A Taxonomy of Software-Defined Networking (SDN)-Enabled Cloud Computing

JUNGMIN SON and RAJKUMAR BUYYA, The University of Melbourne, Australia

Software-Defined Networking (SDN) opened up new opportunities in networking with its concept of the segregated control plane from the data-forwarding hardware, which enables the network to be programmable, adjustable, and reconfigurable dynamically. These characteristics can bring numerous benefits to cloud computing, where dynamic changes and reconfiguration are necessary with its on-demand usage pattern. Although researchers have studied utilizing SDN in cloud computing, gaps still exist and need to be explored further. In this article, we propose a taxonomy to depict different aspects of SDN-enabled cloud computing and explain each element in details. The detailed survey of studies utilizing SDN for cloud computing is presented with focus on data center power optimization, traffic engineering, network virtualization, and security. We also present various simulation and empirical evaluation methods that have been developed for SDN-enabled clouds. Finally, we analyze the gap in current research and propose future directions.

CCS Concepts: • **Computer systems organization** → **Cloud computing**; • **Networks** → *Data center networks*; *Network architectures*;

Additional Key Words and Phrases: Cloud computing, software-defined networking, data center networks

ACM Reference format:
Jungmin Son and Rajkumar Buyya. 2018. A Taxonomy of Software-Defined Networking (SDN)-Enabled Cloud Computing. *ACM Comput. Surv.* 51, 3, Article 59 (May 2018), 36 pages.
https://doi.org/10.1145/3190617

1 INTRODUCTION

The emergence of cloud computing has brought a paradigm shift in provisioning computing and networking resources on demand and offering their services on a pay-as-you-go basis. Previously, application providers built their own data center with a vast amount of investment to run the application and deliver it to their customers. This required the upfront cost of purchasing the hardware and installing the infrastructure, which was not scalable or flexible to the increasing demand. Now, with the expansion of the utility computing model [14], they only need to select a cloud provider that fits their applications, or automated brokering systems can dynamically select a suitable resource for hosting applications. Application providers can lease computing resources from cloud providers by clicking a few mouse buttons within minutes and deploy their applications without any upfront payment. Cloud computing has brought scalable and elastic computing along with the subscription-oriented service model.

This work is partially supported by the Australian Research Council (ARC) Future Fellowship and Discovery Project.
Authors' addresses: J. Son and R. Buyya, Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, VIC 3010, Australia; emails: jungmins@student.unimelb.edu.au, rbuyya@unimelb.edu.au.

https://doi.org/10.1145/3190617

To provide elastic service to their customers, cloud providers operate multiple data centers across multiple geolocations with virtualized resource provisioning. Large data centers consist of thousands of switches connecting tens of thousands of servers. Each server can serve multiple application requests from different users by using virtualization technologies, which have been enhanced vastly in processing and storage resources for the last couple of decades but not in network resources. Currently, cloud customers can rent virtualized computing resources, i.e., virtual machines (VMs), and virtualized storages from providers that are allocated virtually for the customer. However, the network resource virtualization technology is still far from being utilized in the commercial public clouds.

Servers are allocated to multiple tenants if the total requested size does not exceed server capacity. Thus, cloud customers are capable of requesting a smaller-sized resource than the capacity of the entire server, and a server can provide multiple VMs [41]. These VMs communicate to each other and the Internet through a number of network switches and routers. The data center network (DCN) is established and managed in order to transfer traffic between VMs efficiently. As the DCN connects tens of thousands of servers, the network will have high complexity, which makes it difficult to manage and scale in the traditional networking paradigm. In traditional networks, each network switch has its own control logic, which individually decides its behavior based on the information obtained from its neighbors. The traditional network approach is inefficient when it comes to the cloud data center, where a higher density of servers provides multiple VMs that dynamically transform from time to time.

To overcome the shortcomings of traditional networks, cloud data centers started adopting the software-defined networking (SDN) concept in their DCNs. SDN provides a centralized control logic with a global view of the entire network at the central controller and dynamically changes the behavior of the network. It can also adjust the network flow dynamically by the controller, which is well fitted for the dynamic nature of the cloud service. Giant cloud providers such as Google already adopted the SDN concept in their data center to increase scalability and manageability [98].

Although many surveys and taxonomies have been presented in cloud computing and SDN contexts, each of them has addressed a specific problem in the area. For example, Toosi et al. [97] presented a survey focusing on interconnected cloud computing. The article includes interoperability scenarios with multiple data centers and detailed explanations of various approaches to operate and use interconnected cloud data centers. The article described networking challenges for interclouds in a subsection, but the primary focus was on the wider issues of integrating multiple cloud data centers from a cloud broker's perspective. Mastelic et al. [64] also presented a survey on energy efficiency for cloud computing. A systematic category of energy consumption in cloud computing has been suggested in the context of hardware and software infrastructure in clouds. The authors also included a networking aspect emphasizing the DCN, inter-data-center network, and end-user network. A comprehensive survey was presented on various aspects of energy efficiency including networks, but the paper was lacking in the SDN context. Jararweh et al. [50] provided details of software-defined clouds focusing on systems, security, storage, and networking, but with more focus on the system architecture rather than the individual research works in SDN clouds.

In this article, both SDN and cloud computing are considered as the survey topic. Among the enormous amount of studies conducted in both the distributed computing and networking disciplines for cloud computing and SDN, respectively, we select the state of the art considering both aspects simultaneously. Although several surveys have been presented in each context separately, this article is the first attempt to the best of our knowledge that explores the research considering both aspects at the same time. We emphasize SDN utilization and challenges in the context of cloud computing.

The rest of the article is organized as follows: In Section 2, we present an overall background of cloud computing, DCN, and SDN. Section 3 provides the description and definition of SDN-enabled cloud computing, followed by a taxonomy of the usage of SDN in cloud computing in various aspects (Section 4). In Section 5, comprehensive surveys have been undertaken to find achievements and challenges in SDN usage in clouds in the context of energy efficiency, performance, virtualization, and security enhancement. Section 6 presents a survey of simulation and empirical methods developed for evaluation of SDN-enabled cloud computing. We investigate the shortcomings of the current research and propose future directions in Section 7 and then summarize and conclude the article.

2 BACKGROUND

Before we start investigating SDN-enabled cloud computing, the fundamentals of cloud computing and data centers are explained. In this section, we briefly describe cloud computing, DCN, and SDN.

2.1 Cloud Computing

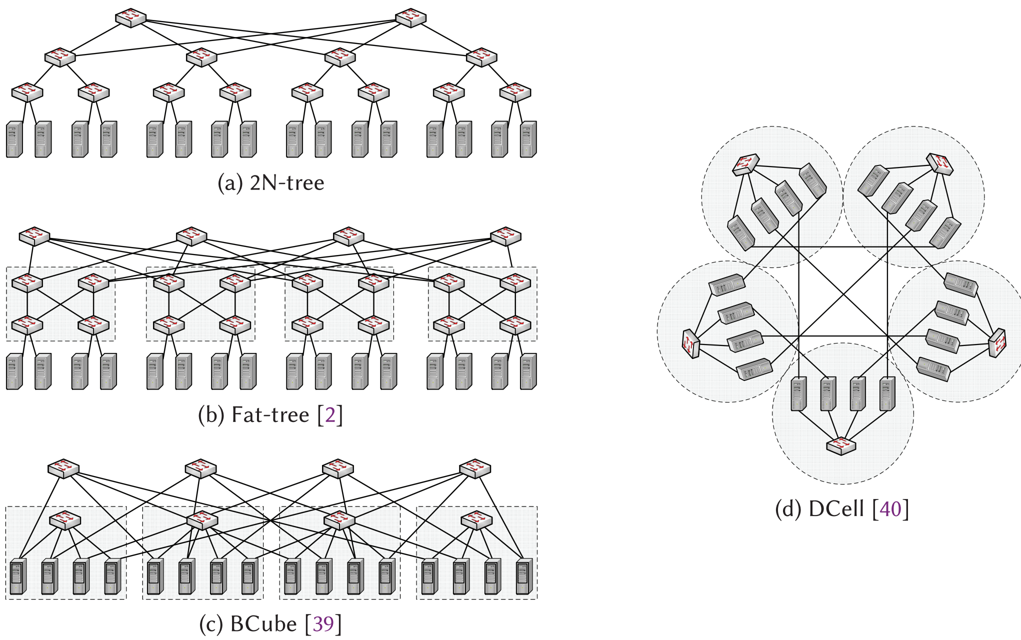
Cloud computing has been studied and implemented for many years and can be considered as three services: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). SaaS provides complete software to cloud customers, which includes cloud-based email services, social network services, scheduler services, and basically any programs running on the cloud. PaaS resides on the underlying layer, where application developers can use the platform services provided by clouds. The software in SaaS may or may not use the platforms provided by the PaaS provider. IaaS is the most fundamental, which provides virtual machine servers and the related infrastructure to cloud customers. The infrastructure can be used for any purpose by the customers, e.g., for deploying their own platforms or developing software running on the virtual machines. SaaS and PaaS can be implemented upon the usage of IaaS.

In order to provide cloud services, the provider should build and maintain one or more large-scale cloud data centers, where tens of thousands of physical hosts are connected through thousands of network switches. With the high complexity of network connections in large scale, the provider has to consider the data center network from a different perspective than the traditional network.

2.2 Data Center Network (DCN)

Host servers in a cloud data center are interconnected through network links and switches, composing the DCN. Typical DCN architecture is made up by a three-tier topology, which consists of edge, aggregate, and core switches on each layer. Host servers within a rack are connected to one or more edge switches, i.e., Top-of-Rack (ToR) switches, and edge switches are connected to upper-tier aggregate switches. Each of the aggregate switches connects to multiple core switches of the top tier.

Since the DCN has a complex structure with a massive number of hosts and networking equipment, performance and scalability issues arise. The high scale brought inflexible and insufficient network bandwidth, which made researchers seek an alternative topology to the canonical 2N-tree. Al-Fares et al. [2] suggested fat-tree topology that shares a similar approach to the Clos network [21] to provide redundant network connection and ensure higher availability. In fat-tree, there are multiple links between switches that lead to a redundant and high-available network. Guo et al. presented DCell [40] to address the scalability and performance issues of the traditional three-tier topology. BCube [39] was also proposed by the same authors, in which hosts serve as both end hosts and relay nodes that transfer network data to another host. BCube was designed for shipping-container-based modular data centers. The aforementioned topologies are shown in



(a) 2N-tree

(b) Fat-tree [2]

(c) BCube [39]

(d) DCell [40]

Fig. 1. Data center network topologies.

Figure 1. More recently, a hypercube-based DCN topology called ExCCC was introduced in [106]. ExCCC substituted each node in the Extended Hypercube network with a cycle and employed link removal for scalability and energy efficiency.

Several survey papers were presented to describe various DCN architectures and topologies [11, 19, 103]. Bari et al. [8] conducted a comprehensive study on the data center network. The authors presented various network topologies for DCN and categorized them based on packet-forwarding schemes, bandwidth guarantee mechanisms, and multipath techniques. The survey is focused on the network topology and architecture for a data center but also discussed the cloud computing context.

2.3 Software-Defined Networking (SDN)

SDN is a new paradigm in networking that manages the entire network at the conceptually centralized programmable software controller. With the central view, the controller can manage network flows more efficiently and dynamically. SDN provides fine-grained network configuration dynamically adaptable to the network condition. Figure 2 shows the conceptual difference between a traditional network and SDN. In the traditional network, the control plane is placed on top of the forwarding plane in each network device. The discrete decision is made by each switch based on the information gathered from the neighbor devices in a distributed matter. Although the network information is shared among devices, the decision is made solely by each device, which can increase the complexity and behavioral unpredictability of the entire network.

On the other hand, SDN centralizes the control plane into a software controller that can observe the entire network. With centrally gathered information, the controller can make more straightforward decisions and deploy it onto forwarding devices efficiently and dynamically. The control logic can be easily programmable in the controller and send control packets to the switches with

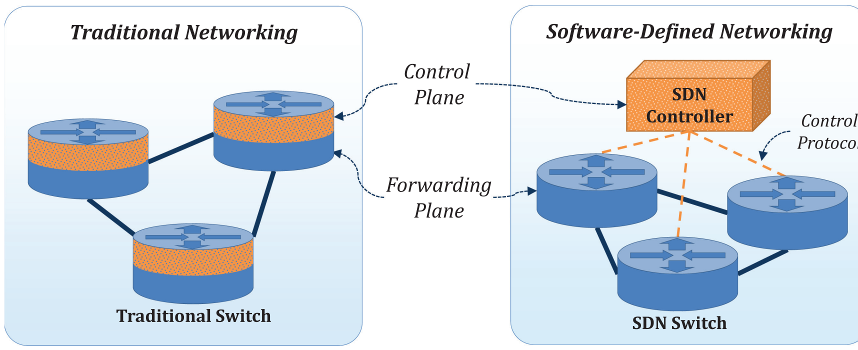


Fig. 2. Comparison of SDN and traditional networking.

SDN protocols. Thus, networking behavior becomes highly customizable depending on different purposes and goals.

As the adhesion between control logic and forwarding hardware has been broken up in SDN, it opened up more opportunities to universities and research institutes in developing new networking protocols and traffic management, virtualization, and software technologies. It led to the introduction of OpenFlow, which is the collective result of a number of universities and became a de facto standard protocol for SDN controllers [68]. Many controllers have been introduced and actively developed based on OpenFlow, including NOX [38], Floodlight [84], RYU [74], and OpenDaylight [77].

SDN has also opened up innovative opportunities by way of developing the control logic of networks. In traditional networks, the network control logic is solely developed by large switch vendors, such as Cisco or Juniper, as it is tightly coupled with the networking hardware. Although individual developers and users could customize the configurations and settings of the hardware, it was still limited to the control logic equipped with the hardware of the vendor. With the introduction of SDN and its decoupled concept of the controlling software from the forwarding hardware, anyone can easily develop and test a new control logic by creating an SDN controller software. As the forwarding hardware is separated and fully controlled by the controller software in SDN, any control logic can be designed and implemented as long as it follows the SDN standard, such as OpenFlow. Thus, SDN has fostered innovations in development of various network control logics that would have been much more limited in traditional networks.

Comprehensive surveys on SDN focusing on the networking perspective have been presented by several researchers [17, 32, 42, 51, 59], in which the emphasis is on SDN's general perspective, with its distinctive features compared to the traditional network. In this article, we concentrate more on SDN integration in cloud computing and discuss the benefits and challenges of SDN usage in clouds.

3 SDN-ENABLED CLOUD COMPUTING

As we described in Section 2, SDN features can bring enormous benefits to cloud computing, such as adapting to dynamically changing workloads, programmable control planes, network security enhancements, network virtualization, a global view of DCN, and integration of hypervisor and cloud computing manager. The nature of cloud computing that supports multitenants in complex networking has formed SDN with the desire of scalability and cost-efficiency [32]. As SDN can manage the network forwarding plane with the centralized control plane, network elements become programmable and controllable dynamically. With the ability of SDN controllers to monitor the entire network, various research projects have been conducted [6, 17, 53, 108] including bandwidth allocation per flow, traffic consolidation, and dynamic configuration, in order to improve QoS and energy efficiency. The dynamics of cloud computing require a quick response to manage traffic demands in real time. In industry, Google started deploying SDN in its cloud data centers and their private wide-area network (WAN) between data centers as described in their papers [47, 48, 98].

