical, MPLS or IP networks, a uniform abstraction named Open Transport Switch (OTS) (Sadasivarao et al., 2013) is built on SDN to deliver optimal multi-layer network resource allocation. Deploying SDN controllers in both IP and optical networks, Yang et al. (2013) can monitor the network status and the flow size, shifting large flows from IP network to alternative optical networks. PANE (Ferguson et al., 2013) can reserve link resources by reserving queue size of switch ports. To avoid allocation conflicts, PANE is designed with guarantee of the minimum requirement for each flow. When PANE cannot allocate a new flow, it will schedule the flow starting time by delaying it. However, PANE only provides a fixed bandwidth reservation, which would not scale with applications' traffic variation. With SDN abstracting cloud tenants, the Cloud Atlas (Baucke et al., 2013) is proposed to build virtual link connections between sites in multi-site cloud networks by maintaining multiple point-to-point paths, where bandwidth allocation is implemented via WAN virtualization technology such as IPSec (Kent and Seo) VPN, VPLS (Lasserre and Kompella), etc. Also for virtual interconnections between virtual DCs, a Breadth First Search (BFS) resource bookkeeping algorithm (Rosa et al., 2014) is proposed to map available network resources to virtual paths, with flow tables of ToRs programmed to count the available bandwidth of their interconnected servers and racks. The efficiency and load balancing benefit under different tenant traffic patterns is evaluated through emulations. For optical networks, the optical add-drop multiplexer can be scheduled with flexible grid technology for resource allocation. For instance, virtual links can be mapped to physical optical fibers in Nejabati et al. (2013), even for hybrid optical and ethernet DC networks, the network abstraction (Channegowda et al., 2013) can allocate connection and bandwidth according to the service requirements. The SDN controller can dynamically assign OFDM subcarrier to flows between ToRs in optical networks, which is first demonstrated with real-time advanced solutions in Ji et al. (2013). However, the OFDM with burst switching is complicated and expensive. Peng et al. (2015) extends OpenFlow protocol to support communication between the extended OpenDaylight SDN controller and the optical DCN devices for flexible resource deployment. Yin et al. (2017) exploit OpenFlow-based unified network control and management for both intra- and inter-DCNs.

Load Balance. For DCN replication operations, the SDN controller has been designed for load balance and congestion avoidance with the load measurements on links and paths (Kanagavelu et al., 2013). For any newly generated flow, the route is optimally selected with load balance and congestion avoidance; however the flow routing would not guarantee the optimal routing set for existing flows. For DCN networks with dynamic and short duration flows, it's impractical to schedule all flows on an SDN controller in real-time. Instead, flow forwarding can be scheduled reactively on SDN switches with real-time link utilization statistics. For example, Baatdaat (Tso and Pezaros, 2013) balances intra-DC traffic by updating available paths of each flow according to real-time network statistics, so that the SDN switches can automatically shift flows to lowest utilized links for balanced link utilizations. For practical cloud DCs, the SDN switches measure the link utilization of down links per $1ms$, as the average inter-arrivals time is reported to be $10 - 15ms$. By rerouting schemes, LABERIO (Long et al., 2013) is proposed to mitigate network congestions for ToR-to-ToR flows reactively, which is the first work of dynamic and global path-switching load balance on middlebox in DCSDN. The load-balance scheduling algorithm can shift the largest flow out of the busiest link reactively in a greedy way, however, the complexity increasing significantly with the traffic volume. The forwarding plane distributed in multiple ToRs can maintain load balance by calculating paths in each ToR Switching with the in-packet Bloom Filter (SiBF) data center architecture (Macapuna et al., 2010). The load balance schemes randomly pick the iBF for each request of new flows, which would use more waypoints compared to the traditional protocol of ECMP.

4.2.2. Discussion

As a supplementary aspect of network based resource allocation, the simplification and hardware acceleration can be exploited to optimize the efficiency of SDN control plane. Hedera (Al-Fares et al., 2010) is proposed to dynamically schedule the reservation of elephant flows by a Simulated Annealing searching method for available paths. Although Hedera reduced the controller load by filtering the request of small flows, adaptive load-aware flow scheduling is still a challenge for data center networking. To reduce the workload of SDN controller, Bi et al. (2013) proposed a practical elephant flow detector on switches, where network flows are sampled and only suspected elephant flows are uploaded to SDN controller. The optimal and adaptive elephant flow detection threshold is formulated and computed with the concept of the central limit theorem. The simulation shows a decrement of detection error rate, compared with the fixed threshold detections in Mahout and DevoFlow (Curtis et al., 2011). To promote the centralized computation of SDN controller, hardware such as GPU with CUDA system has been adopted for the powerful computation, where iterative Parallel Grouping algorithm (IPGA) (Ke et al., 2013) can be designed for TE problems, even for the NP-hard problem of prioritized flow scheduling with QoS guarantee and network energy saving in DCN.

4.3. Hybrid control

For the hybrid control, the SDN controller will control the end hosts and NEs comprehensively. The end hosts will be admitted with allowed flow rate, and the NEs will be programmed to optimize the utilization of available link resources from network visibility. For example, in hadoop applications, the job scheduling strategies can exploit the network visibility, while the routing can be scheduled with awareness of the communication patterns of aggregation, data shuffling and partially overlapping aggregation traffic pattern (Wang et al., 2012) to improve the performance of Hadoop application. The SDN based applications can cooperate with existing network appliances or network protocols, and make full use of hybrid NEs and network resources.

In production DCs distributed in different geographic locations, the inter-DC connections are wide area networks (WAN) with high bandwidth and reliability. However, the WAN links are usually provisioned to 30–40% average utilization. This issue has been studied recently by inter-DC TE with SDN. One of the first and largest real-world deployment of DCSDN reported is Google's B4 data center (Jain et al., 2013), with a private SDN based WAN connecting DCs across the planet (shown in the Fig. 6). DC administrator assigns each flow a bandwidth function corresponding to the flow priority. Then the flows of DC-to-DC can be aggregated into flow groups on the edge switches of each DC and be solved by centralized TE optimization algorithm in B4, which has been implemented at low cost with hardware limited switches in two steps. Firstly, the B4 SDN controller assigns flow groups tunnel bandwidth with fair share; secondly the assigned bandwidth will be quantized by regularizing the flow splitting granularity to reduce the operational complexity of switches. The quantization can be implemented by a greedy approach and heuristic algorithm under the constraints of both fairness and throughput efficiency. Although the inter-DC TE can be solved by LP algorithm (Danna et al., 2012) generally, the centralized TE optimization Algorithm in B4 optimizes the utilization of WAN links with 25x faster performance. The centralized TE services can significantly increase the inter-DC link utilizations, with 2-3x efficiency improvements compared with standard practice. However the adjustment algorithm would not scale with the number of the flow groups. Similar contributions can be found in SWAN data center (Hong et al., 2013), where a centralized SDN controller is exploited to fully utilize link and flexibly allocate resource in inter-DC WAN. The SWAN architecture (in Fig. 7) consists of two middlewares: network agents and the service brokers. The network agents are deployed between the SDN controller and SDN switch, tracking topology and traffic flows, and installing flows. The service brokers are deployed between the SDN controller and service hosts, estimating service's current demand and negotiating allowed rate to each other DC with SDN controller. The SWAN core divides traffic flows into 3 priorities with fair sharing bandwidth allocation in each priority. A general algorithm of flow allocation with max-min fairness (MMF) will consist of iterations of LPs to find the most saturated links. When the network topology and flow demands change, the SWAN controller will have to update network allocation for all flows in a congestion-free way, which has been implemented by a set of intermediate updates step by step. To calculate a set of non-congested intermediate updates, the SWAN controller uses a LP-based algorithm to find the minimal steps of efficient and congestion-free updates.