3.1 Terms and Definitions

With the emergence of SDN adoption in the cloud computing context, several terms have been presented in order to capture the architectural characteristic of the new system. In this section, we propose the organized terms for different purposes based on the collective survey in the order of narrow to wide scope.

The *SDN-enabled cloud data center (SDN-DC)* provides SDN features within the cloud data center. On the basis of the traditional DCN architecture, the SDN-DC replaces traditional networking equipment with SDN-enabled devices. In the SDN-DC, every networking component in a data center is capable of performing SDN functions, which can bring all the aforementioned benefits to a cloud data center. This architecture focuses within a data center and excludes the internetworking part outside of a data center.

SDN-enabled cloud computing (SDN-clouds or SDN-enabled clouds) refers to not only SDN-DC but also intercloud networking that expends the SDN usage across multiple data centers and wide-area networks (WANs). Thus, SDN benefits can be applied to the internetworking domain. In this article, we focus on SDN-clouds to build our taxonomy and survey.

A broader term, *software-defined cloud computing (SDC or software-defined clouds),* has been proposed by Buyya et al. [13] and Jararweh et al. [50] where not only networking but also all infrastructure components in cloud computing are considered to be software defined. The approach is to build fully automated cloud data-center-optimizing configurations autonomously and dynamically. Buyya et al. [13] extended the concept of virtualization from a virtualized server (VM) to all other elements in cloud data centers. The core technologies to enable SDC include server virtualization, SDN, network virtualization, and virtual middleboxes. With these recently evolved technologies, the reconfiguration and the adaptation of physical resources in SDC have become easier and simpler. The proposed architecture was implemented and evaluated in the simulation environment. Jararweh et al. [50] also studied a system focusing on various perspectives of SDC including networking, storage, virtualization, data centers, and security. The authors also built an experimental setup for SDC with the inspected elements in the survey to show the effectiveness of the proposed software-defined cloud architecture.

SDC is more conceptual as it is proposed for research purposes and has not yet been explored extensively. Therefore, in this survey, we focus on SDN-clouds to depict the state of the art and SDN usage in cloud computing.

3.2 SDN-Cloud Architectures

Figure 3 shows the abstract architecture of common SDN-enabled cloud computing systems derived from the literature [7, 14, 93, 104].

The *cloud manager* is managing the tenants and resources of entire clouds. It controls the incoming requests from tenants such as a VM creation and provisions cloud resources to provide the cloud service. Energy-efficient resource management and monitoring of resources are also performed by the cloud manager. OpenStack [78] is an open-source example of a cloud manager widely used to build private clouds.

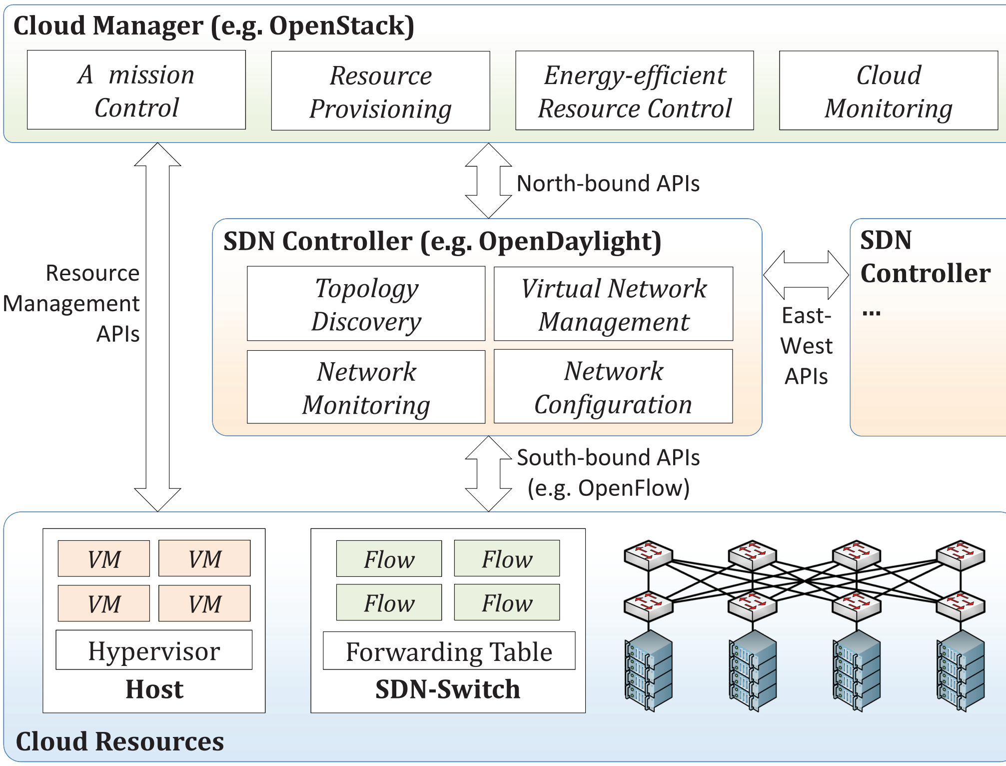


Fig. 3. SDN-enabled cloud computing architecture.

Network-related functions are controlled by the *SDN controller,* which is connected to the cloud manager through north-bound APIs. SDN features, such as network topology discovery, network monitoring, virtual network management, and dynamic network configurations, are enabled through the SDN Controller. Multiple SDN Controllers can be functional to increase scalability by intercommunicating via east-west-bound APIs.

Cloud resources consist of compute and networking resources. Compute resources (hosts) are provisioned by the cloud manager to run VMs on a hypervisor, whereas networking resources (switches) are managed by the SDN controller through updating forwarding tables in switches via south-bound APIs (e.g., OpenFlow). Note that the SDN controller also manages networking functions in hosts, for example, virtual switches, for network virtualization and VM traffic management.

Meridian is an SDN platform for cloud network services proposed by IBM [7]. In typical cloud network services, users are required to set up switches, subnets, and ACLs to be used by cloud applications. Meridian adopts the service-level network model with connectivity and policy abstractions and integrates the high-level application provisioning closely to the cloud network through SDN's programmable network concept.

PDSDN is a policy exchange scheme between the SDN controller and the cloud manager to simplify the interaction of cloud tenants and the controller [28]. The proposed scheme sends network policies to the controller and processes them with the information of tenants' requirements and priorities. The prototype is implemented on Floodlight and validated on OpenStack.

More recently, Mayoral et al. [66] presented cloud architectures with a single SDN controller and multiple controllers. The system orchestrates networking services in a data center with the capability of an SDN. In order to find the differences in one or multiple controllers in a data center, the authors evaluated the proposed architecture on OpenStack and OpenDaylight with the integration of SDN.

The same authors also proposed an end-to-end orchestration architecture of cloud and networks resources that manages both resources in distributed cloud data centers [65]. Virtual infrastructures in clouds are dynamically allocated and managed through the manager. The authors modeled the VM allocation problem as a graph mapping with the parameter of switches, hosts, and network links. The paper also includes a heuristic algorithm to find the minimum number of hosts with enough capacity and selects the shortest path between hosts. The orchestration architecture is validated on an empirical testbed with OpenStack and OpenVSwitch, whereas the algorithm is evaluated in simulation.

Fichera et al. [33] presented an experimental testbed setup for SDN orchestration across edge, cloud, and IoT domains in 5G service. The platform consists of separate orchestrator(s) for the cloud, IoT, and SDN. The SDN orchestrator manages not only IoT and cloud networks but also the transport network between them. On top of individual orchestrators, the service orchestrator oversees service provisioning for applications based on the user's requirements. The proposed system is implemented in a testbed consisting of Mininet for the edge network, OpenStack cloud, and ONOS controller.

The Control Orchestration Protocol (COP) was proposed by researchers from multiple institutions to allow managing heterogeneous SDN control planes and clouds [67]. While OpenFlow specifies control messages between the SDN controller and switches as a southbound interface of the controller, COP is in charge of orchestrating among SDN controllers in different networks as well as cloud controllers working north of controllers. COP aims at a unified transport API for orchestrating different transport networks in the intercloud environment. The protocol provides an abstraction for resource provisioning, topology discovery, and path computation. It is validated through two proof-of-concept experiments for network resource provisioning and recovery.

4 TAXONOMY OF SDN USAGE IN CLOUD COMPUTING

Several methods are proposed for utilizing SDN in cloud computing in different research areas. We carried out a detailed study of research in SDN-enabled cloud computing and propose a taxonomy (shown in Figure 4) to capture various elements of SDN usage. The taxonomy is explored in the context of objective, method scope, target architecture, application model, resource configuration, and evaluation method.

4.1 Objective

Cloud providers deploy SDN in their data centers for different purposes, and researchers consider various objectives in their studies. We categorize the objectives into four types, which are used as main categories to classify the surveyed works in Section 5.

Energy efficiency is one of the most commonly studied objectives for SDN-enabled cloud computing. As cloud data centers have consumed enormous amounts of electricity globally, energy-saving methods for data centers have received more attention over the last decade. The energy consumption of a data center mainly relies on host servers, networking devices, and cooling systems [90] and is proportional to utilization level [79]. If all servers in one server room are not utilized, the servers and the cooling system of the room can be switched off to consume

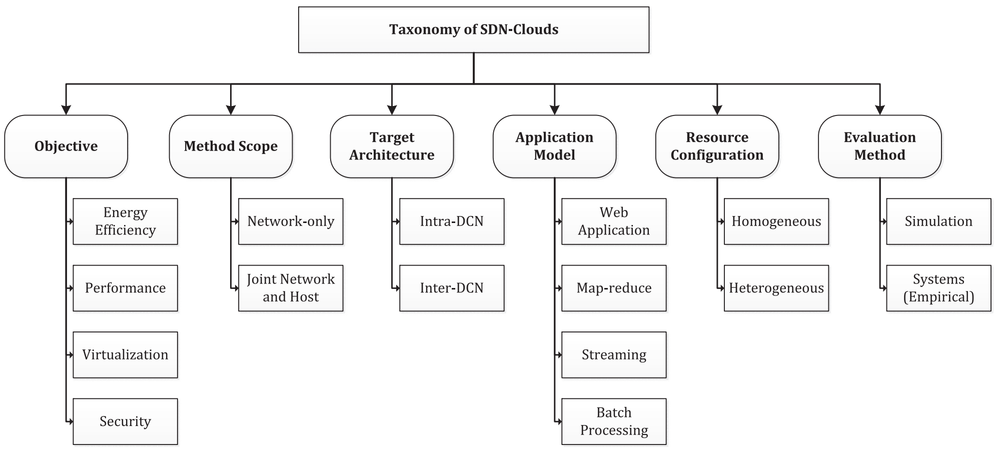


Fig. 4. Taxonomy of SDN usage in cloud computing.

less power. With SDN, network elasticity is possible in addition to computing elasticity. Network traffic can be consolidated into a smaller number of switches with SDN for the duration of low network utilization, which can provide possibilities to turn off unused switches. Not only the network itself but also the joint optimization of network and host is possible for further energy savings.

Performance can be improved by providing fine-grained control over the entire network in SDN. As SDN observes the entire network in the centralized controller, it can easily improve network performance in case of traffic congestion by altering the path of the flows. For instance, SDN can be used as one of the techniques to reduce the VM deployment time in a large-scale data center to boot up a large number of VMs. VM deployment time can be decreased by reducing the amount of data transferred, the bottlenecks at the VM image repository, the CPU load at the target physical machine, or the boot-up time of the VM [89]. The dynamic bandwidth allocation feature between the image repository and physical machines can reduce the image transfer time.

Performance includes not only network throughput and latency but also availability and Quality of Service (QoS). The data center network must be available throughout all operating hours, and SDN can help to increase availability. In case of a hardware malfunction on a network path, the SDN controller can easily notice and modify the following flows to use an alternate path through the programmable switches with appropriate DCN topology configuration. QoS includes providing guaranteed bandwidth to a specific user that was almost impossible in the traditional network as the network medium must be shared by multiple users in cloud data centers. SDN, however, can manipulate the network path to provide dedicated bandwidth to a certain type of user. To improve the QoS, a congested link can be detected, and its traffic can be distributed by changing the path of the network flows. Network performance can be quantified with measurements such as throughput, latency, and error rate. Other works considered the workload acceptance rate and network blocking probability to measure the performance of the resource management scheme [37]. Optimal joint orchestration of cloud and network resources can increase the VM acceptance ratio, which leads to increasing the number of VMs hosting in a cloud data center with limited resources, and thus maximizing the benefit to the cloud provider.

Virtualization can be further realized for DCN by adopting SDN in clouds. Cloud computing is implemented based on a variety of virtualization technologies, which can make it possible to run VMs in clouds [13]. Hypervisors running on physical hosts can virtualize the host resources such as CPU, memory, and storage. SDN is capable of virtualizing network resources. Network resources such as switches and physical links can be virtualized and leased to multiple tenants of a cloud data center. A network embedding problem, for example, is to map the network flows by different tenants into physical resources by virtualizing the infrastructure [34]. Network function virtualization (NFV) is another example of SDN usage for virtualization. NFV was conventionally provided by middleboxes equipped with dedicated hardware, such as firewall, load balancer, and intrusion detectors. The hardware middleboxes are expensive to purchase and difficult to manage and scale. Several researchers attempted to substitute the hardware middleboxes with a series of SDN switches with specific controller programs [85, 92, 99].

Security is an important objective in networks that has not yet been extensively studied in integrating SDN with cloud data centers. There are two research directions in the usage of SDN in clouds: (1) securing networks by utilization of SDN and (2) protecting core components of SDN from attacks. On one hand, SDN technology is widely adopted in cloud data centers to protect and prevent a distributed denial-of-service (DDoS) attack, where overwhelming network traffic is generated to disable online services. By utilizing SDN features such as traffic analysis, centralized control, and dynamic update of network rules, it is easier to detect and react to DDoS attacks [104]. On the other hand, the SDN controller must be protected by a high level of security as its central operation can manage the entire data center network. It is also necessary to secure the management packets communicating between the controller and its switches.

4.2 Method Scope

Here, we propose a taxonomy based on the scope of methodology. As we are discussing SDN-clouds, all methods in our taxonomy are fundamentally targeting the networking functionality in cloud computing. While some researchers consider only networking in their studies, others consider both hosts and networking at the same time.

The scope of *network-only* approaches solely lies within the networking function in cloud computing without consideration of hosts or VMs. These approaches try to collect the network data and alter the forwarding rules in clouds using SDN to solve the research problem. For example, most approaches targeting network performance improvement naturally consider only networking resources in clouds [3, 29, 46, 48].

Joint approaches take not only networking but also hosts and VMs into account simultaneously to orchestrate both resources. Optimizing host and network resources to reduce power consumption of cloud data centers is an example of joint approaches. Before introducing SDN, traditional approaches dominantly consider only hosts for data centers' power optimization [9], whereas joint optimization has been enabled by SDN and recently appealed to many researchers [24, 93, 110]. A joint resource allocation method proposed by Souza et al. [27] also tries to deliver better network performance for VMs hosted in SDN-clouds.

4.3 Target Architecture

As SDN can bring flexibility and controllability to the network with its programmable controller, it applies to various network architectures in cloud data centers.

In many research studies, SDN is applied to the connection between hosts within a data center (*intra-DCN*). SDN-enabled switches replace traditional switches connecting from one host to another. Networking in a data center can be optimized based on the current utilization of network traffic, with the bandwidth usage of incoming and outgoing packets.

We can also find studies focusing on *inter-DCN* architecture, which can optimize network connections between geographically distributed data centers [48, 69, 80]. Cloud providers operating multiple data centers can benefit from SDN by developing their own SDN controller to manage the WAN between the data centers.

Others proposed orchestration methods considering both inter- and intra-DCN architectures [12, 33, 62] to interconnect and manage multiple cloud and/or edge data center networks. The upper-layer controller oversees individual SDN controllers within geographically distributed data centers as well as in transport network interconnecting data centers. With the orchestration of inter- and intra-DCN management, SDN can provide end-to-end QoS for applications and a virtual network environment for VMs hosting in different data centers.

4.4 Application Model

Researchers can consider a specific application model to benefit from SDN on their studies or a common generic approach that does not consider any specific application model. For instance, improving network performance between cloud data centers with SDN is rarely related to the specific application running on the clouds [48]. In contrast, consolidating VMs and their network traffic using correlation analysis of the utilization can be highly relevant to the application running on the VMs [108, 110]. Thus, it is worth considering various application models before exploring the research works.

The *web application* model consists of multiple tiers such as load balancers, application servers, and databases that communicate with the upper- and lower-tier servers. As the end-users of the web application expect a spontaneous response from the Internet, response time and latency are crucial in this model. With the popularity of web applications hosted in clouds, many researchers consider the web application model for energy efficiency research [93, 100, 110].

With the *map-reduce* model, a big task is split into smaller tasks and processed on multiple machines simultaneously. In this model, a large amount of network bandwidth and computing power are necessary to process the workload, where the underlying network support is crucial. For instance, network burst between mappers and reducers can be a bottleneck as the mappers send the processed data to the reducers at the same time if the tasks are equally distributed and processed at a similar speed. With a proper control logic in SDN, the burst traffic can be distributed dynamically, which can eventually decrease the effect of the bottleneck [22].

The *streaming* model is for the application that generates data continuously, which leads to transferring large flows of data across networks. Streaming applications include not only audio and video streaming applications but also the applications for recently introduced Internet of Things appliances, both of which need a seamless network connection for real-time transmission. With the emergence of the Internet of Things (IoT), where every device with sensors can generate and transfer its data to clouds for analysis and data mining, support for streaming applications is acquiring more attention. As streaming applications generate an enormous amount of data continuously, an enhanced methodology is necessary for seamless network transmission and real-time processing in clouds. Researchers have studied both multimedia streaming applications [80] and IoT stream data [101] for SDN-clouds.

In *batch processing*, jobs are processed in order. It may need to transfer data between jobs, in which network transmission is necessary with a constant and reliable connection. In most batch processing tasks such as scientific computation, providing constant bandwidth is more crucial than the latency or the response time. Thus, the network controller should provide a different approach. For example, Seetharam et al. [91] presented the SDN-enabled network management system for multiple clouds in the university campus for scientific workloads.

4.5 Resource Configuration

Resource configuration refers to the heterogeneity of hardware resources composing a cloud data center. Resource configuration should be considered in designing a new method for SDN-clouds, because the results of the method may differ depending on the configuration. For example, a VM consolidation method for energy efficiency designed for homogeneous configuration may not be effective in a heterogeneous environment if the algorithm does not consider the power models of various host specifications.

In a *homogeneous* configuration, all the hardware in the data center is assumed to have the same specifications. Physical machines and switches in an edge rack share the same hardware specifications across the entire data center, and all the racks have an identical number of hosts and switches. Homogeneous configuration is useful for research in energy efficiency in order to simplify energy modeling of hardware, because various hardware specifications make the modeling more complicated.

A *heterogeneous* configuration is set up with different types of hardware. A homogeneous configuration is impractical in the real world, as most data centers would upgrade their systems periodically and purchase newer machines to expand capacity, which results in having a heterogeneous configuration in the end. A homogeneous configuration is useful, though, in a research study with a limited testing environment, which can alleviate the complexity of the evaluation process; thus, many researchers studied with a homogeneous hardware configuration for their preliminary experiments. Research on network performance for inter-DCN generally considers a heterogeneous configuration as the internetworking system is involved with various networking operators who use different networking devices. Some works also include the heterogeneous configuration for energy efficiency research in order to model the power consumption of various switch devices [44].

4.6 Evaluation Method

Evaluating cloud data centers is challenging, with its high complexity made up by a tremendous number of hosts and switches and their connections. In order to test the new approaches with limited research resources, researchers use two main methods for evaluation.

Simulation is the most affordable method to test a new approach with acceptable accuracy. Cloud data centers can be set up in simulators with the arbitrary configuration without much cost. It is also easy to implement the proposed approach in simulation and to evaluate the result quickly. However, the result of the simulation can be inaccurate when the configuration becomes far different from the validated environment, as the simulated results are statistically calculated from certain settings. Several simulation tools have been implemented including Mininet [60] and CloudSimSDN [94], and more details will be explained in Section 6. Evaluation with the network simulator integrated with an empirical SDN controller is also considered as a simulation method, although the developed SDN controller can be adapted to the real system directly, since the evaluation still relies on the simulation tool.

Empirical evaluation can provide more practical results compared to the simulation as the test is being used on the real system. Although the result is more accurate, evaluating in large scale is impractical on the real systems, because the cost can be too expensive and the management of the cloud data center elements can be significantly complex. In several studies including ElasticTree [44], CARPO [100], and CometCloud [80], the empirical approach has been used for evaluation.

5 CURRENT RESEARCH ON SDN USAGE IN CLOUD COMPUTING

In this section, we present relevant studies conducted in SDN-enabled cloud computing following the taxonomy proposed in Section 4. We categorize the state of the art primarily based on the main objective of the article using the taxonomy described in the previous section. A research work typically aims to address more than one objective at the same time with a different priority. We adopt the main objective of the study for the classification in this section. Figure 5 presents

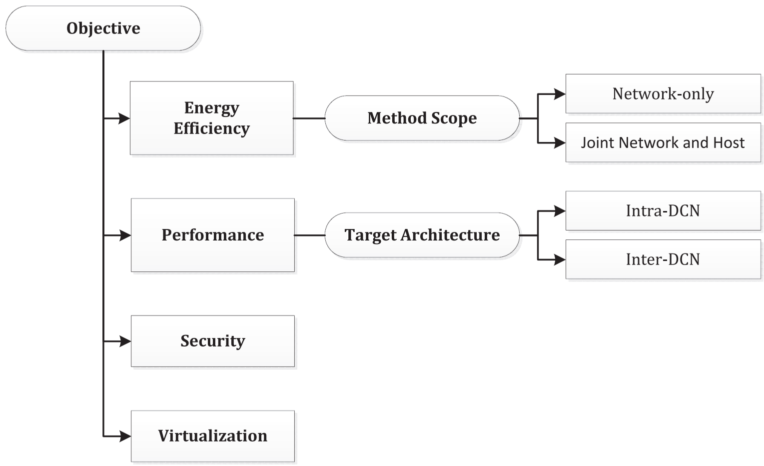


Fig. 5. Subcategories used for our literature review.

subcategories used for further classification of the surveyed studies in this section. The detailed literature in each category is presented later.

5.1 Energy Efficiency

Energy efficiency is one of the utmost research topics in SDN-enabled cloud computing. Based on our observations, SDN is exploited in two ways to improve energy efficiency of cloud data centers: network optimization and joint host-network optimization. Table 1 presents a summary of reviewed studies for energy efficiency.

Network optimization amends network elements of DCN including switches and network adapters in hosts. Link rate adaptation is a similar approach to dynamic voltage and frequency scaling (DVFS) that lowers the link rate of a network device in case of underutilization [1]. Dynamic topology alteration changes the network topology dynamically depending on the network traffic load [44]. Network flows in underutilized links are consolidated into the smaller number of links, resulting in the alteration of the network topology. Correlation-aware optimization considers the correlation of network flows for traffic consolidation [100]. Both dynamic topology alteration and correlation-aware optimization are feasible with the introduction of SDN in cloud data centers.

Joint host-network optimization considers both the host and network simultaneously to reduce the energy consumption for VM and network placement, consolidation, and overbooking. VM placement optimizes the initial placement of VMs in order to minimize the total power consumption of the entire data center. Similarly, VM migration and consolidation consider migration of VMs after the initial placement, which can consolidate VMs into a smaller number of hosts. These techniques often use a host overbooking method that places more VMs into a host than its capacity to maximize resource utilization. Instead of optimizing separately, the combined techniques are applied in order to consolidate both the network and hosts at the same time.

5.1.1 Network Optimization. Several studies have been conducted in optimizing the DCN to reduce the energy consumption of data centers.

ElasticTree [44] is presented that dynamically changes DCN topology and adjusts links and switches for power saving. It is designed for the fat-tree topology to consolidate traffic flows into a minimum number of links to utilize as few switches as possible. After consolidating the data

Table 1. Summary of Current Research on SDN Usage for Energy Efficiency in Cloud Computing

Project	Description	Author	Organization
ElaticTree	Traffic consolidation	Heller et al. [44]	Stanford University, USA
CARPO	Correlation analysis, traffic consolidation	Wang et al. [100]	The Ohio State University, USA
DISCO	Distributed traffic flow consolidation	Zheng et al. [109]	The Ohio State University, USA
FCTcon	Dynamic control of flow completion time in DCN	Zheng and Wang [107]	The Ohio State University, USA
GETB	Energy-aware traffic engineering	Assuncao et al. [26]	University of Lyon, France
VMPlanner	VM Grouping, VM consolidation	Fang et al. [31]	Beijing Jiaotong University, China
VM-Routing	VM placement problem to a routing problem, shortest path	Jin et al. [53]	Florida International University, USA
PowerNetS	Correlation analysis, VM placement, migration	Zheng et al. [108, 110]	The Ohio State University, USA
S-CORE	VM management considering host-network	Cziva et al. [24]	University of Glasgow, UK
QRVE	Energy efficient VM placement and routing	Habibi et al. [43]	Amirkabir University of Technology, Iran
SLAEE-DO	Overbooking for VMs and network flows	Son et al. [93]	University of Melbourne, Australia
ODM-BD	Big data workload slicing in edge-cloud environment	Aujla et al. [4]	Tharpar University, India

flows into a smaller number of links, the unused switches can be switched off to save more energy. The authors evaluated the proposed method for tradeoffs between performance, robustness, and energy savings using real traces collected from production data centers. An empirical prototype was implemented with a latency monitor and feeding various traffic patterns for evaluation.

In CARPO [100], correlation between traffic flows is considered in addition to the traffic flow consolidation. If the traffic flows are less correlated to each other, those flows can be placed on the same network to maximize the overall utilization. By placing less correlated traffic flows together, each link can hold more flows in the limited capacity, which leads to turning more switches off. Wikipedia traces were observed for correlation analysis, and the authors found that different data traffic does not hit the peak at the same time, and the off-peak utilization is less than the peak. Thus, off-peak utilization is used for the consolidation to maximize the energy savings while still minimizing the performance degradation. CARPO also includes link rate adaptation [1] that changes the speed of each port based on the link utilization. The proposed method is implemented on the empirical prototype in small scale and on a simulation environment for performance evaluation with the Wikipedia workload. The results show that the proposed correlation-aware traffic consolidation method can outperform on the web application workload compared to ElasticTree.

Zheng et al. [109] proposed a distributed traffic management framework with the same correlation analysis method. The framework is capable of analyzing the correlation between flows and consolidating the flows based on the scalability, energy savings, and performance. The new approach focused more on the scalability of the correlation-aware flow consolidation and tried to reduce the size of the calculation for analyzing the correlation and the decision making. The authors proposed flow- or switch-based traffic consolidation algorithms that optimize each flow/switch with a limited visibility on the flow or switch, instead of taking the entire DCN flows into account. As the new approach omits the less related flows in its analysis and consolidation, it can be applicable to a larger-scale DCN. The system is empirically implemented and tested on OpenFlow-based switch with 2-pod fat-tree topology, and its performance was evaluated on a simulation set up with the Wikipedia web application workload.

The same authors studied a dynamic strategy to control flow completion time [107]. This study focused on the flow transmission time while achieving energy efficiency. The effort in balancing between energy savings through traffic consolidation and the performance guarantee of the network transmission time brought a new framework to control flows with different requirements and adjusted bandwidth demands accordingly. It was evaluated on the real hardware system and the simulation.

Recently, Assuncao et al. [26] proposed energy-aware traffic engineering methods based on SDN services. The authors designed SDN services to manage switches and control network traffic. The idea is to switch off idle switches after redirecting network traffic into an alternate path that is enough for serving the combined network traffic. A proof-of-concept platform is implemented in a small-scale real system, while the large-scale evaluation was performed in simulation. The authors showed that the proposed architecture could reduce energy consumption by consolidating network traffic into a smaller number of links while maintaining the QoS of the network. Traffic consolidation, however, may incur QoS degradation in case of a network burst when the amount of packets exceeds the capacity of the link. The authors suggested a continuous and frequent monitoring of the network in the SDN controller, e.g., checking the network burst every second, and showed that the system can turn the link on once it detects an overutilized link.

All the aforementioned studies consider intra-DCN architecture, focusing on the energy consumption within a data center in their network-only optimization.

5.1.2 Joint Host-Network Optimization. More recently, researchers proposed joint host-network optimization methods considering both host and network resources simultaneously. SDN can be utilized for VM resource provisioning and optimization. Medina and Garcia explained various migration techniques of VMs in a cloud data center [70]. The authors used three categories for classifying migration methods: process migration, memory migration, and suspend/resume migration. Kapil et al. [54] also presented a detailed survey on the issues of live VM migration. The authors used downtime, migration time, and transferred data size to measure the performance of the live migration techniques and categorized them into postcopy approaches and precopy approaches.

VMPlanner [31] also optimizes VM placement and network routing with three approximation algorithms. The joint VM placement and routing problems are formulated with three NP-complete problems: traffic-aware VM grouping, distance-aware VM-group to server-rack mapping, and power-aware inter-VM traffic flow routing. The authors formulated the problem and proved that minimization of energy consumption considering both VM placement and network routing is a NP-complete problem. The proposed solution is to integrate approximation algorithms to solve these three problems. The solution includes VM grouping that consolidates VMs with high mutual traffic, VM group placement that assigns the VM to the rack of the VM group, and traffic flow consolidation to minimize the interrack traffic.

Jin et al. [53] addressed the joint host-network optimization problem with an integer linear algorithm. The problem is first formulated as an integer linear problem, then converted from a

VM placement problem to a routing problem to combine both problems. To solve the problem, the depth-first search (DFS) is used to determine the best host for VM placement. The proposed algorithm is implemented on the OpenFlow-based prototype with fat-tree topology and evaluated with a number of workloads in the prototype as well as a simulation environment.