The DCN traffic can be divided into deadline guaranteed and bandwidth guaranteed service. The former requires a fixed deadline, while the latter requires an immediately guaranteed bandwidth. With network awareness in DCSDN, Time-aware SDN (Ta-SDN) (Zhao et al., 2013) is proposed to determine the rate and sending interval of each flow with programmable network and applications, which is the first work on inter-DC traffic scheduling in flexi-grid optical transport networks. Ta-SDN models the DCN flow allocation by bandwidth-delay product, which can be spectrum rectangle fragments in the resource distribution figure of each link. To maximize the bandwidth utilization and minimize the network blocking probability, Ta-SDN has proposed a time-correlated Path Computation Element (PCE) algorithm, assigning each flow the path with both feasible bandwidth and minimal utilization. However, the SDN controller will not scale well with large scale DCN, as Ta-SDN has to schedule transmission time and network bandwidth of each link at the same time. As shown in Table 6, the SDN controller can handle inter-DC TE through different methods.

For application reliability, DCN would possess replicas of files in multiple servers, which would case duplicate content of DCN traffic, for example, up to 50% for http and over 30% Mapreduce. With SDN monitoring redundant traffic, the Traffic Redundancy Elimination (TRE) in Data Center (TREDaCeN) problem (Cui et al., 2013) was first formulated and solved by caching task assignment. TREDaCeN can detect the redundancy of task traffic and reduce the redundant transmissions from sources to cache routers. The main functions of TREDaCeN are cache task assignment, and cache replacement. The cache task assignment should decide the cache routers as well as the cached data for a long term benefit, which is NP-hard as it can be transformed from a cycle cover problem. In TREDaCeN, the NP-hard problem is solved by greedy offline algorithms or online algorithms with cache assignment and cache replacement. The TRE problem is simplified with all router cache capacity the same, which may vary in heterogeneous networks.

To optimize the link utilization and energy saving, flow migration and VM migration can be exploited simultaneously with the traffic awareness. The first combination of flow and VM migrations (Lin et al.,

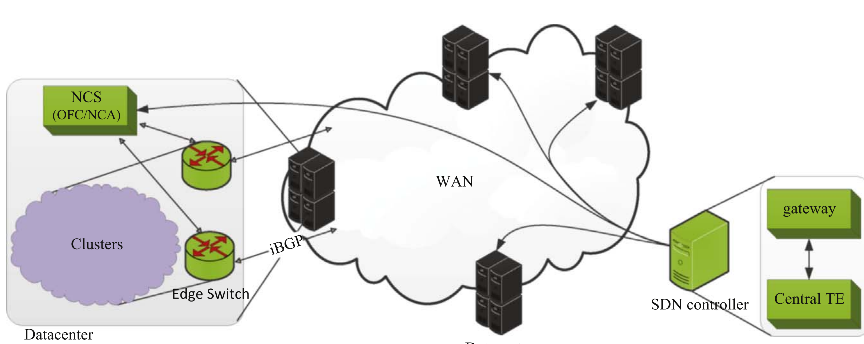


Fig. 6. B4 architecture. The SDN controller deployed on a central server carries out Inter-DC TE, while the network control services (NCS) such as OpenFlow Controller (OFC) and Network Control Applications (NCA) monitor the traffic between hosts in DC clusters.

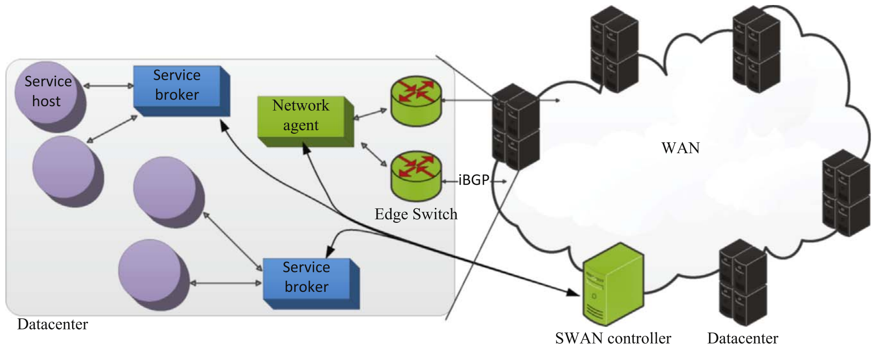


Fig. 7. SWAN architecture. The SWAN controller calculates the inter-DC Resource Allocation by fully exploiting the global visibility of the inter-DC WAN, while the service brokers inside each DC are collecting traffic requirements from service hosts and the network agents are monitoring the WAN visibility from switches.

Table 6
Inter-DC RA in DCSDN.

Publication	Problem model	Solution
B4 Jain et al. (2013)	TE on Flow groups	MMF, quantization
SWAN Hong et al. (2013)	Network fabric	α-approximated algorithm
TC-PCE Zhao et al. (2013)	TE on Flow groups	Time-correlated PCE

2013) is proposed with awareness of both energy and topology. The combination scheme first chooses the over- and under-utilized server, and then migrates relevant VMs to a nearby server with awareness of network distance. The experiment result shows a 42.5% increment in throughput and 2.2% decrement in energy overhead. However, the link threshold for over- and under-utilization are set manually, which lacks fine-tunable and reasonable setting schemes.

Chen et al. (2013) investigates the issue of virtual machine placement in a geographically distributed cloud system. It highlights the cloud system's intrinsic trade-off between local Power Usage Effectiveness (PUE) and wide-area network communication costs, and formulates the Cost-Aware Virtual Machine Placement (CAVP) optimization problem to characterize this tradeoff. Since the CAVP problem is NP-hard, Chen et al. (2013) develops a metaheuristic solution to optimize a large cloud system's operating costs within reasonable computation time.

5. Lessons learned and future work

As the DCSDN architecture provides programmable interfaces of network management and application scheduling, there are some lessons we have learned. Firstly, the practical DCSDN scenario in real-world DCN experiments or data-driven simulations is always built with variation of traffic workload and packet size. The real-world platform always built by hardware and software switches with SDN. Secondly, the SDN based traffic optimization solutions over multiple DCNs can deploy synchronized SDN controllers cross multiple DCN, however, the scalability would be limited by the communication capacity between SDN controllers, which may be a bottleneck. Finally, the SDN control plane can be abstracted or integrated for further applications to manage various DCN architecture simultaneously. In the following section, we consider the tendencies of future DCSDN.

5.1. Configuration for programmed network function

The DCSDN automatic configuration depends on the DCN traffic pattern, so the network configurations can be abstracted and packaged for typical applications, such as virtual topology of tenant networks, and inter-DC VM migration. For DCN automatic configuration, existing researches focus on configuring strategies with configuration conflicts detection. For inner-DC automatic configuration, inter-DC transport network always contains NE without SDN support, which should be considered in future innovations of automatic configuration in DCSDN.

5.2. Efficient network security service

In SDN networks, any SDN switch can be programmed with network security functions to handle traffic in different granaries of packet, flow, and aggregated flow. Most work are focusing on implementation and placement of intelligent middleboxes, which provides virtual network security for traffic over the same physical networks. In SDN architecture, network security functions can be implemented on a sequence of network appliances for higher security and stability. For example, the policy programming in StEERING (Zhang et al., 2013) can dynamically determine sequence of security policies for flows of each subscribe, in which the middlebox placement problem can be solved by a heuristic algorithm with the global network visibility. As the SDN bring benefit to DCN, it also bring the threats of SDN to DCN, such as the Distributed Denial of Service (DDoS) Attacks (Yan et al., 2016) to both DCN and SDN in SDN based cloud computing environments. Besides, the SDN technique decouples the control and data plane, and exposes REST API to the third-party control applications, where attacks may happen to SDN NEs, controller, the third-party control applications and administrators.

5.3. Comprehensive resource allocation in DCSDN

Most of resource allocation methods in DCSDN focus on revisiting traditional networking methods in SDN architecture and designing controller functions with associated functions programmed on NEs. The integration of DC controller and SDN controller can bring innovations on network aware applications and application aware networking by assigning DC server resources and network resources simultaneously. Besides, the delay of control plane should be considered in large scale DCN for large amount and variation of DC traffic flows with different time sensitivities.