More recently, PowerNetS was proposed, which optimizes both VM and network traffic using correlation [108, 110]. Inspired from CARPO [100], the authors extended the network optimization to joint host-network optimization. Correlation of VMs is leveraged for VM consolidation in addition to CARPO's traffic consolidation. The detailed power model is also included to estimate power consumption by a switch chassis, with each port on the switch; idle power of a server; and maximum power of the server. Using Wikipedia and Yahoo traces, correlation coefficients between traffic flows were measured and presented. The authors then designed the PowerNetS framework based on the observed correlation and implemented both the prototype and simulation. The system was evaluated with the same workload traces from the correlation analysis, and the results were compared with non-correlation-based VM placement, the optimal solution, and CARPO results.

Cziva et al. [24] presented an SDN-based VM management platform for live VM migration based on the network-wide communication cost. A prototype was implemented on a real system in a cloud data center testbed. The hierarchical system architecture is presented to support OpenFlow switches and Libvirt hypervisors, and the Ryu SDN controller is used for host and network topology discovery, L2 switching management, statistics gathering , and link weight calculation. They also proposed VM orchestration algorithms that perform VM migration to reduce VM-to-VM communication cost. The algorithm is recurring periodically and hierarchically to reduce the network traffic cost. It is first applied to the core switch layer (the most expensive communication cost) and then gradually applied to the lower layers. The evaluation showed that overall cost converged to the minimum after several rounds of migrations.

Habibi et al. [43] proposed a VM placement and network routing approach for energy efficiency and QoS. The system consists of monitoring and flow detection, QoS-aware routing, network topology storage, and VM placement. The algorithm exploits dynamic flow routing to maintain network QoS and VM placement and migration methods to consolidate VMs for energy efficiency. The authors explained the inevitable tradeoff between energy efficiency and QoS, and proposed a combined approach to address both problems. They measured network throughput, energy consumption, and utilization of the data center on a simulation environment implemented on Mininet tool.

More recently, Son et al. [93] studied a dynamic overbooking strategy of joint compute and network optimization for energy efficiency and SLA compliance. Based on the overbooking strategy that initially allocates fewer resources to prevent overprovisioning, the authors proposed a dynamically adapting overbooking ratio depending on real-time workload for both VM and network flows. The dynamic overbooking ratio for computing resources is periodically calculated from the correlation of CPU utilization of the hosts and VMs. For networking resources, the same method is applied to the network utilization of the hosts, switches, and VMs. The algorithms were evaluated on a simulation platform.

Aujla et al. [4] proposed a workload slicing decision-making scheme for data-intensive applications in a multiedge cloud environment exploiting SDN technology. They consider energy consumption, communication cost, SLA, bandwidth, and revenues in their system model for the optimal decision of data migrations between the central and edge clouds. The SDN-based controller decides flow paths between data centers and edge nodes as well as migration of data among multiple data centers. The proposed scheme is evaluated based on simulation with Google workload traces.

5.2 Performance

Several approaches have been proposed to empower the network performance of cloud computing utilizing SDN technologies. We categorize them into intra-DCN and inter-DCN approaches based on the target architecture in the study. Intra-DCN approaches are meant to improve the performance within a data center including network throughput, latency, and availability on the network transmission. Inter-DCN approaches are mainly for improving the QoS for WANs between tenants and cloud data centers. Many studies [29, 48, 91, 99, 102] rely on the dynamic per-flow routing feature of SDN, which can change the route of network traffic dynamically based on the network condition, e.g., finding an alternate route in case of a network burst to meet QoS requirements. Although these approaches can improve the performance of the network within/between data centers, it may introduce a new challenge incurred from the performance of SDN controllers and switches. Because the centralized SDN controller utilizes a large amount of computing and memory resources to manage all the switches in a network, it is difficult to scale out for a growing network size. We review several studies focusing on improving network performance in the following sections and further discuss the SDN scalability in Section 7.3. Table 2 summarizes reviewed works for performance in this section and the details are presented later.

5.2.1 Intra-DCN Performance Improvement. Wang et al. [99] at Princeton University proposed an OpenFlow-based load balancing technique substituting the expensive dedicated load balancer (LB) server in a data center. A front-end LB is normally used in a data center to redirect client requests to one application server replica for even distribution of workloads across multiple servers. Instead of using an expensive LB server, the authors proposed OpenFlow switches to perform the same load balancing in a data center, by altering the network route for each incoming traffic to be distributed across the servers. The challenge incurred in this approach is how to deal with the excessive size of traffic rules installed in a switch for each flow for different servers. The authors presented an algorithm that finds wildcard rules for the same group of flows directing to the same server, so that it can achieve a target distribution of the traffic with scalability. The proposed method was implemented and evaluated on the NOX OpenFlow controller.

Ishimori et al. [46] proposed QoSFlow, a QoS management system that controls multiple packet schedulers within a Linux kernel to provide a flexible QoS controllability over the integrated SDN network. The traffic schedulers in Linux kernels, such as HTB and SFQ, can become a part of the OpenFlow network in QoSFlow, which allows the SDN controller to manage the QoS using the Linux traffic schedulers. QoSFlow was designed to be an extension to the standard OpenFlow switches to schedule data flows passing through the switch. The proposal was implemented and tested in commodity switches and showed that various Linux schedulers worked on OpenFlow switches. Although the maximum bandwidth has been reduced after applying QoSFlow because of the control overhead of the OpenFlow packets, the approach can provide an easy framework to implement the QoS-aware SDN network in a data center with the classic Linux traffic schedulers.

Wendong et al. [101] proposed another approach to improve network QoS by combining SDN with Autonomic Networking technology, which can be applied in a DCN. The proposed approach applies QoS requirements to SDN automatically by configuring the controller. The autonomic QoS control module is designed to configure queue management methods, packet schedulers, and their parameters on an OpenFlow-based controller, and the flows are managed by finding a matching packet header and changing the forwarding rule accordingly. Also, the authors introduced Packet Context, an information set of the packet characteristics that is carried in the IP header to mark the packet. Using this mark, packets are prioritized in a queue of each switch port to provide a different QoS based on the Packet Context. The proposal requires customizing IP packet headers,

Table 2. Summary of Current Research for Performance Improvement in Cloud Computing with SDN

Project	Description	Author	Organization
OF-SLB	Server load balancing	Wang et al. [99]	Princeton University, USA
QoSFlow	QoS management system for traffic engineering	Ishimori et al. [46]	Federal University of Para, Brazil
AQSDN	Autonomic QoS management system	Wendong et al. [101]	Beijing University of Posts and Telecom., China
SDN-Orch	SDN-based orchestration for host-network resources	Martini et al. [63]	CNIT and Sant'Anna School, Italy
C-N-Orch	Cloud and network orchestration in a data left	Gharbaoui et al. [37]	Sant'Anna School and University of Pisa, Italy
Orch-Opti	SDN-based virtual resource orchestration system for optical DCN	Spadaro et al. [95]	UPC, Spain
OpenQoS	QoS-aware network traffic controller	Egilmez et al. [29]	Koc University, Turkey
B4	SDN-WAN for geographically distributed data lefts	Jain et al. [48]	Google Inc.
CNG	Enhanced networking of distributed VMs	Mechtri et al. [69]	Telecom SudParis, France
ADON	Network management system for scientific workload	Seetharam et al. [91]	University of Missouri, USA
CometCloud	SDN-enabled cloud federation for smart buildings	Petri et al. [80]	Rutgers University, USA
SD-IC	Inter-connection for federated SDN-clouds	Risdianto et al. [88]	GIST, Korea
Orch-IC	Network resource orchestration for inter-clouds	Kim et al. [56]	ETRI, Korea
VIAS	SDN overlay bypass for inter-cloud VPN	Jeong and Figueiredo [52]	University of Florida, USA
CL-Orch	Cross-layer network orchestration signaling framework	Cerroni et al. [18]	University of Bologna and Sant'Anna School, Italy
SVC	SDN-based vehicular cloud for software updates distribution	Azizian et al. [5]	Univeristy of Sherbrooke, Canada
BDT	Optimization model for bulk data transfers	Wu et al. [102]	The University of Hong Kong, Hong Kong
SDN-TE	Fault-tolerant cloud resource management	Amarasinghe et al. [3]	University of Ottawa, Canada

which makes it difficult to implement in a general-purpose network, but it can be realized for intra-DCN traffic with a higher degree of customization possibility.

Researchers also studied improving the VM acceptance rate and network blocking rate of a cloud data center by provisioning both computing and networking resources jointly with SDN technology. Martini et al. [63] presented a management system for both computing and networking resources in virtualized cloud data centers. The system integrates SDN with VM-host management to orchestrate both resources and improve the service acceptance rate and the data center's utilization level while maintaining the quality of user experience. The system consists of resource selection and composition, coordinated configuration and provisioning, and monitoring and registration functions. The authors proposed a network traffic estimation combining instantaneous and historical values to predict the network load. Also, the system uses a server-driven or a network-driven resource allocation algorithm that attempts to find a server or a network first for a VM request. Both simulation and empirical setups are used for evaluating the proposed system and algorithms.

An SDN-based orchestration system for cloud data centers is proposed by Gharbaoui et al. [37]. to control both computing and network resources. The system architecture includes an orchestration layer on top of the SDN controller and a VM manager that perform resource selection, provisioning, and monitoring. The authors also propose a joint resource selection algorithm for both computing and networking resources. The algorithm selects a physical machine to allocate a VM first and estimates network traffic of the edge switch. If the estimated traffic is overloaded, another physical machine is tried until the available physical machine is found. The proposed system and algorithm are evaluated extensively in a simulation environment by measuring the VM request rejection ratio and link utilization overloading ratio.

Spadaro et al. [95] presented an SDN-based VM and virtual network orchestration system for optical DCN. The orchestrator module is integrated in OpenStack and manages an OpenDaylight SDN controller via north-bound API of the controller. An optimization algorithm for provisioning virtual data centers (i.e., VMs and virtual networks) is also proposed to increase the request acceptance ratio by jointly mapping VMs and virtual links. The authors consider joint optimization of cloud and network resources for the intra-DCN domain and evaluate the performance by measuring acceptance demands and blocking probability.

5.2.2 Inter-DCN Traffic Engineering. OpenQoS is an OpenFlow-compatible SDN controller designed to support the end-to-end QoS for multimedia traffic between end-users and streaming servers [29]. The network flows for the multimedia traffic are dynamically rerouted to meet the QoS requirement. The incoming traffic flows are grouped into two categories: multimedia flows and the other data flows, where multimedia flows have specific QoS requirements. The multimedia streaming flows are placed on a dedicated route differentiated from the other flows, which are transferred through the typical shortest path. The proposal exploits SDN's flow separation and dynamic routing features, which differ from the traditional QoS approaches utilizing a resource reservation or priority routing approach. In OpenQoS, multiple flows are bounded to a group by filtering packet header fields. However, due to the processing cost for packet filtering in switches, it should be wisely defined and aggregated. The system was implemented and tested on a small-scale test bed. The fundamental idea of OpenQoS can be extended to the intra-DCN and inter-DCN traffic prioritization in clouds.

B4 is a private WAN connecting geographically distributed data centers implemented by Google [48]. Google adopted SDN concepts in B4 to separate the control plane from the data forwarding plane to enable the network to be programmable. B4 is designed to fulfill the unique characteristics of Google's inter-DCN traffic, such as its massive bandwidth requirements, elastic traffic demand, and full control of the edge servers and network. SDN principles and OpenFlow are implemented to support standard routing protocols and traffic engineering. It can leverage competing demands at traffic congestion, leverage the network capacity using multipath forwarding, and reallocate bandwidth dynamically when the link failure or application demand changes. In their evaluation, the network utilization of many links reached to about 100%, with all links running on average at 70% of utilization. Compared to the traditional network where the average utilization is typically 30% to 40% due to overprovisioning, B4 resulted in 2 to 3 times efficiency improvements with the same fault tolerance.

Mechtri et al. [69] studied SDN for intercloud networking. The authors presented a general-purpose SDN controller for an intercloud networking gateway that can be configured and managed by authorized customers. It enables multiple VMs in geographically distributed data centers to interconnect through the SDN-enabled gateway and customize the network for allocation and configuration based on the network requirement of the VMs. The cloud broker in the middle is introduced in order to manage the VMs in distributed data centers and the network infrastructure connecting the VMs. It controls switches in each data center involving the interconnection to enable the network configuration to fulfill the requirement. The proposed algorithm is evaluated in simulation.

ADON is an SDN-based network management system for the hybrid campus cloud architecture to run data-intensive science applications without network bottleneck [91]. As these science applications running in the private cloud of the university sometimes need excessive resources from public clouds, some workloads should be delivered to the external public cloud. In such cases, network bottlenecks can happen while competing for remote resources by multiple applications. The authors deployed an SDN controller to prioritize the requests based on the QoS requirements and observed the characteristics of each scientific application. Using the per-flow routing and bandwidth allocation feature of OpenFlow, application flows are assigned to a different route with prioritized bandwidth by the application type. The testbed is implemented in practice in two campus locations in addition to the Mininet emulation study.

Petri et al. [80] proposed SDN integration for intercloud federations to compute smart building sensor data. The authors focused on a data-intensive application in need of significant data processing in real time. The proposed architecture utilized the SDN concept over the federated clouds, which connects multiple clouds to provide resource elasticity and network programmability. The proposed architecture was applied to a smart building scenario aimed at improving processing times and costs for building power usage optimization. The prototype has been empirically implemented and deployed on three sites in the United Kingdom and the United States, and validated with workloads obtained from real sensors. The evaluation showed that in-transit processing using SDN capability reduced overall task execution time for time-constrained workloads.

Risdianto et al. [88] discussed the SDN usage to interconnect federated cloud data centers. The approach combined the usage of both L2 tunnel-based overlay virtual networking and L3 BGP-leveraged routing exchange for connecting multiple data centers. With the usage of SDN, the proposed system can flexibly configure the selected sites and their forwarding path for better redundancy and load balancing. Leveraging the two techniques enabled the flexible selection between L2 and L3 for interconnection of data centers. The ONOS SDN controller [76] is used to deploy L3 routing exchange, and OpenDaylight [77] is for L2 management in the implemented prototype.

Kim et al. [56] presented SDN orchestration architecture for integrating SDN-enabled clouds and transport networks. The paper investigates the coordination methodology for inter-DCN transmission utilizing SDN functionality to manage both cloud and intercloud networks. Intra-DCN and inter-DCN are managed under separate control domains, named Cloud SDN (C-SDN) and Transport SDN (T-SDN), respectively. On top of them, the virtual network coordinator and transport network coordinator oversee orchestrating the transport network between data centers, such as creating a global virtual network for VMs in different data centers. The proposed system is implemented on their testbed using OpenDaylight, ONOS, and OpenStack.

Jeong and Figueiredo [52] proposed an SDN-enabled intercloud virtual private networking technique to increase the flexibility of the overlay flows and alleviate the network overhead caused by the tunneling protocols. SDN is integrated into the overlay controllers, which creates the virtualized network for inter-DCN and selectively bypasses the tunneling packets. The approach is applied to containers running on VMs where the virtual private network is configured among multiple VMs located in different providers and data centers. The prototype is implemented on a Ryu controller that controls the OpenVSwitches running on the VMs and evaluates the throughput among the containers running within and across cloud providers.

Cerroni et al. [18] proposed a signaling framework architecture following the SDN principle for multidomain data transport network orchestration to provide different QoS requirements of applications. The application-driven configuration of network resources is handled by the Application-Oriented Module (AO-M), which implements the network resource description parser, SIP proxy server, and network module that controls the underlying network service plane (i.e., SDN controller layer). The framework is implemented and validated on an empirical testbed in multiple locations with various commercial equipment.

More recently, Azizian et al. [5] proposed using SDN and cloud computing to distribute software updates of vehicles. The SDN controller controls both the inter-DCN flows of multiple data centers and the base station networking device where the update information is sent through from the cloud data center to the vehicle.

Bulk data transfers across geographically distributed cloud data centers have been studied by Wu et al. [102]. Bulk volumes of data are transferred between data centers for VM migrations, replication of large contents, and aggregation of MapReduce frames. In this work, the bulk transfer problem has been formulated to an optimal chunk routing problem to maximize the aggregated utility gain before the deadlines [102]. Three dynamic algorithms are proposed that exploit bandwidth allocation, route adjustment, and low-complexity heuristics amended from the first algorithm. Time-constrained bulk transmissions are considered in the problem formulation. The proposed system changes the flow table in a core switch connected to a gateway in data centers to alter the bandwidth reservation or routing schedules. The authors evaluated the proposed algorithms on 10 emulated data centers on a real system and measured the job acceptance rate, the total number of accepted jobs on different urgencies, and the scheduling delay.

For the purpose of network availability, Amarasinghe et al. [3] proposed a fault-tolerant resource management scheme for clouds using SDN. The authors presented reactive traffic engineering techniques for network failure restoration using network monitoring and dynamic routing features of SDN. The system identifies unexpected link failure and recovers the network by reconfiguring the forwarding switches. The prototype was implemented on a POX controller and evaluated on the Mininet emulation environment.

5.3 Virtualization

SDN plays a key role for network virtualization and NFV in cloud computing. Network virtualization is to segment the physical network resources in cloud data centers into smaller segmentations and lease it to cloud tenants, like leasing VMs in clouds enabled by host virtualization. NFV utilizes a generic computing resource, such as VMs, for providing specific network functions that require high computing power. Instead of purchasing expensive dedicated hardware for CPU-intensive network functions such as firewall, NAT, or load balancing, NFV can provide a cheaper alternative that utilizes generic hardware with virtualization technology.

Table 3. Summary of Current Research for Virtualization in Cloud Computing with the Usage of SDN

Project	Description	Author	Organization
FairCloud	Tradeoffs sharing networks in cloud computing	Popa et al. [82]	UC Berkeley, USA
QVIA-SDN	Virtual infrastructure allocation in SDN-clouds	Souza et al. [27]	Santa Catarina State University, Brazil
Opti-VNF	Optimal VNF allocation in an SDN-cloud	Leivadeas et al. [61]	Carleton University, Canada
Dyn-NFV	Dynamic NFV deployment with SDN	Callegati et al. [16]	University of Bologna, Italy
E2E-SO	End-to-end NFV orchestration for edge and cloud data centers	Bonafiglia et al. [12]	Politecnico di Torino, Italy

While SDN intends a clear separation of the network control plane from the forwarding plane to enable the programmability of networks, NFV shifts the paradigm of the network function deployment through advanced virtualization technologies [16]. In the concept of NFV, network functions are provisioned in virtualized resources instead of being tightly coupled in the dedicated hardware, which enables one to provision and migrate network functions across the infrastructure elastically and dynamically. Although NFV can be realized without the aid of SDN, the integration of SDN with NFV can accelerate the NFV deployment process by offering a scalable and flexible underlying network architecture [75].

A summary of reviewed works for the virtualization objective is presented in Table 3, and the details of each work are explained later.

FairCloud was proposed to virtualize the network in cloud data centers similar to using VMs for computing power virtualization [82]. The authors referred the challenges of sharing the network in cloud computing into three aspects: minimum bandwidth guarantee, achieving higher utilization, and network proportionality. Network proportionality was described as the fair share of the network resources among cloud tenants where every tenant has the same proportion of the network. According to the authors, the fundamental tradeoffs are necessary between three aspects. For example, if we aim to guarantee minimum bandwidth, the network proportionality cannot be achieved, and vice versa. A similar tradeoff is necessary between network proportionality and high utilization. For network proportionality, network bandwidth should be evenly shared by cloud customers if they use the same type of VMs and network plans, even if their actual bandwidth usages are different. Thus, strict network proportionality lowers the overall bandwidth utilization of the data center if the disparity of network usage exists between customers. In consideration of these tradeoffs, three network-sharing policies are proposed and evaluated in simulation.

Souza et al. [27] studied a QoS-aware virtual infrastructure (VMs and their network connections) allocation problem on SDN-clouds as a mixed-integer program. The authors formulate the online virtual infrastructure allocation in SDN-enabled cloud data centers. In order to solve the mixed integer problem, the authors used a relaxed linear program, rounding techniques, and heuristic approaches. They introduced a new VM selection method that considers not only a geographical location of the available zone but also the end-to-end latency requirement of the VMs. The formula also includes the constraints of server and link capacity, forwarding table size, and latency. The evaluation was performed under a simulation environment by measuring five metrics: revenue-cost ratio, data center fragmentation, a runtime for allocation, acceptance ratio, and mean latency of the allocated virtual infrastructure.

In NFV, where networking middleboxes (e.g., NAT, firewalls, intrusion detection) are turned into software-based virtual nodes, virtualized network functions (VNFs) are decoupled from dedicated hardware and can be run on any generic machines, similar to running VMs on a physical machine. The survey by Mijumbi et al. [71] focused on Network Function Virtualization studies and its relationship with SDN. NFV architectures and business models were provided in addition to a detailed explanation of the relationship with cloud computing and SDN. The main survey covered the standardization effort on NFV and the major collaborative projects in both industry and academia. Esposito et al. [30] presented a survey on the slice embedding problem on network virtualization. The authors defined the slice embedding problem as a subproblem of resource allocation that was composed of three steps: resource discovery, virtual network mapping, and allocation. For each step, the surveyed literature was characterized by constraint type, type of dynamics, and resource allocation method.

As VNFs can be placed at any hardware, the VNF allocation problem has received increasing attention with the emergence of NFV technology. Recently, Leivadeas et al. [61] presented an optimal VNF allocation method for an SDN-enabled cloud. The authors considered single or multiple services provided to a single tenant or multiple tenants in the model. The NFV orchestrator controls both the SDN controller and the cloud controller to select the optimized place to allocate the VNFs. The formulated problem includes both servers and switches in a cloud to minimize the operational cost of the cloud provider. The optimal solution is presented by mixed integer programming, and four heuristics are proposed. The proposed algorithms are evaluated on simulation to measure the operational cost of the cloud provider, the number of utilized nodes and links, and the utilization. The paper showed the optimal solution of the VNF allocation problem and proposed simple and basic heuristics. More delicate heuristics can be studied and proposed to complement their study for further cost savings or energy efficiency.

Callegati et al. [16] presented a proof-of-concept demonstration of dynamic NFV deployment in the cloud environment. The system is capable of dynamic SDN control integrated with cloud management for telecommunication operators and service providers implementing NFVs enabling orchestration of intra-DCN and inter-DCN. The authors consider single or multiple VMs hosting various VNFs that dynamically adapt to the network condition. The proof of concept is implemented in an Ericsson Cloud Lab environment running Ericsson Cloud Manager on top of OpenStack.

Bonafiglia et al. [12] also presented an open-source framework that manages NFV deployment on edge and cloud data centers along with interdomain networks that connect data centers. This work considers both intra- and inter-DCN architecture as well as orchestration of cloud and network resources in a data center. For edge and cloud data centers, OpenStack is used to control the resources, whereas thw OpenDaylight or ONOS controller manages inter-DC SDN networks. On top of these heterogeneous domain controllers, Overarching Orchestrator (OO) oversees all domains to provide end-to-end service orchestration. OO interacts with edge/cloud and network domains through OpenStack and SDN Domain Orchestrator, respectively, which handles a specific domain to communicate with the underlying infrastructure controller. The system also supports network function deployment in any domain, i.e., either cloud or SDN domains. The system is validated with OpenStack, ONOS, and Mininet setup by experimenting with the NAT function deployment on either the cloud or SDN network.

5.4 Security

Many studies have been proposed for enhancing security utilizing SDN features to detect and prevent DDoS attacks [104], and some researchers also explained the security vulnerability of the SDN controller itself. However, it is difficult to find much literature specifically targeting the cloud computing environment. Although the general approaches using SDN for security can be applied

Table 4. Summary of Current Research for Security in Cloud Computing with the Usage of SDN

Project	Description	Author	Organization
GBSF	Game-based attack analysis and countermeasure selection	Chowdhary et al. [20]	Arizona State University, USA
Brew	Security framework to check flow rule conflicts in SDN	Pisharody et al. [81]	Arizona State University, USA

to cloud computing, we will exclude those general approaches in this survey as our intention is solely on cloud computing. Table 4 presents the list of surveyed studies for security.

Yan et al. [104] presented a comprehensive survey on how to prevent DDoS attack using SDN features. The capabilities of SDN can make DDoS detection and reaction easier, while the SDN platform itself is vulnerable to security attacks noted in its centralized architecture. The authors discussed both sides: the detailed characteristic of the DDoS attack in cloud computing and defense mechanism using SDN, and DDoS attacks launching on SDN and prevention approaches.

A security framework in the SDN-cloud environment was proposed by Chowdhary et al. [20] to prevent DDoS attacks using dynamic game theory. The framework is based on reward and punishment in the usage of network bandwidth so that the attackers' bandwidth will be downgraded dynamically for a certain period. The framework was implemented on top of an ODL SDN controller that functions through the north-bound API of the controller, and evaluated with Mininet.

Recently, Pisharody et al. [81] from the same institution proposed a security policy analysis framework to check the vulnerability of flow rules in SDN-based cloud environments. They describe possible conflicts among flow rules in SDN's forwarding table that can cause information leakage. The framework detects flow rule conflicts in multiple SDN controllers. The detection mechanism is extended from the firewall rule conflict detection methods in traditional networks. The system is implemented on an OpenDaylight SDN controller and tested in empirical systems.

5.5 Summary and Comparison

All studies covered in the survey are summarized and compared in Table 5 based on our taxonomy in Figure 4. Researchers are actively studying for energy efficiency and performance optimization using SDN in clouds. The scope is varied depending on the proposed method, some focusing on the network-only method and others considering joint computing and networking resource optimization. For studies on energy efficiency, all surveyed papers consider intra-DCN architecture in their model, which focuses on power savings within a data center. This reflects the energy trend in recent years that an enormous amount of electricity is consumed by data centers, which has been increasing rapidly [72]. On the other hand, for performance improvement, many studies consider inter-DCN architecture exploiting SDN on WAN to enhance the QoS and network bandwidth. Although some studies consider the network performance within a data center, there are more opportunities to exploit SDN technology in a WAN environment where limited network resources have to be provisioned for cloud tenants. Using SDN's dynamic configuration and network optimization, cloud tenants can acquire more availability and reliability in intra-DCN, resulting in better QoS.

For the application model, many studies have no target application explicitly considered in the proposal, which has no tick symbol in the table. These studies propose generic approaches so that any applications running on the cloud can be beneficial from the proposed method. For energy efficiency, a number of studies consider the web application model, reflecting the popularity of web application hosting on clouds. Also, it is accessible to acquire web application workloads because

Table 5. Characteristics of SDN Usage in Cloud Computing

Project	**Objective**	**Scope**	**Arch**		**App**			**Rsrc**		**Eval**	
			intra	inter	web	str	bat	hom	het	sim	emp
ElaticTree [44]	Energy	Network	✓					✓			✓
CARPO [100]	Energy	Network	✓		✓			✓			✓
DISCO [109]	Energy	Network	✓		✓			✓		✓	✓
FCTcon [107]	Energy	Network	✓		✓			✓		✓	
GETB [26]	Energy	Network	✓					✓		✓	✓
VMPlanner [31]	Energy	Joint	✓					✓		✓	
VM-Routing [53]	Energy	Joint	✓					✓		✓	✓
PowerNetS [108]	Energy	Joint	✓		✓			✓			✓
S-CORE [24]	Energy	Joint	✓					✓			✓
QRVE [43]	Energy	Joint	✓					✓		✓	
SLAEE-DO [93]	Energy	Joint	✓		✓			✓		✓	
ODM-BD [4]	Energy	Joint		✓					✓	✓	
OF-SLB [99]	Performance	Network	✓					✓		✓	
QoSFlow [46]	Performance	Network	✓				✓	✓			✓
AQSDN [101]	Performance	Network	✓			✓		✓			✓
SDN-Orch [63]	Performance	Joint	✓					✓		✓	✓
C-N-Orch [37]	Performance	Joint	✓					✓		✓	
Orch-Opti [95]	Performance	Joint	✓						✓		✓
OpenQoS [29]	Performance	Network		✓		✓		✓			✓
B4 [48]	Performance	Network		✓					✓		✓
CNG [69]	Performance	Joint		✓					✓	✓	
ADON [91]	Performance	Network		✓			✓		✓		✓
CometCloud [80]	Performance	Network		✓		✓			✓		✓
SD-IC [88]	Performance	Network		✓					✓	✓	✓
Orch-IC [56]	Performance	Network		✓					✓		✓
VIAS [52]	Performance	Joint		✓	✓				✓		✓
CL-Orch [18]	Performance	Network		✓					✓		✓
SVC [5]	Performance	Joint		✓	✓				✓	✓	
BDT [102]	Performance	Network		✓				✓			✓
SDN-TE [3]	Performance	Network		✓				✓		✓	
FairCloud [82]	Virtualization	Network	✓					✓		✓	✓
QVIA-SDN [27]	Virtualization	Joint	✓					✓		✓	
Opti-VNF [61]	Virtualization	Joint	✓					✓		✓	
Dyn-NFV [16]	Virtualization	Joint	✓	✓					✓		✓
E2E-SO [12]	Virtualization	Joint	✓	✓					✓		✓
GBSF [20]	Security	Network	✓						✓	✓	
Brew [81]	Security	Network	✓						✓		✓

Arch: Target architecture - intra (Intra-DCN) or inter (Inter-DCN); **App**: Application model - web (web application), str (streaming), or bat (batch processing); **Rsrc**: Resource configuration - hom (homogeneous) or het (heterogeneous); **Eval**: Evaluation method - sim (simulation) or emp (empirical).

many datasets are publicly available online, including Wikipedia[1] and Yahoo![2] traces. On resource configuration, most studies for energy efficiency consider homogeneous resources to simplify the research problem because consideration of heterogeneous resource types leads to adding extra parameters to the problem formula. There are a number of studies using both simulation and the empirical method for evaluation, while most studies choose either one of them. Details of available evaluation methods are explained in the following section.