5.4. Network awareness DCN application

The main methods of DCN power saving are increasing the energy efficiency of power devices, as DC servers and switches are assumed to support sleep status with lower power cost. The characters of DC power devices can be considered in the future work. For example, the energy cost of switching time between sleep and active status would not be non-trivial when the DC traffic are short and burst.

The most general methods of energy efficient are consolidating the traffic into a minimum subset of NEs, leaving other idle NEs in lower power mode. The scalability is challenged in the following cases. First, when the huge amount of DCN traffic are randomly and dynamically generated, the consolidation subgraph would have to be updated for the overall energy optimization. The energy efficient algorithm of SDN controller will have to update the energy efficient consolidation once a flow or connection changes in the network, and the control plane will have to synchronize flow routings and re-routings. Second, if the traffic is generated uniformly, the consolidation subgraph would expand to the full graph, which will bring less energy savings.

5.5. Lower cost architecture for DCSDN

There are some practical challenges for DCSDN architecture in optical networks. For optical networks, the SDN has to synergy with optical networks in programmability, visualization and end-to-end optimization (Pandey, 2013), as the next generation DCN will have to be optimized, elastic and agile with full use of optical links by SDN techniques (DeCusatis, 2014). As the future DCN and HPC platforms will require enhanced scalability, latency and throughput (Perelló et al., 2013). So the DCSDN solutions should empowers transmission technologies of optical circuit switching and optical packet switching with superior flexibility, manageability and customizability by elaborating on the envisioned DCN data plane and the unified SDN-enabled control plane.

The DCSDN has been a new trend for DCN, as the DCSDN controller can integrate SDN controller and DCN controller, monitoring and managing the network with awareness of DCN applications as well as DCN traffic in a central controller. On the one hand, DCSDN controller can improve the network manageability, scalability and dynamism in DCN, with the network infrastructure programmable and manageable at scale. On the other hand, DCSDN controller can predict the traffic pattern and demands from the service pattern in DCN.

6. Conclusions

The DCN with SDN has attracted more and more attentions in recent years. In this paper, we survey the main issues in DCSDN, including the automatic configuration and security, the resource allocation and energy saving. The direct motivation of DCSDN comes from the SDN benefit on automatic configuration for inter-and intra-VM migration, rerouting, and NE replacement for DCN. The network security methods in DCSDN have been surveyed on the network or traffic isolation, traffic filtering, IDS and tenants access control for multi-tenant virtual DCN networks. The resource allocation issues have been solved in various ways, which have been presented from aspects of host-based control, network control and hybrid control. For energy saving, the aggregations can be formulated based on different models with different assumption of power elements. In addition we present the lessons and trend for the promising DCSDN. We expect that this comprehensive survey on data center with SDN will attract more attention to this area.

Acknowledgements

This work was supported by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant No. 2012BAH93F01, the Innovation Research Fund of Huazhong University of Science and Technology, No. 2015MS037 and the National Natural Science Foundation of China under Grant No. 60803005.

References

A.G., 2015. 30% of servers are sitting comatose. ⟨http://anthesisgroup.com/30-of-servers-are-sitting-comatose⟩ (Accessed 23 January 2016.

Adami, D., Martini, B., Gharbaoui, M., Castoldi, P., Antichi, G., Giordano, S., 2013. Effective resource control strategies using openflow in cloud data center. In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 568–574.

Agarwal, S., Kodialam, M., Lakshman, T., 2013. Traffic engineering in software defined networks. In: INFOCOM, 2013 Proceedings IEEE, pp. 2211–2219. ⟨http://dx.doi.org/10.1109/INFCOM.2013.6567024⟩.

Albert, G., Hamilton, J.R., Navendu, J., Srikanth, K., Changhoon, K., Parantap, L., Maltz, D.A., Parveen, P., Sudipta, S., 2009. Vl2: a scalable and flexible data center network. Acm Sigcomm Comput. Commun. Rev. 39 (4), 95–104.

Al-Fares, M., Loukissas, A., Vahdat, A., 2008. A scalable, commodity data center network architecture. Acm Sigcomm Comput. Commun. Rev. 38 (4), 63–74.

Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., Vahdat, A., 2010. Hedera: dynamic flow scheduling for data center networks. Netw. Syst. Des. Implement., 281–296.

Azodolmolky, S., Wieder, P., Yahyapour, R., Sdn-based cloud computing networking, In: 2013 Proceedings of the 15th International Conference on Transparent Optical Networks (ICTON), pp. 1–4. ⟨http://dx.doi.org/10.1109/ICTON.2013.6602678⟩.

B.-S. Lee, R. Kanagavelu, K. Aung, An efficient flow cache algorithm with improved fairness in software-defined data center networks. In: Proceedings of the 2nd International Conference on Cloud Networking (CloudNet), 2013 IEEE , pp. 18–24. ⟨http://dx.doi.org/10.1109/CloudNet.2013.6710553⟩.

Bari, M.F., Boutaba, R., Esteves, R., Granville, L.Z., Podlesny, M., Rabbani, M.G., Zhang, Q., Zhani, M.F., 2013. Data center network virtualization: a survey. IEEE Commun. Surv. Tutor. 15 (2), 909–928. http://dx.doi.org/10.1109/SURV.2012.090512.00043.

Bates, A., Butler, K., Haeberlen, A., Sherr, M., Zhou, W., 2014. Let sdn be your eyes: Secure forensics in data center networks. In: Proceedings of the NDSS workshop on security of emerging network technologies (SENT'14).

Baucke, S., Ben Ali, R., Kempf, J., Mishra, R., Ferioli, F., Carossino, A., 2013. Cloud atlas: A software defined networking abstraction for cloud to wan virtual networking. In: 2013 IEEE Proceedings of the Sixth International Conference on Cloud Computing (CLOUD), pp. 895–902. ⟨http://dx.doi.org/10.1109/CLOUD.2013.44⟩.

Belbekkouche, A., Hasan, M.M., Karmouch, A., 2012. Resource discovery and allocation in network virtualization. IEEE Commun. Surv. Tutor. 14 (4), 1114–1128. http://dx.doi.org/10.1109/SURV.2011.122811.00060.

Beloglazov, A., Abawajy, J., Buyya, R., 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Generation Computer Systems 28 (5) 755–768, special Section: Energy efficiency in large-scale distributed systems. ⟨http://www.sciencedirect.com/science/article/pii/S0167739×11000689⟩ ⟨http://dx.doi.org/10.1016/j.future.2011.04.017⟩.

Benson, T., Akella, A., Shaikh, A., Sahu, S., 2011. Cloudnaas: A cloud networking platform for enterprise applications. In: Proceedings of the 2nd ACM Symposium on Cloud Computing, SOCC '11, ACM, New York, NY, USA, pp. 8:1-8:13. ⟨http://dx.doi.org/10.1145/2038916.2038924⟩.

Bi, C., Luo, X., Ye, T., Jin, Y., 2013. On precision and scalability of elephant flow detection in data center with sdn. In: Globecom Workshops (GC Wkshps), 2013 IEEE, pp. 1227–1232. ⟨http://dx.doi.org/10.1109/GLOCOMW.2013.6825161⟩.

Bilal, K., Malik, S.U.R., Khalid, O., Hameed, A., Alvarez, E., Wijaysekara, V., Irfan, R., Shrestha, S., Dwivedy, D., Ali, M., Khan, U.S., Abbas, A., Jalil, N., Khan, S.U., 2014. A taxonomy and survey on green data center networks, Future Generation Computer Systems 36 189–208, special Section: Intelligent Big Data ProcessingSpecial Section: Behavior Data Security Issues in Network Information PropagationSpecial Section: Energy-efficiency in Large Distributed Computing ArchitecturesSpecial Section: eScience Infrastructure and Applications. ⟨http://www.sciencedirect.com/science/article/pii/S0167739×13001519⟩ ⟨http://dx.doi.org/10.1016/j.future.2013.07.006⟩.