6 EVALUATION METHODS AND TECHNOLOGIES

For accelerating innovation and development of SDN-enabled cloud computing, tools and toolkits are required to build a testbed for testing OpenFlow and SDN systems in a cloud data center. The testbed also has the capability to measure the energy consumption to evaluate proposed algorithms. In this section, simulation tools and empirical methods are explained.

6.1 Simulation Platforms and Emulators

The simulation platform provides a reproducible and controlled environment for evaluation with ease of configuration and alteration. For cloud computing, many simulation tools have been introduced to evaluate new approaches to managing and controlling the cloud data center and various scenarios. CloudSim [15] is a popular cloud simulator implemented in Java, providing a discrete event-based simulation environment capable of simulating cloud data centers, hosts, VMs, and brokers. Various scenarios can be implemented in CloudSim, including the VM placement policy, VM migration policy, brokering policy, and other data center management policy. It also supports workload simulation executing in the VMs. With its easy-to-use discrete event-based architecture, additional elements can be added to send and receive simulation events, as well as extending the existing entities to provide extra functionality. However, CloudSim does not support network events in details.

In order to fulfill the lack of network simulation capability of CloudSim, NetworkCloudSim [36] is introduced to simulate applications with network communication tasks. Additional network elements are added in NetworkCloudSim, including network switches and links that receive network events and calculate estimated network transmission time. Although NetworkCloudSim includes extensive network functionality to simulate data center network and message-passing applications in a data center, the support of SDN is not considered in the design and implementation.

For SDN support in CloudSim, CloudSimSDN [94] was recently introduced to enable simulating SDN features such as adjustable bandwidth allocation or dynamic network configuration. CloudSimSDN is aimed to simulate all features of SDN that can be deployed in cloud data centers including intra-DCN, inter-DCN, and NFV support. Using CloudSimSDN, an SDN controller can be implemented within the simulator to mimic the behavior of the controller to alleviate the complexity of emulating the OpenFlow controller in the simulation. Although it provides only simple results in networking simulation, the simplified architecture provides essential results with the support of SDN functionalities in a short running time.

GreenCloud [57] is an NS-2 [73]-based simulation framework that captures the energy aspect of cloud data centers including computing and communication elements. With the integration of NS-2, which can capture the network pattern accurately on the packet level, GreenCloud can also provide accurate network results. The simulation entities include hosts and switches with power consumption models such as DVFS. Workload models are also predefined and provided in the framework for three types of jobs, e.g., compute intensive, data intensive, and balanced. Although

[1] www.wikibench.eu.
[2] webscope.sandbox.yahoo.com/catalog.php?datatype=s.

GreenCloud provides a comprehensive simulation environment to evaluate network aspects in clouds, evaluating SDN-based applications on GreenCloud is not straightforward because NS-2 and accordingly GreenCloud have not specifically considered SDN features in their design.

For SDN emulation, Mininet [60] is a popular emulation tool to enable testing SDN controllers. Mininet uses virtualization techniques provided by the Linux kernel, which is capable of emulating hundreds of nodes with arbitrary network topologies. As it uses a real kernel of Linux, it can produce more accurate results including delays and congestion generated at the operating system level. Any OpenFlow controller can be tested in Mininet with a capability of executing Linux programs virtually in the emulated host in Mininet. NS-3 [87] is another discrete-event network simulator that provides a simulation for various network protocols on wired and wireless networks. Although Mininet and NS-3 are reliable network emulation and simulation tools, they are not suitable for testing cloud-specific features such as workload schedulers or VM placement policies.

Teixeira et al. [96] proposed a combination framework of Mininet with POX [83], a Python controller for OpenFlow, in order to support simulation of SDN features in cloud computing environments. Mininet is used to emulate network topologies and data traffic in a data center running an OpenFlow controller in POX. With the usage of Mininet and POX, it provides practical results and ready-to-use software in a real SDN environment. The simulation tool, however, is lacking support for cloud-specific features such as defining heterogeneous VM types or executing various application workloads at the simulated host.

6.2 Empirical Platforms

OpenStackEmu [10] is a testbed framework combining network emulation with OpenStack [78] and SDN. The authors combined the SDN controller of OpenStack with another network emulator to enable emulating a large-scale network connected to the OpenStack infrastructure. It also included a data center traffic generator in the framework. Different VM migration, load balancing, and routing strategies can be evaluated on real VMs connected through the emulated network topology.

OpenDaylight (ODL) [77] and ONOS [76] are open-source SDN controllers that support the SDN integration for OpenStack via plug-ins. Neutron, the networking module in the OpenStack suite, can be configured to use an alternative SDN controller instead of Neutron's own functions. For instance, ODL implements a specific feature called NetVirt (Network Virtualization) for OpenStack integration. By enabling the NetVirt feature in ODL and configuring Neutron to use ODL as a default SDN controller, an OpenStack-enabled private cloud can be used as a testbed for evaluating SDN features in cloud computing.

7 GAP ANALYSIS AND FUTURE DIRECTIONS

Although many studies have investigated the use of SDN technology in cloud computing, there are still several aspects that have not been explored comprehensively.

7.1 Energy-Efficient Cloud Computing

Cloud data centers consume an enormous amount of electricity that has been increasing rapidly. The annual amount of energy consumption of US data centers was estimated at 91 billion kilowatt-hours in 2013, which is enough to power all households in New York City for two years. It is even expected to increase to approximately 140 billion kilowatt-hours in 2020, which will cost $13 billion annually [72]. Thus, improving energy efficiency in cloud computing has become an utmost research problem in academia and industries. Although many researchers have already contributed to reducing the energy consumption in cloud computing, most of them try to optimize computing resources or network resources separately.

Joint optimization of computing and networking resources for reducing power consumption is one of the less explored aspects. Joint optimization considering both computing and network resources simultaneously is rare due to the lack of supportive technology that can control both at the same time. With the recent evolution of SDN, joint optimization is feasible in technology and can be investigated further to save more energy and provide better QoS. Service-Level Agreement (SLA) must also be fulfilled in the power optimization to guarantee profit maximization for cloud providers.

7.2 Supporting Big Data Applications

Cloud computing becomes a fundamental infrastructure for Big Data applications with its massive resource requirements for computation jobs and network transmission. Utilizing SDN-enabled cloud infrastructure can bring vital benefits to Big Data applications to reduce network bottlenecks during application processing. For example, aggregation of the mapped data from mappers to the reducer can cause a network burst in a short time when the mapping processes are completed at a similar time. SDN can be exploited for network-aware scheduling of Big Data workloads in a map-reduce platform such as Hadoop. Network bandwidth is considered using SDN for Hadoop job scheduling to guarantee data locality and optimize task assignment among available workers [86]. In addition to the computation processing time of the workload, network limitation between workers can also be considered when scheduling and distributing jobs.

Researchers also investigate SDN usage for large data transfers in distributed file access systems and for content delivery cache networks. SDN can be exploited to collect network information and find an optimal path between the closest data source and the consumer in a distributed file system [58]. For a content delivery network, distributed cache nodes can be considered as network switches in the SDN paradigm that are controlled by a cache network controller that optimizes data transfers between cache nodes by updating the popularity statistics of contents measured at each cache node [23]. As few researchers have studied how to utilize SDN in cloud data centers for Big Data applications, more studies are necessary in this regard.

7.3 Scalability of SDN

Scalability of the SDN controller is a critical concern for SDN deployment due to its centralized logic of the architecture and the separation of the control plane from the forwarding plane. As the controller gathers all network information and manages every switch in the entire network, it is challenging to make the controller scalable and prevent the single point of failure at the controller. In clouds, this pitfall can be a significant issue due to the size and complexity of DCN and the expected SLA for the provider. The SDN controller for the cloud DCN can easily become a single point of failure as controllers in the DCN can be overloaded due to the tremendous number of switches. The problem can be tackled by distributing controllers or proactively installing the flow rules on switches [105].

A single controller design is simple and easy for maintenance with a global network view, but it is considered less scalable compared to other topological architecture of SDN controllers [55]. SDN controllers can be distributed with a flat or hierarchical architecture via the west-east-bound APIs. The west-east-bound APIs can be utilized to communicate between the controllers so that the controllers can be scaled by exchanging network information and policies through the APIs. Empowering the performance of controller machines is also exploited to increase SDN scalability, which leads the controllers to handle more packet flows and reduce overhead. Multicore parallel processing and routing optimization are proposed to improve I/O performance and reduce memory overhead [55].

7.4 Network Virtualization and NFV

There are also gaps in realizing network virtualization in clouds, which can be more feasible with the integration of SDN. Although the traditional networking technologies such as VLAN and VPN have been widely adopted in industry, integrating SDN with the traditional approaches can bring even better performance and more flexibility to clouds. As DCN has a complex topology with a large number of devices connected through various switches, network virtualization in traditional DCN was difficult to achieve due to the complexity of network configuration and the lack of adaptability. With SDN, flows between VMs can be fine-controlled more flexibly and dynamically, adapting to the changing traffic. Switches can be dynamically configured in real time based on the decision of the controller, which can oversee the entire network. Thus, network virtualization for network SLA guarantee is possibly investigated and introduced in practice with more extensive research in SDN-enabled cloud data centers.

Recently, NFV has acquired significant attention in both cloud computing and networking research communities. By virtualizing network functions that used to be provided by expensive and tightly coupled hardware, providers can reduce the cost of purchasing and upgrading the hardware as the network functions can run on generic hardware like running VM on generic hosts. In networking aspects, how to virtualize those network functions is one of the utmost research topics, whereas resource allocation and provisioning for NFV have been studied in cloud computing aspects. Both aspects should be further explored in order to enhance and implement NFV.

7.5 Enhancing Security in Clouds and SDN

For public cloud providers, the security of the data center and their data is a crucial factor of the business that must be fulfilled. While security in cloud data centers has been explored extensively with diverse approaches, security of SDN usage, in particular, still needs to be investigated to provide insurance for providers applying SDN in their data centers. Current security issues include encrypting SDN control packets between a switch and the controller, protecting the SDN controller from an abnormal access, and protecting packets passing in the virtualized network. Also, researchers put more effort into investigating the vulnerability of SDN itself from DDoS attacks [49] and the security issue of the stateful SDN data planes [25]. As a massive number of different tenants share the same network medium in the cloud, it is crucial to ensure that the virtualized network is secure and isolated from other tenants.

7.6 Realization of Software-Defined Clouds and Edge Computing

Although the fundamental architecture of autonomous SDCs is proposed in several works [13, 50], the implementation of SDCs is still in the early stage due to the lack of virtualization technologies for cloud resources and the integration of these techniques in clouds. Leasing computing and storage resources have already been implemented in commercial cloud platforms thanks to the recent enhancement in virtualization technologies, but further investigation in network slicing and dynamic management is needed to support fully automated SDCs. In order to support network slicing resource management, SDN is a core technology that can bring autonomic network function and application deployment in clouds.

Furthermore, SDN can function as a key networking technology for edge computing where the current cloud computing concept is extended to the edge of the network. In edge computing, time-critical and recurring workloads can be processed at the edge nodes close to end-users without transferring to the central cloud infrastructures so that the application responsiveness and the networking efficiency can be improved [35]. It is heavily driven by emerging IoT applications where the low-profile devices equipped with sensors collectively generate a massive amount of data continuously. With the emergence of edge computing, the large volume of data can be processed at the edge of the network without transmission to the central cloud infrastructure. An optimal decision has to be made in the edge-cloud environment whether migrating data to edge nodes or hosting in the central cloud [4]. Recently, researchers also have started exploring virtualized network function (VNF) placement for edge computing to enable autoscaling and placing the VNFs across the edges and the clouds. SDN can be utilized for WAN optimization to manage the traffic between the edge and cloud on top of network slicing and VNF placement [45].

8 SUMMARY AND CONCLUSIONS

This article presented a taxonomy of SDN-enabled cloud computing and the survey of the state of the art in building SDN-based cloud computing environments. We categorized necessary aspects of existing works in how to make SDN-enabled cloud computing focusing on networking aspects with SDN technology. The elements include architecture in the usage of SDN, the objective of the research, the application model, hardware configuration, and the evaluation method to test the proposed approaches. Each element in the taxonomy was explained in detail, and the corresponding papers were presented accordingly. We also described various research projects conducted for energy-efficient cloud data centers. There are three main approaches to reduce the energy consumption in data centers: host optimization, network optimization, and joint optimization. Recently, many works have been focusing on joint optimization that considers the host and network simultaneously to decrease power usage and save operational cost. Afterward, network QoS management methods based on SDN were explained, following various research tools for simulation and energy modeling. Some tools focus on the network with the SDN controller while others focus on hosts in the data center.

SDN has brought many opportunities in networking that have enabled dynamic adaptation and reconfiguration with its separation of the control plane from the forwarding device controlled by the centralized controller. The SDN controller manages the network flows dynamically and individually with the global view of the entire network. The emergence of SDN with dynamic controllability can bring various benefits to cloud data centers where in essence the requirements and utilization change dynamically on demand. Although many studies have been presented in utilizing SDN with various objectives, more research is necessary to fill the gap between current studies and the ultimate autonomous and optimized cloud data center.

ACKNOWLEDGMENTS

We thank editors (Prof. Sartaj Sahni and Prof. Antonio Corradi) and anonymous reviewers for their insightful comments and helpful suggestions on improving this manuscript. We also thank Dr. Adel Nadjaran Toosi for providing valuable feedback.