Biran, O., Corradi, A., Fanelli, M., Foschini, L., Nus, A., Raz, D., Silvera, E., 2012. A stable network-aware vm placement for cloud systems. In: Cluster, Cloud and Grid Computing (CCGrid), 2012 Proceedings of the 12th IEEE/ACM International Symposium on, pp. 498–506. ⟨http://dx.doi.org/10.1109/CCGrid.2012.119⟩.

Bist, M., Wariya, M., Agarwal, A., 2013. Comparing delta, open stack and xen cloud platforms: A survey on open source iaas. In: Advance Computing Conference (IACC), 2013 IEEE 3rd International, pp. 96–100. ⟨http://dx.doi.org/10.1109/IAdCC.2013.6514201⟩.

Blenk, A., Basta, A., Reisslein, M., Kellerer, W., 2016. Survey on network virtualization hypervisors for software defined networking. IEEE Commun. Surv. Tutor. 18 (1), 655–685. http://dx.doi.org/10.1109/COMST.2015.2489183.

Boughzala, B., Ali, R.B., Lemay, M., Lemieux, Y., Cherkaoui, O., 2011. Openflow supporting inter-domain virtual machine migration. In: IFIP International Conference on Wireless and Optical Communications Networks, pp. 1–7. ⟨http://dx.doi.org/10.1109/WOCN.2011.5872945⟩.

Buneman, P., Khanna, S., Tan, W.C., 2001. Why and where: A characterization of data provenance. In: ICDT, pp. 316–330.

Capone A., Cascone, C., Nguyen, A.Q.T., Sansò, B., 2015. Detour planning for fast and reliable failure recovery in sdn with openstate. In: 2015 Proceedings of the 11th International Conference on the Design of Reliable Communication Networks (DRCN), pp. 25–32. ⟨http://dx.doi.org/10.1109/DRCN.2015.7148981⟩.

Cavdar, D., Alagoz, F., 2012. A survey of research on greening data centers. In: Global Communications Conference (GLOBECOM), 2012 IEEE, pp. 3237–3242. ⟨http://dx.doi.org/10.1109/GLOCOM.2012.6503613⟩.

Channegowda, M., Nejabati, R., Simeonidou, D., 2013. Software-defined optical networks technology and infrastructure: enabling software-defined optical network operations [invited]. J. Opt. Commun. Netw. 5 (10), A274–A282. http://dx.doi.org/10.1364/JOCN.5.00A274, ⟨http://jocn.osa.org/abstract.cfm?URI=jocn-5-10-A274⟩ 1⟨http://dx.doi.org/10.1364/JOCN.5.00A274⟩..

Chen, K., Hu, C., Zhang, X., Zheng, K., Chen, Y., Vasilakos, A.V., 2011. Survey on routing in data centers: insights and future directions. IEEE Netw. 25 (4), 6–10. http://dx.doi.org/10.1109/MNET.2011.5958002.

Chen, M., Jin, H., Wen, Y., Leung, V.C.M., 2013. Enabling technologies for future data center networking: a primer. IEEE Netw. 27 (4), 8–15. http://dx.doi.org/10.1109/