REFERENCES

[1] Dennis Abts, Michael R. Marty, Philip M. Wells, Peter Klausler, and Hong Liu. 2010. Energy proportional datacenter networks. In *Proceedings of the 37th Annual International Symposium on Computer Architecture (ISCA'10)*. ACM, New York, 338–347. DOI: http://dx.doi.org/10.1145/1815961.1816004

[2] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. 2008. A scalable, commodity data center network architecture. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication (SIGCOMM'08)*. ACM, New York, 63–74. DOI: http://dx.doi.org/10.1145/1402958.1402967

[3] Heli Amarasinghe, Abdallah Jarray, and Ahmed Karmouch. 2017. Fault-tolerant IaaS management for networked cloud infrastructure with SDN. In *Proceedings of the 2017 IEEE International Conference on Communications (ICC'17)*. 1–7. DOI: http://dx.doi.org/10.1109/ICC.2017.7996342

[4] Gagangeet Singh Aujla, Neeraj Kumar, Albert Y. Zomaya, and Rajiv Rajan. 2018. Optimal decision making for big data processing at edge-cloud environment: An SDN perspective. *IEEE Transactions on Industrial Informatics* 14, 2 (2018), 778–789. DOI: http://dx.doi.org/10.1109/TII.2017.2738841

[5] Meysam Azizian, Soumaya Cherkaoui, and Abdelhakim Senhaji Hafd. 2017. Vehicle software updates distribution with SDN and cloud computing. *IEEE Communications Magazine* 55, 8 (2017), 74–79. DOI: http://dx.doi.org/10.1109/MCOM.2017.1601161

[6] Siamak Azodolmolky, Philipp Wieder, and Ramin Yahyapour. 2013. Cloud computing networking: Challenges and opportunities for innovations. *IEEE Communications Magazine* 51, 7 (July 2013), 54–62. DOI: http://dx.doi.org/10.1109/MCOM.2013.6553678

[7] Mohammad Banikazemi, David Olshefski, Anees Shaikh, John Tracey, and Guohui Wang. 2013. Meridian: An SDN platform for cloud network services. *IEEE Communications Magazine* 51, 2 (February 2013), 120–127. DOI: http://dx.doi.org/10.1109/MCOM.2013.6461196

[8] Md Faizul Bari, Raouf Boutaba, Rafael Esteves, Lisandro Zambenedetti Granville, Maxim Podlesny, Md Golam Rabbani, Qi Zhang, and Mohamed Faten Zhani. 2013. Data center network virtualization: A survey. *IEEE Communications Surveys Tutorials* 15, 2 (2013), 909–928. DOI: http://dx.doi.org/10.1109/SURV.2012.090512.00043

[9] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems* 28, 5 (May 2012), 755–768. DOI: http://dx.doi.org/10.1016/j.future.2011.04.017

[10] Cristian Hernandez Benet, Robayet Nasim, Kyoomars Alizadeh Noghani, and Andreas Kassler. 2017. OpenStack-Emu - A cloud testbed combining network emulation with OpenStack and SDN. In *Proceedings of the 2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC'17)*. 566–568. DOI: http://dx.doi.org/10.1109/CCNC.2017.7983169

[11] Kashif Bilal, Saif Ur Rehman Malik, Osman Khalid, Abdul Hameed, Enrique Alvarez, Vidura Wijaysekara, Rizwana Irfan, Sarjan Shrestha, Debjyoti Dwivedy, Mazhar Ali, Usman Shahid Khan, Assad Abbas, Nauman Jalil, and Samee U. Khan. 2014. A taxonomy and survey on green data center networks. *Future Generation Computer Systems* 36 (2014), 189–208. DOI: http://dx.doi.org/10.1016/j.future.2013.07.006

[12] Roberto Bonafiglia, Gabriele Castellano, Ivano Cerrato, and Fulvio Risso. 2017. End-to-end service orchestration across SDN and cloud computing domains. In *2017 IEEE Conference on Network Softwarization (NetSoft'17)*. 1–6. DOI: http://dx.doi.org/10.1109/NETSOFT.2017.8004234

[13] Rajkumar Buyya, Rodrigo N. Calheiros, Jungmin Son, Amir Vahid Dastjerdi, and Young Yoon. 2014. Software-defined cloud computing: Architectural elements and open challenges. In *Proceedings of the 3rd International Conference on Advances in Computing, Communications and Informatics (ICACCI'14)*. IEEE, IEEE Press.

[14] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 25, 6 (2009), 599–616.

[15] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya. 2011. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience* 41, 1 (2011), 23–50.

[16] Franco Callegati, Walter Cerroni, Chiara Contoli, Rossella Cardone, Matteo Nocentini, and Antonio Manzalini. 2016. SDN for dynamic NFV deployment. *IEEE Communications Magazine* 54, 10 (October 2016), 89–95. DOI: http://dx.doi.org/10.1109/MCOM.2016.7588275

[17] Martin Casado, Nate Foster, and Arjun Guha. 2014. Abstractions for software-defined networks. *Communications of the ACM* 57, 10 (September 2014), 86–95. DOI: http://dx.doi.org/10.1145/2661061.2661063

[18] Walter Cerroni, Molka Gharbaoui, Barbara Martini, Aldo Campi, Piero Castoldi, and Franco Callegati. 2015. Cross-layer resource orchestration for cloud service delivery: A seamless SDN approach. *Computer Networks* 87, Supplement C (2015), 16–32. DOI: http://dx.doi.org/10.1016/j.comnet.2015.05.008

[19] Tao Chen, Xiaofeng Gao, and Guihai Chen. 2016. The features, hardware, and architectures of data center networks: A survey. *Journal of Parallel and Distributed Computing* 96 (2016), 45–74. DOI: http://dx.doi.org/10.1016/j.jpdc.2016.05.009

[20] Ankur Chowdhary, Sandeep Pisharody, Adel Alshamrani, and Dijiang Huang. 2017. Dynamic game based security framework in SDN-enabled cloud networking environments. In *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization (SDN-NFVSec'17)*. ACM, New York, 53–58. DOI: http://dx.doi.org/10.1145/3040992.3040998

[21] Charles Clos. 1953. A study of non-blocking switching networks. *Bell System Technical Journal* 32, 2 (March 1953), 406–424. DOI: http://dx.doi.org/10.1002/j.1538-7305.1953.tb01433.x

[22] Laizhong Cui, F. Richard Yu, and Qiao Yan. 2016. When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE Network* 30, 1 (January 2016), 58–65. DOI: http://dx.doi.org/10.1109/MNET.2016.7389832

[23] Yong Cui, Jian Song, Minming Li, Qingmei Ren, Yangjun Zhang, and Xuejun Cai. 2017. SDN-based big data caching in ISP networks. *IEEE Transactions on Big Data* (2017). DOI: http://dx.doi.org/10.1109/TBDATA.2017.2651901

[24] Richard Cziva, Simon Jout, David Stapleton, Fung Po Tso, and Dimitrios P. Pezaros. 2016. SDN-based virtual machine management for cloud data centers. *IEEE Transactions on Network and Service Management* 13, 2 (June 2016), 212–225. DOI: http://dx.doi.org/10.1109/TNSM.2016.2528220

[25] Tooska Dargahi, Alberto Caponi, Moreno Ambrosin, Giuseppe Bianchi, and Mauro Conti. 2017. A survey on the security of stateful SDN data planes. *IEEE Communications Surveys Tutorials* 19, 3 (2017), 1701–1725. DOI: http://dx.doi.org/10.1109/COMST.2017.2689819

[26] Marcos Dias de Assunção, Radu Carpa, Laurent Lefévre, and Olivier Glýck. 2017. On designing SDN services for energy-aware traffic engineering. In *Proceedings of the 11th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*. DOI: http://dx.doi.org/10.1007/978-3-319-49580-4_2

[27] Felipe Rodrigo de Souza, Charles Christian Miers, Adriano Fiorese, and Guilherme Piegas Koslovski. 2017. QoS-aware virtual infrastructures allocation on SDN-based clouds. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'17)*. IEEE Press, Piscataway, NJ, 120–129. DOI: http://dx.doi.org/10.1109/CCGRID.2017.57

[28] Xuekai Du, Zhihui Lv, Jie Wu, Chengrong Wu, and Shi Chen. 2016. PDSDN: A policy-driven SDN controller improving scheme for multi-tenant cloud datacenter environments. In *Proceedings of the 2016 IEEE International Conference on Services Computing (SCC'16)*. 387–394. DOI: http://dx.doi.org/10.1109/SCC.2016.57

[29] Hilmi E. Egilmez, S. Tahsin Dane, K. Tolga Bagci, and A. Murat Tekalp. 2012. OpenQoS: An openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks. In *Proceedings of the 2012 Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA ASC'12)*. 1–8.

[30] Flavio Esposito, Ibrahim Matta, and Vatche Ishakian. 2013. Slice embedding solutions for distributed service architectures. *Computing Surveys* 46, 1, Article 6 (July 2013), 29 pages. DOI: http://dx.doi.org/10.1145/2522968.2522974

[31] Weiwei Fang, Xiangmin Liang, Shengxin Li, Luca Chiaraviglio, and Naixue Xiong. 2013. VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers. *Computer Networks* 57, 1 (2013), 179–196. DOI: http://dx.doi.org/10.1016/j.comnet.2012.09.008

[32] Nick Feamster, Jennifer Rexford, and Ellen Zegura. 2013. The road to SDN. *Queue* 11, 12, Article 20 (December 2013), 21 pages. DOI: http://dx.doi.org/10.1145/2559899.2560327

[33] Silvia Fichera, Molka Gharbaoui, Piero Castoldi, Barbara Martini, and Antonio Manzalini. 2017. On experimenting 5G: Testbed set-up for SDN orchestration across network cloud and IoT domains. In *2017 IEEE Conference on Network Softwarization (NetSoft'17)*. 1–6. DOI: http://dx.doi.org/10.1109/NETSOFT.2017.8004245

[34] Andreas Fischer, Felipe Botero Botero, Michael Till Beck, Hermann de Meer, and Xavier Hesselbach. 2013. Virtual network embedding: A survey. *IEEE Communications Surveys Tutorials* 15, 4 (2013), 1888–1906. DOI: http://dx.doi.org/10.1109/SURV.2013.013013.00155

[35] Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. 2015. Edge-centric computing: Vision and challenges. *SIGCOMM Computer Communication Review* 45, 5 (September 2015), 37–42. DOI: http://dx.doi.org/10.1145/2831347.2831354

[36] Saurabh Kumar Garg and Rajkumar Buyya. 2011. Networkcloudsim: Modelling parallel applications in cloud simulations. In *Proceedings of the 2011 4th IEEE International Conference on Utility and Cloud Computing (UCC'11)*. IEEE Computer Society, Washington, DC, 105–113. DOI: http://dx.doi.org/10.1109/UCC.2011.24

[37] Molka Gharbaoui, Barbara Martini, Davide Adami, Stefano Giordano, and Piero Castoldi. 2016. Cloud and network orchestration in SDN data centers: Design principles and performance evaluation. *Computer Networks* 108, Supplement C (2016), 279–295. DOI: http://dx.doi.org/10.1016/j.comnet.2016.08.029

[38] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. 2008. NOX: Towards an operating system for networks. *ACM SIGCOMM Computer Communication Review* 38, 3 (July 2008), 105–110. DOI: http://dx.doi.org/10.1145/1384609.1384625

[39] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. 2009. BCube: A high performance, server-centric network architecture for modular data centers. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication (SIGCOMM'09)*. ACM, New York, 63–74. DOI: http://dx.doi.org/10.1145/1592568.1592577

[40] Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu. 2008. Dcell: A scalable and fault-tolerant network structure for data centers. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication (SIGCOMM'08)*. ACM, New York, 75–86. DOI: http://dx.doi.org/10.1145/1402958.1402968

[41] Uma Gurav and Rizwana Shaikh. 2010. Virtualization: A key feature of cloud computing. In *Proceedings of the International Conference and Workshop on Emerging Trends in Technology (ICWET'10)*. ACM, New York, 227–229. DOI: http://dx.doi.org/10.1145/1741906.1741957

[42] Vijay K. Gurbani, Michael Scharf, T. V. Lakshman, Volker Hilt, and Enrico Marocco. 2012. Abstracting network state in software defined networks (SDN) for rendezvous services. In *Proceedings of the 2012 IEEE International Conference on Communications (ICC'12)*. 6627–6632. DOI: http://dx.doi.org/10.1109/ICC.2012.6364858

[43] Pooyan Habibi, Masoud Mokhtari, and Masoud Sabaei. 2016. QRVE: QoS-aware routing and energy-efficient VM placement for software-defined datacenter networks. In *Proceedings of the 8th International Symposium on Telecommunications (IST'16)*. 533–539. DOI: http://dx.doi.org/10.1109/ISTEL.2016.7881879

[44] Brandon Heller, Srini Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, and Nick McKeown. 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, 17–17. Retrieved from http://dl.acm.org/citation.cfm?id=1855711.1855728.

[45] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. 2015. Mobile edge computing a key technology towards 5G. *ETSI White Paper* 11, 11 (2015), 1–16.

[46] Airton Ishimori, Fernando Farias, Eduardo Cerqueira, and Antonio Abelem. 2013. Control of multiple packet schedulers for improving QoS on openflow/SDN networking. In *Proceedings of the 2013 2nd European Workshop on Software Defined Networks (EWSDN'13)*. 81–86. DOI: http://dx.doi.org/10.1109/EWSDN.2013.20

[47] Raj Jain and Subharthi Paul. 2013. Network virtualization and software defined networking for cloud computing: A survey. *IEEE Communications Magazine* 51, 11 (November 2013), 24–31. DOI: http://dx.doi.org/10.1109/MCOM.2013.6658648

[48] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. 2013. B4: Experience with a globally-deployed software defined wan. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM (SIGCOMM'13)*. ACM, New York, 3–14. DOI: http://dx.doi.org/10.1145/2486001.2486019

[49] Saksit Jantila and Kornchawal Chaipah. 2016. A security analysis of a hybrid mechanism to defend DDoS attacks in SDN. *Procedia Computer Science* 86, Supplement C (2016), 437–440. DOI: http://dx.doi.org/10.1016/j.procs.2016.05.072

[50] Yaser Jararweh, Mahmoud Al-Ayyoub, Ala Darabseh, Elhadj Benkhelifa, Mladen Vouk, and Andy Rindos. 2016. Software defined cloud: Survey, system and evaluation. *Future Generation Computer Systems* 58 (2016), 56–74. DOI: http://dx.doi.org/10.1016/j.future.2015.10.015

[51] Yosr Jarraya, Taous Madi, and Mourad Debbabi. 2014. A survey and a layered taxonomy of software-defined networking. *IEEE Communications Surveys Tutorials* 16, 4 (2014), 1955–1980. DOI: http://dx.doi.org/10.1109/COMST.2014.2320094

[52] Kyuho Jeong and Renato Figueiredo. 2016. Self-configuring software-defined overlay bypass for seamless inter- and intra-cloud virtual networking. In *Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC'16)*. ACM, New York, 153–164. DOI: http://dx.doi.org/10.1145/2907294.2907318

[53] Hao Jin, Tosmate Cheocherngngarn, Dmita Levy, Alex Smith, Deng Pan, Jason Liu, and Niki Pissinou. 2013. Joint host-network optimization for energy-efficient data center networking. In *Proceedings of the 2013 IEEE 27th International Symposium on Parallel Distributed Processing (IPDPS'13)*. 623–634. DOI: http://dx.doi.org/10.1109/IPDPS.2013.100

[54] Divya Kapil, Emmanuel S. Pilli, and Ramesh C. Joshi. 2013. Live virtual machine migration techniques: Survey and research challenges. In *Proceedings of the 3rd IEEE International Advance Computing Conference (IACC'13)*. 963–969. DOI: http://dx.doi.org/10.1109/IAdCC.2013.6514357

[55] Murat Karakus and Arjan Durresi. 2017. A survey: Control plane scalability issues and approaches in software-defined networking (SDN). *Computer Networks* 112 (2017), 279–293. DOI: http://dx.doi.org/10.1016/j.comnet.2016.11.017

[56] Younghwa Kim, Saehoon Kang, Chunglae Cho, and Soomyung Pahk. 2016. SDN-based orchestration for interworking cloud and transport networks. In *2016 International Conference on Information and Communication Technology Convergence (ICTC'16)*. 303–307. DOI: http://dx.doi.org/10.1109/ICTC.2016.7763490

[57] Dzmitry Kliazovich, Pascal Bouvry, and Samee Ullah Khan. 2012. GreenCloud: A packet-level simulator of energy-aware cloud computing data centers. *Journal of Supercomputing* 62, 3 (December 2012), 1263–1283. DOI: http://dx.doi.org/10.1007/s11227-010-0504-1

[58] Spiros Koulouzis, Adam S. Z. Belloum, Marian T. Bubak, Zhiming Zhao, Miroslav Živković, and Cees T. A. M. de Laat. 2016. SDN-aware federation of distributed data. *Future Generation Computer Systems* 56, Supplement C (2016), 64–76. DOI: http://dx.doi.org/10.1016/j.future.2015.09.032

[59] Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Verssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE* 103, 1 (January 2015), 14–76. DOI: http://dx.doi.org/10.1109/JPROC.2014.2371999

[60] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A network in a laptop: Rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 19.