MNET.2013.6574659.
Chen, K.y., Junuthula, A.R., Siddhrau, I.K., Xu, Y., Chao, H.J., 2016. Sdnshield: Towards more comprehensive defense against ddos attacks on sdn control plane. In: 2016 IEEE Conference on Communications and Network Security (CNS), pp. 28–36. ⟨http://dx.doi.org/10.1109/CNS.2016.7860467⟩.
Chen, K.Y., Xu, Y., Xi, K., Chao, H.J., 2013. Intelligent virtual machine placement for cost efficiency in geo-distributed cloud systems. In: 2013 IEEE International Conference on Communications (ICC), pp. 3498–3503. ⟨http://dx.doi.org/10.1109/ICC.2013.6655092⟩.
Cui, Y., Xiao, S., Liao, C., Stojmenovic, I., Li, M., 2013. Data centers as software defined networks: traffic redundancy elimination with wireless cards at routers. IEEE J. Sel. Areas Commun. 31 (12), 2658–2672. http://dx.doi.org/10.1109/JSAC.2013.131207.
Curtis, A.R., Mogul, J.C., Tourrilhes, J., Yalagandula, P., Sharma, P., Banerjee, S., 2011. Devoflow: scaling flow management for high-performance networks, In: SIGCOMM'11, pp. 254–265.
Danna, E., Mandal, S., Singh, A., 2012. A practical algorithm for balancing the max-min fairness and throughput objectives in traffic engineering. In: INFOCOM, 2012 Proceedings IEEE, pp. 846–854. ⟨http://dx.doi.org/10.1109/INFCOM.2012.6195833⟩.
Davie, B., Gross, J., Jun. 2016. A stateless transport tunneling protocol for network virtualization (stt), Internet-Draft draft-davie-stt-08, Internet Engineering Task Force, Work in Progress. URL ⟨https://tools.ietf.org/html/draft-davie-stt-08.
Dayarathna, M., Wen, Y., Fan, R., 2016. Data center energy consumption modeling: a survey. IEEE Commun. Surv. Tutor. 18 (1), 732–794. http://dx.doi.org/10.1109/COMST.2015.2481183.
DeCusatis, C., 2014. Optical interconnect networks for data communications. J. Light. Technol. 32 (4), 544–552. http://dx.doi.org/10.1109/JLT.2013.2279203.
Dias, D.S., Costa, L.H.M.K., 2012. Online traffic-aware virtual machine placement in data center networks. In: 2012 Global Information Infrastructure and Networking Symposium (GIIS), pp. 1–8. ⟨http://dx.doi.org/10.1109/GIIS.2012.6466665⟩.
Ebrahimirad, V., Goudarzi, M., Rajabi, A., 2015. Energy-aware scheduling for precedence-constrained parallel virtual machines in virtualized data centers. J. Grid Comput. 13 (2), 233–253. http://dx.doi.org/10.1007/s10723-015-9327-x.
Escalona, E., Aznar Baranda, J., Contreras Murillo, L., Gonzalez de Dios, O., Cossu, G., Facca, F., Salvadori, E., 2013. Using sdn for cloud services provisioning: The xifi use-case. In: Future Networks and Services (SDN4FNS), 2013 IEEE SDN for, pp. 1–7. ⟨http://dx.doi.org10.1109/SDN4FNS.2013.6702561⟩.
Fang, W., Liang, X., Li, S., Chiaraviglio, L., Xiong, N., 2013. Vmplanner: optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers. Comput. Netw. 57 (1), 179–196. http://dx.doi.org/10.1016/j.comnet.2012.09.008, (⟨http://dx.doi.org/10.1016/j.comnet.2012.09.008⟩).
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., Oct. 2015. The Locator/ID Separation Protocol (LISP), RFC 6830. ⟨https://rfc-editor.org/rfc/rfc6830.txt⟩ ⟨http://dx.doi.org/10.17487/rfc6830⟩.
Ferguson, A.D., Guha, A., Liang, C., Fonseca, R., Krishnamurthi, S., 2013. Participatory networking: an API for application control of sdns. SIGCOMM, Comput. Commun. Rev. 43 (4), 327–338. http://dx.doi.org/10.1145/2534169.2486003.
Foundation, O.N., 2013. Openflow switch specification ver 1.4. Tech. rep., Open Networking Foundation.
Georgiou, S., Tsakalozos, K., Delis, A., 2013. Exploiting network-topology awareness for vm placement in iaas clouds. In: 2013 Proceedings of the Third International Conference on Cloud and Green Computing (CGC), pp. 151–158. ⟨http://dx.doi.org/10.1109/CGC.2013.30⟩.
Ghobadi, M., Yeganeh, S.H., Ganjali, Y., 2012. Rethinking end-to-end congestion control in software-defined networks. In: Proceedings of the 11th ACM Workshop on Hot Topics in Networks, HotNets-XI, ACM, New York, NY, USA, pp. 61–66. ⟨http://dx.doi.org/10.1145/2390231.2390242.
Guo, Z., Duan, Z., Xu, Y., Chao, H.J., 2013. Cutting the electricity cost of distributed datacenters through smart workload dispatching. IEEE Commun. Lett. 17 (12), 2384–2387. http://dx.doi.org/10.1109/LCOMM.2013.102213.131831.
Guo, Z., Duan, Z., Xu, Y., Chao, H.J., 2014. Jet: electricity cost-aware dynamic workload management in geographically distributed datacenters. Comput. Commun. 50, 162–174, (URL ⟨http://dblp.uni-trier.de/db/journals/comcom/comcom50.html#GuoDXC14⟩).
Guo, Z., Xu, Y., Cello, M., Zhang, J., Wang, Z., Liu, M., Chao, H.J., 2015. Jumpflow. Comput. Netw. 92 (P2), 300–315. http://dx.doi.org/10.1016/j.comnet.2015.09.030.
Guo, J., Liu, F., 2014. On efficient bandwidth allocation for traffic variability in datacenters. In: INFOCOM, pp. 0–0.
Guo, J., Liu, F., Tang, H., Lian, Y., Jin, H., Lui, J., 2013. Falloc: Fair network bandwidth allocation in iaas datacenters via a bargaining game approach. In: 2013 Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP), pp. 1–10. ⟨http://dx.doi.org/10.1109/ICNP.2013.6733583⟩.
H.T., Google architecture., 2008. ⟨http://highscalability.com/google-architecture. (Accessed 20 January 2016).
Hakiri, A., Gokhale, A., Berthou, P., Schmidt, D.C., Gayraud, T., 2014. Software-defined networking: challenges and research opportunities for future internet. Comput. Netw. 75, 453–471.
Hamilton, R., Iyengar, J., Swett, I., Wilk, A., Jul. 2016. QUIC: A UDP-based secure and reliable transport for HTTP/2, Internet-Draft draft-hamilton-early-deployment-quic-00, Internet Engineering Task Force, Work in Progress. URL ⟨https://tools.ietf.org/html/draft-hamilton-early-deployment-quic-00⟩.
Hammadi, A., Mhamdi, L., 2014. A survey on architectures and energy efficiency in data center networks. Comput. Commun. 40, 1–21, (URL ⟨http://www.sciencedirect.com/science/article/pii/S0140366413002727⟩ ⟨http://dx.doi.org/10.1016/j.comcom.2013.11.005⟩.).
Hao, F., Lakshman, T.V., Mukherjee, S., Song, H., 2009. Enhancing dynamic cloud-based services using network virtualization. Comput. Commun. Rev. 40, 37–44. http://dx.doi.org/10.1145/1592648.1592655.
Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., McKeown, N., 2010. Elastictree: Saving energy in data center networks. In: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, NSDI'10, USENIX Association, Berkeley, CA, USA, pp. 17–17.
Hong, C.-Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., Wattenhofer, R., 2013. Achieving high utilization with software-driven wan. SIGCOMM Comput. Commun. Rev. 43 (4), 15–26. http://dx.doi.org/10.1145/2534169.2486012.
Hu, Y.-L., Su, W.-B., Wu, L.-Y., Huang, Y., Kuo, S.-Y., 2013. Design of event-based intrusion detection system on openflow network. In: 2013 Proceedings of the 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 1–2. ⟨http://dx.doi.org/10.1109/DSN.2013.6575335⟩.
Iyer, A., Kumar, P., Mann, V., 2014. Avalanche: data center multicast using software defined networking. In: 2014 Proceedings of the Sixth International Conference on Communication Systems and Networks (COMSNETS), pp. 1–8. ⟨http://dx.doi.org/10.1109/COMSNETS.2014.6734903⟩.
Jain, R., Paul, S., 2013. Network virtualization and software defined networking for cloud computing: a survey. IEEE Commun. Mag. 51 (11), 24–31. http://dx.doi.org/10.1109/MCOM.2013.6658648.
Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S., Vahdat, A., 2013. B4: experience with a globally-deployed software defined wan. SIGCOMM Comput. Commun. Rev. 43 (4), 3–14. http://dx.doi.org/10.1145/2534169.2486019.
Jakab, L., Cabellos-Aparicio, A., Coras, F., Saucez, D., Bonaventure, O., 2010. Lisp-tree: a dns hierarchy to support the lisp mapping system. IEEE J. Sel. Areas Commun. 28 (8), 1332–1343. http://dx.doi.org/10.1109/JSAC.2010.101011.
Ji, P., Qian, D., Sethuraman, K., Hu, J., Aono, Y., Tajima, T., Blakney, W., Wang, T., Xia, T., Wellbrock, G., 2013. First demonstration of real-time all-optical software-defined intra-data center star network using ofdm and burst switching. In: OptoElectronics and Communications Conference held jointly with 2013 International Conference on Photonics in Switching (OECC/PS), 2013 18th, pp. 1–2.
Jouet, S., Pezaros, D., 2013. Measurement-based tcp parameter tuning in cloud data centers. In: 2013 Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP), , pp. 1–3. ⟨http://dx.doi.org/10.1109/ICNP.2013.6733644⟩.
Juliadotter, N.V., Choo, K.K.R., 2015. Cloud attack and risk assessment taxonomy. IEEE Cloud Comput. 2 (1), 14–20. http://dx.doi.org/10.1109/MCC.2015.2.
Kachris, C., Tomkos, I., 2012. A survey on optical interconnects for data centers. IEEE Commun. Surv. Tutor. 14 (4), 1021–1036. http://dx.doi.org/10.1109/SURV.2011.122111.00069.
Kanagavelu R., Lee, B.S., Felipe Miguel, R., Dat, L., Mingjie, L., 2013. Software defined network based adaptive routing for data replication in data centers, In: Proceedings of the 19th IEEE International Conference on Networks (ICON), 2013, pp. 1–6. ⟨http://dx.doi.org/10.1109/ICON.2013.6781967⟩.
Kannan, K., Banerjee, S., 2012. Scissors: Dealing with header redundancies in data centers through sdn. In: Network and service management (cnsm), In: 2012 Proceedings of the 8th international conference and 2012 workshop on systems virtualiztion management (svm), pp. 295–301.
Kawashima, R., Matsuo, H., 2013. Non-tunneling edge-overlay model using openflow for cloud datacenter networks. In: Proceedings of the 5th International Conference on Cloud Computing Technology and Science (CloudCom), 2013 IEEE, vol. 2, pp. 176–181. ⟨http://dx.doi.org/10.1109/CloudCom.2013.122⟩.
Ke, B.-Y., Tien, P.-L., Hsiao, Y.-L., 2013. Parallel prioritized flow scheduling for software defined data center network. In: 2013 IEEE Proceedings of the 14th International Conference on High Performance Switching and Routing (HPSR), pp. 217–218. ⟨http://dx.doi.org/10.1109/HPSR.2013.6602317⟩.
Kempf, J., Zhang, Y., Mishra, R., Beheshti, N., 2013. Zeppelin – a third generation data center network virtualization technology based on sdn and mpls. In: 2013 IEEE Proceedings of the 2nd International Conference on Cloud Networking (CloudNet), pp. 1–9. ⟨http://dx.doi.org/10.1109/CloudNet.2013.6710551⟩.
Kent, S., Seo, K., Security architecture for the internet protocol, RFC 4301.
Khurshid, A., Zou, X., Zhou, W., Caesar, M., Godfrey, P.B., Veriflow: Verifying network-wide invariants in real time, In: Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, nsdi'13, USENIX Association, Berkeley, CA, USA, pp. 15–28.
Kliazovich, D., Arzo, S.T., Granelli, F., Bouvry, P., Khan, S.U., 2013. e-stab: Energy-efficient scheduling for cloud computing applications with traffic load balancing. In: Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, pp. 7–13. ⟨http://dx.doi.org/10.1109/GreenCom-iThings-CPSCom.2013.28⟩.
Kliazovich, D., Bouvry, P., Khan, S.U., 2010. Dens: Data center energy-efficient network-aware scheduling. In: Green Computing and Communications (GreenCom), 2010 IEEE/ACM International'l Conference on International'l Conference on Cyber, Physical and Social Computing (CPSCom), pp. 69–75. ⟨http://dx.doi.org/10.1109/GreenCom-CPSCom.2010.31⟩.
Koerner, M., Kao, O., 2014. Oftables: A distributed packet filter. In: Proceedings of the Sixth International Conference on Communication Systems and Networks (COMSNETS), 2014 , pp. 1–4. ⟨http://dx.doi.org/10.1109/COMSNETS.2014.6734922⟩.
L. Popa, A. Krishnamurthy, S. Ratnasamy, I. Stoica, Faircloud: Sharing the network in cloud computing. In: Proceedings of the 10th ACM Workshop on Hot Topics in Networks, HotNets-X, ACM, New York, NY, USA, 2011, pp. 22:1-22:6. ⟨http://dx.doi.org/10.1145/2070562.2070584⟩.
Lago, D.G.D., Madeira, E.R.M., Bittencourt, L.F., 2011. Power-aware virtual machine

scheduling on clouds using active cooling control and dvfs. In: Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science, MGC '11, ACM, New York, NY, USA, pp. 2:1-2:6. ⟨http://dx.doi.org/10.1145/2089002.2089004⟩.

Lara, A., Kolasani, A., Ramamurthy, B., 2014. Network innovation using openflow: a survey. IEEE Commun. Surv. Tutor. 16 (1), 493–512. http://dx.doi.org/10.1109/SURV.2013.081313.00105.

Lasserre, M., Kompella, V. Virtual private lan service (vpls) using label distribution protocol (ldp) signaling, RFC 4762.

Li, D., Shang, Y., Chen, C., 2014. Software defined green data center network with exclusive routing. In: INFOCOM, pp. 0–0.

Lin, W.-C., Liao, C.-H., Kuo, K.-T., Wen, C.-P., 2013. Flow-and-vm migration for optimizing throughput and energy in sdn-based cloud datacenter. In: Proceedings of the 5th International Conference on Cloud Computing Technology and Science (CloudCom), 2013 IEEE , vol. 1, 2013, pp. 206–211. ⟨http://dx.doi.org/10.1109/CloudCom.2013.35⟩.

Liu, H.H., Wu, X., Zhang, M., Yuan, L., Wattenhofer, R., Maltz, D., 2013. zupdate: updating data center networks with zero loss. SIGCOMM, Comput. Commun. Rev. 43 (4), 411–422. http://dx.doi.org/10.1145/2534169.2486005.

Liu, Z., Li, Y., Su, L., Jin, D., Zeng, L., 2013. M2cloud: software defined multi-site data center network control framework for multi-tenant. SIGCOMM Comput. Commun. Rev. 43 (4), 517–518. http://dx.doi.org/10.1145/2534169.2491725.

Liu, J., Jin, D., Software defined live virtual machine migration. In:Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP), 2013 , 2013, pp. 1–3. ⟨http://dx.doi.org/10.1109/ICNP.2013.6733642⟩.

Long, H., Shen, Y., Guo, M., Tang, F., 2013. Laberio: Dynamic load-balanced routing in openflow-enabled networks. In: Proceedings of the 27th International Conference on Advanced Information Networking and Applications (AINA), 2013 IEEE , pp. 290–297. ⟨http://dx.doi.org/10.1109/AINA.2013.7⟩.

Luo, M.-Y., Chen, J.-Y., 2013. Software defined networking across distributed datacenters over cloud, In: Proceedings of the 5th International Conference on Cloud Computing Technology and Science (CloudCom). 2013 IEEE, vol. 1, pp. 615–622. ⟨http://dx.doi.org/10.1109/CloudCom.2013.87⟩.

Macapuna, C., Rothenberg, C., Magalhas, M., 2010. In-packet bloom filter based data center networking with distributed openflow controllers. In: GLOBECOM Workshops (GC Wkshps), 2010 IEEE, pp. 584–588. ⟨http://dx.doi.org/10.1109/GLOCOMW.2010.5700387⟩.

Mahadevan, P., Banerjee, S., Sharma, P., Shah, A., Ranganathan, P., 2011. On energy efficiency for enterprise and data center networks. IEEE Commun. Mag. 49, 94–100.

Mahalingam, M., Sridhar, T., Bursell, M., Kreeger, L., Wright, C., Duda, K., Agarwal, P., Dutt, D., 2015. Virtual extensible local area network (vxlan): A framework for overlaying virtualized layer 2 networks over layer 3 networks, RFC 7348. ⟨https://rfc-editor.org/rfc/rfc7348.txt⟩ ⟨http://dx.doi.org/10.17487/rfc7348⟩.

Malacarne, A., Paolucci, F., Cugini, F., Mastropaolo, A., Bottari, G., Potì, L., 2014. Multiplexing of asynchronous and independent ask and psk transmissions in sdn-controlled intra-data center network. J. Light. Technol. 32 (9), 1794–1800. http://dx.doi.org/10.1109/JLT.2014.2312536.

Mann, V., Dutta, P., Kalyanaraman, S., 2011. Vmflow: leveraging vm mobility to reduce network power costs in data centers. Networking, 198–211. http://dx.doi.org/10.1007/978-3-642-20757-0_16.

Mann, V., Kannan, K., Vishnoi, A., Iyer, A., 2013. Ncp: Service replication in data centers through software defined networking. In: International Symposium on Integrated Network Management (IM 2013), 2013 IFIP/IEEE, pp. 561–567.

Mann, V., Vishnoi, A., Kannan, K., Kalyanaraman, S., 2012. Crossroads: Seamless vm mobility across data centers through software defined networking. In: Network Operations and Management Symposium (NOMS), 2012 IEEE, pp. 88–96. ⟨http://dx.doi.org/10.1109/NOMS.2012.6211886⟩.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J., 2008. Openflow: enabling innovation in campus networks. SIGCOMM J. Turn. Openflow: 38 (2), 69–74. http://dx.doi.org/10.1145/1355734.1355746.

Meng, X., Pappas, V., Zhang, L., 2010. Improving the scalability of data center networks with traffic-aware virtual machine placement. In: INFOCOM, 2010 Proceedings IEEE, pp. 1–9. ⟨http://dx.doi.org/10.1109/INFCOM.2010.5461930⟩.

Nejabati, R., Peng, S., Simeonidou, D., 2013. Role of optical network infrastructure virtualization in data center connectivity and cloud computing. In: Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC), 2013, pp. 1–3.

Niranjan Mysore, R., Pamboris, A., Farrington, N., Huang, N., Miri, P., Radhakrishnan, S., Subramanya, V., Vahdat, A., 2009. Portland: a scalable fault-tolerant layer 2 data center network fabric. SIGCOMM Comput. Commun. Rev. 39 (4), 39–50, ⟨http://dx.doi.org/10.1145/1594977.1592575⟩..

Nunes, B.A.A., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T., 2014. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Commun. Surv. Tutor. 16 (3), 1617–1634. http://dx.doi.org/10.1109/SURV.2014.012214.00180.

Nunes, R., Pontes, R., Guedes, D., Virtualized network isolation using software defined networks. In: Proceedings of the 38th Conference on Local Computer Networks (LCN), 2013 IEEE, pp. 683–686. ⟨http://dx.doi.org/10.1109/LCN.2013.6761310⟩.

Pandey, V., 2013. Towards widespread sdn adoption: Need for synergy between photonics and sdn within the data center. In: Photonics Society Summer Topical Meeting Series, 2013 IEEE, pp. 227–228. ⟨http://dx.doi.org/10.1109/PHOSST.2013.6614478⟩.

Pascual, J.A., Lorido-Botrán, T., Miguel-Alonso, J., Lozano, J.A., 2015. Towards a greener cloud infrastructure management using optimized placement policies. J. Grid Comput. 13 (3), 375–389. http://dx.doi.org/10.1007/s10723-014-9312-9.