[61] Aris Leivadeas, Matthias Falkner, Ioannis Lambadaris, and George Kesidis. 2017. Optimal virtualized network function allocation for an SDN enabled cloud. *Computer Standards & Interfaces* 54, P4 (November 2017), 266–278. DOI: http://dx.doi.org/10.1016/j.csi.2017.01.001

[62] Melissa Licciardello, Matteo Fiorani, Marija Furdek, Paolo Monti, Carla Raffaelli, and Lena Wosinska. 2017. Performance evaluation of abstraction models for orchestration of distributed data center networks. In *2017 19th International Conference on Transparent Optical Networks (ICTON'17)*. 1–4. DOI: http://dx.doi.org/10.1109/ICTON.2017.8025184

[63] Barbara Martini, Davide Adami, Molka Gharbaoui, Piero Castoldi, Lisa Donatini, and Stefano Giordano. 2016. Design and evaluation of SDN-based orchestration system for cloud data centers. In *Proceedings of the 2016 IEEE International Conference on Communications (ICC'16)*. 1–6. DOI: http://dx.doi.org/10.1109/ICC.2016.7511144

[64] Toni Mastelic, Ariel Oleksiak, Holger Claussen, Ivona Brandic, Jean-Marc Pierson, and Athanasios V. Vasilakos. 2014. Cloud computing: Survey on energy efficiency. *Computing Surveys* 47, 2, Article 33 (December 2014), 36 pages. DOI: http://dx.doi.org/10.1145/2656204

[65] Arturo Mayoral, Raul Munoz, Ricard Vilalta, Ramon Casellas, Ricardo Martínez, and Victor López. 2017. Need for a transport API in 5G for global orchestration of cloud and networks through a virtualized infrastructure manager and planner. *Journal of Optical Communications and Networking* 9, 1 (January 2017), A55–A62. DOI: http://dx.doi.org/10.1364/JOCN.9.000A55

[66] Arturo Mayoral, Ricard Vilalta, Raul Muñoz , Ramon Casellas, and Ricardo Martínez. 2017. SDN orchestration architectures and their integration with cloud computing applications. *Optical Switching and Networking* 26 (2017), 2–13. DOI: http://dx.doi.org/10.1016/j.osn.2015.09.007

[67] Arturo Mayoral, Ricard Vilalta, Raul Munoz, Ramon Casellas, Ricardo Martínez, Michela Svaluto Moreolo, Josep M. Fabrega, Alejandro Aguado, Shuangyi Yan, Dimitra Simeonidou, Jose M. Gran, Víctor López, Pavel Kaczmarek, Rafal Szwedowski, Thomas Szyrkowiec, Achim Autenrieth, Norboru Yoshikane, Takehiro Tsuritani, Itsuro Morita, Masaki Shiraiwa, Naoya Wada, Masato Nishihara, T. Tanaka, Tomoo Takahara, Jens C. Rasmussen, Yuki Yoshida, and Ken ichi Kitayama. 2017. Control orchestration protocol: Unified transport API for distributed cloud and network orchestration. *Journal of Optical Communications and Networking* 9, 2 (February 2017), A216–A222. DOI: http://dx.doi.org/10.1364/JOCN.9.00A216

[68] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 38, 2 (2008), 69–74.

[69] Marouen Mechtri, Ines Houidi, Wajdi Louati, and Djamal Zeghlache. 2013. SDN for inter cloud networking. In *Proceedings of the 2013 IEEE SDN for Future Networks and Services (SDN4FNS'13)*. 1–7. DOI: http://dx.doi.org/10.1109/SDN4FNS.2013.6702552

[70] Violeta Medina and Juan Manuel García. 2014. A survey of migration mechanisms of virtual machines. *Computing Surveys* 46, 3, Article 30 (January 2014), 33 pages. DOI: http://dx.doi.org/10.1145/2492705

[71] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niel Bouten, Filip De Turck, and Raouf Boutaba. 2016. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials* 18, 1 (2016), 236–262. DOI: http://dx.doi.org/10.1109/COMST.2015.2477041

[72] Natural Resources Defense Council. 2014. America's Data Centers Consuming and Wasting Growing Amounts of Energy. Retrieved from http://www.nrdc.org/energy/data-center-efficiency-assessment.asp.

[73] ns2. 1995–2011. The Network Simulator NS-2. Retrieved from http://www.isi.edu/nsnam/ns/.

[74] NTT Corporation. 2017. RYU the Network Operating System (NOS). Retrieved from https://osrg.github.io/ryu/.

[75] ONF Solution Brief. 2014. OpenFlow-Enabled SDN and Network Functions Virtualization. *Open Network Foundation*. Retrieved from https://www.opennetworking.org/wp-content/uploads/2013/05/sb-sdn-nvf-solution.pdf.

[76] Open Networking Lab. 2014. Introducing ONOS - A SDN Network Operating System for Service Providers. Retrieved from http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf.

[77] OpenDaylight. 2017. OpenDaylight: Open Source SDN Platform. Retrieved from https://www.opendaylight.org/.

[78] OpenStack Foundation. 2017. Open Source Software for Creating Private and Public Clouds. Retrieved from https://www.openstack.org/.

[79] Ehsan Pakbaznia and Massoud Pedram. 2009. Minimizing data center cooling and server power costs. In *Proceedings of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED'09)*. ACM, New York, 145–150. DOI: http://dx.doi.org/10.1145/1594233.1594268

[80] Ioan Petri, Mengsong Zou, Ali Reza Zamani, Javier Diaz-Montes, Omer Rana, and Manish Parashar. 2015. Integrating software defined networks within a cloud federation. In *Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'15)*. 179–188. DOI: http://dx.doi.org/10.1109/CCGrid.2015.11

[81] Sandeep Pisharody, Janakarajan Natarajan, Ankur Chowdhary, Abdullah Alshalan, and Dijiang Huang. 2017. Brew: A security policy analysis framework for distributed SDN-based cloud environments. *IEEE Transactions on Dependable and Secure Computing* (2017). DOI: http://dx.doi.org/10.1109/TDSC.2017.2726066

[82] Lucian Popa, Gautam Kumar, Mosharaf Chowdhury, Arvind Krishnamurthy, Sylvia Ratnasamy, and Ion Stoica. 2012. FairCloud: Sharing the network in cloud computing. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'12).* ACM, New York, 187–198. DOI: http://dx.doi.org/10.1145/2342356.2342396

[83] POX. 2013. POX. Retrieved from http://www.noxrepo.org/pox/about-pox/.

[84] Project Floodlight. 2017. Floodlight OpenFlow Controller. Retrieved from http://www.projectfloodlight.org/floodlight/.

[85] Zafar Ayyub Qazi, Cheng-Chun Tu, Luis Chiang, Rui Miao, Vyas Sekar, and Minlan Yu. 2013. SIMPLE-fying middlebox policy enforcement using SDN. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM (SIGCOMM'13).* ACM, New York, 27–38.

[86] Peng Qin, Bin Dai, Benxiong Huang, and Guan Xu. 2017. Bandwidth-aware scheduling with SDN in hadoop: A new trend for big data. *IEEE Systems Journal* 11, 4 (December 2017), 2337–2344. DOI: http://dx.doi.org/10.1109/JSYST.2015.2496368

[87] George F. Riley and Thomas R. Henderson. 2010. The ns-3 network simulator. In *Modeling and Tools for Network Simulation*, Klaus Wehrle, Mesut Güneş, and James Gross (Eds.). Springer, Berlin,15–34. DOI: http://dx.doi.org/10.1007/978-3-642-12331-3_2

[88] Aris Cahyadi Risdianto, Jun-Sik Shin, and Jong Won Kim. 2016. Deployment and evaluation of software-defined inter-connections for multi-domain federated SDN-cloud. In *Proceedings of the 11th International Conference on Future Internet Technologies (CFI'16).* ACM, New York, 118–121. DOI: http://dx.doi.org/10.1145/2935663.2935683

[89] Durgesh Samant and Umesh Bellur. 2016. Handling boot storms in virtualized data centers—a survey. *Computing Surveys* 49, 1, Article 16 (June 2016), 36 pages. DOI: http://dx.doi.org/10.1145/2932709

[90] Donald J. Schumacher and William C. Beckman. 2002. Data center cooling system. (April 23, 2002). US Patent 6,374,627.

[91] Sripriya Seetharam, Prasad Calyam, and Tsegereda Beyene. 2014. ADON: Application-driven overlay network-as-a-service for data-intensive science. In *Proceedings of the 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet'14).* 313–319. DOI: http://dx.doi.org/10.1109/CloudNet.2014.6969014

[92] Justine Sherry, Shaddi Hasan, Colin Scott, Arvind Krishnamurthy, Sylvia Ratnasamy, and Vyas Sekar. 2012. Making middleboxes someone else's problem: Network processing as a cloud service. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'12).* ACM, New York, 13–24. DOI: http://dx.doi.org/10.1145/2342356.2342359

[93] Jungmin Son, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, and Rajkumar Buyya. 2017. SLA-aware and energy-efficient dynamic overbooking in SDN-based cloud data centers. *IEEE Transactions on Sustainable Computing* 2, 2 (April 2017), 76–89. DOI: http://dx.doi.org/10.1109/TSUSC.2017.2702164

[94] Jungmin Son, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, Xiaohui Ji, Young Yoon, and Rajkumar Buyya. 2015. CloudSimSDN: Modeling and simulation of software-defined cloud data centers. In *Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'15).* 475–484. DOI: http://dx.doi.org/10.1109/CCGrid.2015.87

[95] Salvatore Spadaro, Albert Pagès, Fernando Agraz, Rafael Montero, and Jordi Perelló. 2016. Resource orchestration in SDN-based future optical data centres. In *2016 International Conference on Optical Network Design and Modeling (ONDM'16).* 1–6. DOI: http://dx.doi.org/10.1109/ONDM.2016.7494057

[96] Jose Teixeira, Gianni Antichi, Davide Adami, Alessio Del Chiaro, Stefano Giordano, and Alexandre Santos. 2013. Datacenter in a box: Test your SDN cloud-datacenter controller at home. In *Proceedings of the 2013 2nd European Workshop on Software Defined Networks (EWSDN'13).* 99–104. DOI: http://dx.doi.org/10.1109/EWSDN.2013.23

[97] Adel Nadjaran Toosi, Rodrigo N. Calheiros, and Rajkumar Buyya. 2014. Interconnected cloud computing environments: Challenges, taxonomy, and survey. *Computing Surveys* 47, 1, Article 7 (May 2014), 47 pages. DOI: http://dx.doi.org/10.1145/2593512

[98] Amin Vahdat, David Clark, and Jennifer Rexford. 2015. A purpose-built global network: Google's move to SDN. *Queue* 13, 8, Article 100 (October 2015), 26 pages. DOI: http://dx.doi.org/10.1145/2838344.2856460

[99] Richard Wang, Dana Butnariu, and Jennifer Rexford. 2011. OpenFlow-based server load balancing gone wild. In *Proceedings of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE'11).* USENIX Association, Berkeley, CA, 12–12. Retrieved from http://dl.acm.org/citation.cfm?id=1972422.1972438.

[100] Xiaodong Wang, Yanjun Yao, Xiaorui Wang, Kefa Lu, and Qing Cao. 2012. CARPO: Correlation-aware power optimization in data center networks. In *Proceedings of the 2012 IEEE INFOCOM (INFOCOM'12)*. 1125–1133. DOI: http://dx.doi.org/10.1109/INFCOM.2012.6195471

[101] Wang Wendong, Qi Qinglei, Gong Xiangyang, Hu Yannan, and Que Xirong. 2014. Autonomic QoS management mechanism in software defined network. *Communications, China* 11, 7 (July 2014), 13–23. DOI: http://dx.doi.org/10.1109/CC.2014.6895381

[102] Yu Wu, Zhizhong Zhang, Chuan Wu, Chuanxiong Guo, Zongpeng Li, and Francis C. M. Lau. 2017. Orchestrating bulk data transfers across geo-distributed datacenters. *IEEE Transactions on Cloud Computing* 5, 1 (January 2017), 112–125. DOI: http://dx.doi.org/10.1109/TCC.2015.2389842

[103] Wenfeng Xia, Peng Zhao, Yonggang Wen, and Haiyong Xie. 2017. A survey on data center networking (DCN): Infrastructure and operations. *IEEE Communications Surveys Tutorials* 19, 1 (2017), 640–656. DOI: http://dx.doi.org/10.1109/COMST.2016.2626784

[104] Qiao Yan, F. Richard Yu, Qingxiang Gong, and Jianqiang Li. 2016. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys Tutorials* 18, 1 (2016), 602–622. DOI: http://dx.doi.org/10.1109/COMST.2015.2487361

[105] Soheil Hassas Yeganeh, Amin Tootoonchian, and Yashar Ganjali. 2013. On scalability of software-defined networking. *IEEE Communications Magazine* 51, 2 (February 2013), 136–141. DOI: http://dx.doi.org/10.1109/MCOM.2013.6461198

[106] Zhen Zhang, Yuhui Deng, Geyong Min, Junjie Xie, and Shuqiang Huang. 2017. ExCCC-DCN: A highly scalable, cost-effective and energy-efficient data center structure. *IEEE Transactions on Parallel and Distributed Systems* 28, 4 (April 2017), 1046–1060. DOI: http://dx.doi.org/10.1109/TPDS.2016.2609428

[107] Kuangyu Zheng and Xiaorui Wang. 2017. Dynamic control of flow completion time for power efficiency of data center networks. In *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS'17)*. 340–350. DOI: http://dx.doi.org/10.1109/ICDCS.2017.138

[108] Kuangyu Zheng, Xiaodong Wang, Li Li, and Xiaorui Wang. 2014. Joint power optimization of data center network and servers with correlation analysis. In *Proceedings of the 2014 IEEE INFOCOM (INFOCOM'14)*. 2598–2606. DOI: http://dx.doi.org/10.1109/INFOCOM.2014.6848207

[109] Kuangyu Zheng, Xiaorui Wang, and Jia Liu. 2017. DISCO: Distributed traffic flow consolidation for power efficient data center network. In *Proceedings of the 16th International IFIP TC6 Networking Conference, Networking*. IFIP Open Digital Library. Retrieved from http://dl.ifip.org/db/conf/networking/networking2017/1570334940.pdf.

[110] Kuangyu Zheng, Wenli Zheng, Li Li, and Xiaorui Wang. 2017. PowerNetS: Coordinating data center network with servers and cooling for power optimization. *IEEE Transactions on Network and Service Management* 14, 3 (September 2017), 661–675. DOI: http://dx.doi.org/10.1109/TNSM.2017.2711567

Received March 2017; revised January 2018; accepted February 2018