Peng, S., Guo, B., Jackson, C., Nejabati, R., Agraz, F., Spadaro, S., Bernini, G., Ciulli, N., Simeonidou, D., 2015. Multi-tenant software-defined hybrid optical switched data centre. J. Light. Technol. 33 (15), 3224–3233. http://dx.doi.org/10.1109/JLT.2015.2438398.

Perelló, J., Spadaro, S., Ricciardi, S., Careglio, D., Peng, S., Nejabati, R., Zervas, G., Simeonidou, D., Predieri, A., Biancani, M., Dorren, H.J.S., Lucente, S.D., Luo, J., Calabretta, N., Bernini, G., Ciulli, N., Sancho, J.C., Iordache, S., Farreras, M., Becerra, Y., Liou, C., Hussain, I., Yin, Y., Liu, L., Proietti, R., 2013. All-optical packet/circuit switching-based data center network for enhanced scalability, latency, and throughput. IEEE Netw. 27 (6), 14–22. http://dx.doi.org/10.1109/MNET.2013.6678922.

Pisa, P.S., Fernandes, N.C., Carvalho, H.E.T., Moreira, M.D.D., Campista, M.E.M., Costa, L.H.M.K., Duarte, O.C.M.B., 2010. OpenFlow and Xen-Based Virtual Network Migration, Vol. 327 of IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg, Ch. Communications: Wireless in Developing Countries and Networks of the Future, pp. 170–181.

Popa, L., Yalagandula, P., Banerjee, S., Mogul, J.C., Turner, Y., Santos, J.R., 2013. Elasticswitch: Practical work-conserving bandwidth guarantees for cloud computing. In: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13, ACM, New York, NY, USA, pp. 351–362. ⟨http://dx.doi.org/10.1145/2486001.2486027⟩.

Pries, R., Jarschel, M., Goll, S., 2012. On the usability of openflow in data center environments. In: 2012 IEEE International Conference on Communications (ICC), pp. 5533–5537. ⟨http://dx.doi.org/10.1109/ICC.2012.6364891⟩.

Rahman, N.H.A., Choo, K.-K.R., 2015. A survey of information security incident handling in the cloud. Comput. Secur. 49, 45–69, (URL ⟨http://www.sciencedirect.com/science/article/pii/S0167404814001680⟩ ⟨http://dx.doi.org/10.1016/j.cose.2014.11.006⟩).

Ramos, R., Martinello, M., Esteve Rothenberg. C., 2013. Slickflow: Resilient source routing in data center networks unlocked by openflow. In: 2013 IEEE Proceedings of the 38th Conference on Local Computer Networks (LCN), pp. 606–613. ⟨http://dx.doi.org/10.1109/LCN.2013.6761297⟩.

Ranjana, R., Raja, J., A survey on power aware virtual machine placement strategies in a cloud data center, In: 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. 747–752. ⟨http://dx.doi.org/10.1109/ICGCE.2013.6823533⟩.

Rodriguez-Natal, A., Portoles-Comeras, M., Ermagan, V., Lewis, D., Farinacci, D., Maino, F., Cabellos-Aparicio, A., 2015. Lisp: a southbound sdn protocol? IEEE Commun. Mag. 53 (7), 201–207. http://dx.doi.org/10.1109/MCOM.2015.7158286.

Rosa, R.V., Rothenberg, C.E., Madeira, E., 2014. Virtual data center networks embedding through software defined networking. In: Network Operations and Management Symposium (NOMS), IEEE, 2014, pp. 1–5. ⟨http://dx.doi.org/10.1109/NOMS.2014.6838352⟩.

Rotsos, C., Mortier, R., Madhavapeddy, A., Singh, B., Moore, A.W., 2012. Cost, performance & flexibility in openflow: Pick three. In: ICC, pp. 6601–6605. ⟨http://dx.doi.org/10.1109/ICC.2012.6364690⟩.

Sadasivarao, A., Syed, S., Pan, P., Liou, C., Monga, I., Guok, C., Lake, A., 2013. Bursting data between data centers: Case for transport sdn. In: High-Performance Interconnects (HOTI), 2013 IEEE Proceedings of the 21st Annual Symposium on, pp. 87–90. ⟨http://dx.doi.org/10.1109/HOTI.2013.20⟩.

Sankaran, G.C., Sivalingam, K.M., 2016. Optical traffic grooming-based data center networks: node architecture and comparison. IEEE J. Sel. Areas Commun. 34 (5), 1618–1630. http://dx.doi.org/10.1109/JSAC.2016.2520214.

Scott-Hayward, S., O'Callaghan, G., Sezer, S., 2013. Sdn security: A survey. In: Future Networks and Services (SDN4FNS), 2013 IEEE SDN for, pp. 1–7. ⟨http://dx.doi.org/10.1109/SDN4FNS.2013.6702553⟩.

Sharkh, M.A., Ouda, A., Shami, A., 2013. A resource scheduling model for cloud computing data centers. In: 2013 Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 213–218. ⟨http://dx.doi.org/10.1109/IWCMC.2013.6583561⟩.

Sherwood, R., Gibb, G., Yap, K.-K., Appenzeller, G., Casado, M., McKeown, N., Parulkar, G., Flowvisor: A network virtualization layer, Tech. Rep. OPENFLOW-TR-2009-1.

Shieh, A., Kandula, S., Greenberg, A., Kim, C., Saha, B., 2011. Sharing the data center network. In: NSDI, pp. 23–23.

Shimonishi, H., Shinohara, Y., Chiba, Y., 2013. Vitalizing data-center networks using openflow. In: Photonics Society Summer Topical Meeting Series, 2013 IEEE, pp. 250–251. ⟨http://dx.doi.org/10.1109/PHOSST.2013.6614565⟩.

da Silva, R.A.C., da Fonseca, N.L.S., 2016. Topology-aware virtual machine placement in data centers. J. Grid Comput. 14 (1), 75–90. http://dx.doi.org/10.1007/s10723-015-9343-x.

Srikantaiah, S., Kansal, A., Zhao, F., 2008. Energy aware consolidation for cloud computing. In: Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08, USENIX Association, Berkeley, CA, USA, pp. 10–10.URL ⟨http://dl.acm.org/citation.cfm?Id=1855610.1855620⟩.

Subashini, S., Kavitha, V., 2011. A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appl. 34 (1), 1–11, ⟨http://www.sciencedirect.com/science/article/pii/S1084804510001281⟩ ⟨http://dx.doi.org/10.1016/j.jnca.2010.07.006⟩.

Tavakoli, A., Casado, M., Koponen, T., Shenker, S., 2009. Applying nox to the datacenter. In: Proceedings of workshop on Hot Topics in Networks (HotNets-VIII), p. 6.

Thanh, N.H., Cuong, B.D., Thien, T.D., Nam, P.N., Thu, N.Q., Huong, T.T., Nam, T.M., Ecodane: A customizable hybrid testbed for green data center networks, In: International Conference on Advanced Technologies for Communications (ATC), 2013 , 2013, pp. 312–317. ⟨http://dx.doi.org/10.1109/ATC.2013.6698128⟩.

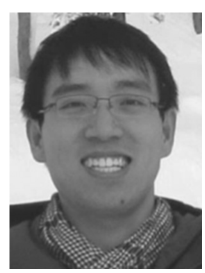
Thanh, N.H., Nam, P.N., Truong, T.-H., Hung, N.T., Doanh, L.K., Pries, R., 2012. Enabling experiments for energy-efficient data center networks on openflow-based

platform. In: Proceedings of the Fourth International Conference on Communications and Electronics (ICCE), 2012 , pp. 239–244. ⟨http://dx.doi.org/10.1109/CCE.2012.6315905⟩.
Tso, F.P., Pezaros, D., 2013. Baatdaat: Measurement-based flow scheduling for cloud data centers. In: 2013 IEEE Symposium on Computers and Communications (ISCC), , 2013, pp. 000765–000770. ⟨http://dx.doi.org/10.1109/ISCC.2013.6755041⟩.
Wang, B., Qi, Z., Ma, R., Guan, H., Vasilakos, A.V., 2015. A survey on data center networking for cloud computing. Comput. Netw. 91, 528–547.
Wang, G., Ng, T.E., Shaikh, A., 2012. Programming your network at run-time for big data applications. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN '12, ACM, New York, NY, USA, pp. 103–108. ⟨http://dx.doi.org/10.1145/2342441.2342462⟩.
Wang, X., Liu, Z., Qi, Y., Li, J., 2012. Livecloud: a lucid orchestrator for cloud datacenters. In: 2012 IEEE Proceedings of the 4th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 341–348. ⟨http://dx.doi.org/10.1109/CloudCom.2012.6427544⟩.
Wang, X., Yao, Y., Wang, X., Lu, K., Cao, Q., 2012. Carpo: Correlation-aware power optimization in data center networks. In: INFOCOM, 2012 Proceedings IEEE, pp. 1125–1133. ⟨http://dx.doi.org/10.1109/INFCOM.2012.6195471⟩.
Wang, Y.-S., Garg, P., Oct. 2015. Nvgre: Network virtualization using generic routing encapsulation, RFC 7637. ⟨https://rfc-editor.org/rfc/rfc7637.txt⟩ ⟨http://dx.doi.org/10.17487/rfc7637⟩.
Widjaja, I., Walid, A., Luo, Y., Xu, Y., Jonathan Chao, H., 2014a. The importance of switch dimension for energy-efficient datacenter design. Comput. Commun. 50, 152–161. http://dx.doi.org/10.1016/j.comcom.2014.02.017.
Widjaja, I., Walid, A., Luo, Y., Xu, Y., Chao, H.J., 2014b. Switch sizing for energy-efficient datacenter networks. SIGMETRICS Perform. Eval. Rev. 41 (3), 98–100. http://dx.doi.org/10.1145/2567529.2567558.
Widjaja, I., Walid, A., Luo, Y., Xu, Y., Chao, H.J., 2013. Small versus large: Switch sizing in topology design of energy-efficient data centers. In: 2013 IEEE/ACM Proceedings of the 21st International Symposium on Quality of Service (IWQoS), pp. 1–6. ⟨http://dx.doi.org/10.1109/IWQoS.2013.6550264⟩.
Yan, Q., Yu, F.R., Gong, Q., Li, J., 2016. Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: a survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. 18 (1), 602–622. http://dx.doi.org/10.1109/COMST.2015.2487361.
Yan, B., Xu, Y., Xing, H., Xi, K., Chao, H.J., 2014. Cab: A reactive wildcard rule caching system for software-defined networks. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, HotSDN '14, ACM, New York, NY, USA, pp. 163–168. ⟨http://dx.doi.org/10.1145/2620728.2620732⟩.
Yang, H., Zhao, Y., Zhang, J., Wang, S., Gu, W., Han, J., Lin, Y., Lee, Y., 2013. Multi-stratum resources integration for data center application based on multiple openflow controllers cooperation. In: Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013, Optical Society of America, p. NTu3F.7. ⟨http://www.osapublishing.org/abstract.cfm?URI=NFOEC-2013-NTu3F.7⟩ ⟨http://dx.doi.org/10.1364/NFOEC.2013.NTu3F.7⟩.
Yin, Y., Liu, L., Proietti, R., Yoo, S.J.B., 2017. Software defined elastic optical networks for cloud computing. IEEE Netw. 31 (1), 4–10. http://dx.doi.org/10.1109/MNET.2016.1300091NM.
Zhang, J., Ren, F., Lin, C., 2013. Survey on transport control in data center networks. IEEE Netw. 27 (4), 22–26. http://dx.doi.org/10.1109/MNET.2013.6574661.
Zhang, J., Zhao, Y., Yu, X., Zhang, J., Song, M., Ji, Y., Mukherjee, B., 2015. Energy-efficient traffic grooming in sliceable-transponder-equipped ip-over-elastic optical networks *invited*. J. Opt. Commun. Netw. 7 (1), A142–A152. http://dx.doi.org/10.1364/JOCN.7.00A142, ⟨http://jocn.osa.org/abstract.cfm?URI=jocn-7-1-A142⟩.
Zhang, S., Tornatore, M., Shen, G., Zhang, J., Mukherjee, B., 2014. Evolving traffic grooming in multi-layer flexible-grid optical networks with software-defined elasticity. J. Light. Technol. 32 (16), 2905–2914. http://dx.doi.org/10.1109/JLT.2014.2317576.
Zhang, Y., Beheshti, N., Beliveau, L., Lefebvre, G., Manghirmalani, R., Mishra, R., Patneyt, R., Shirazipour, M., Subrahmaniam, R., Truchan, C., Tatipamula, M., 2013. Steering: a software-defined networking for inline service chaining. In: 2013 Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP), pp. 1–10. ⟨http://dx.doi.org/10.1109/ICNP.2013.6733615⟩.
Zhao, Y., Zhang, J., Yang, H., Yu, X., 2013. Data center optical networks (dcon) with openflow based software defined networking (sdn). In: 2013 Proceedings of the 8th International ICST Conference on Communications and Networking in China (CHINACOM), pp. 771–775. ⟨http://dx.doi.org/10.1109/ChinaCom.2013.6694698⟩.
Zhao, Y., Zhang, J., Zhou, T., Yang, H., Gu, W., Lin, Y., Han, J., Li, G., Xu, H., 2013. Time-aware software defined networking (ta-sdn) for flexi-grid optical networks supporting data center application. In: Globecom Workshops (GC Wkshps), 2013 IEEE, pp. 1221–1226. ⟨http://dx.doi.org/10.1109/GLOCOMW.2013.6825160⟩.

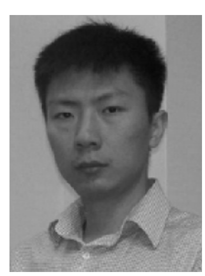
Bin Dai received the B. Eng, the M. Eng degrees and the Ph.D. degree from Huazhong University of Science and Technology of China, P. R. China in 2000, 2002 and 2006, respectively. From 2007–2009, he was a Research Fellow at the City University of Hong Kong. He is currently an associate professor at School of Electronic Information and Communications, Huazhong University of Science and Technology, P. R. China. His research interests include SDN network, wireless network, network coding.

Guan Xu received the B.S. degree in Electronics and Information Engineering from Huazhong University of science and technology, Wuhan, P. R. China, in 2008. He is currently a Ph.D. Candidate in the School of Electronic Information and Communications at the Huazhong University of science and technology. His research interests are in the areas of practical network coding in P2P network, IP switch networks and SDN networks with emphasis on routing algorithms and rate control algorithms.

Benxiong Huang received the B.S. in 1987 and Ph.D. in 2003 from HUST. He is currently a professor at School of Electronic Information and Communications, Huazhong University of Science and Technology, P. R. China. His research interests include signal processing and network security.



Peng Qin received the B. S. degree in Electronics and Information Engineering from Huazhong University of Science and Technology, Wuhan, P. R. China, in 2009. He is currently a Research Fellow in the China Academy of Electronics and Information Technology. He was a visiting student at the University of Victoria in British Columbia, Canada. His research interests are in the areas of network tomography, network measurement, and SDN.



Yang Xu (S′05-M′07) is a Research Associate Professor in the Department of Electrical & Computer Engineering in New York University Polytechnic School of Engineering, where his research interests include Software-Defined Networks, Data Center Networks, Network on Chip, and High Speed Network Security. From 2007–2008, he was a Visiting Assistant Professor in NYU-Poly. Prior to that, he completed a Ph.D. in Computer Science and Technology from Tsinghua University, China in 2007. He received the Master of Science degree in Computer Science and Technology from Tsinghua University in 2003 and Bachelor of Engineering degree from Beijing University of Posts and Telecommunications in 2001.