IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 23, 2019, accepted July 4, 2019, date of publication July 10, 2019, date of current version July 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927822

Software-Defined Cloud Computing: A Systematic Review on Latest Trends and Developments

AAQIF AFZAAL ABBASI[1], ALMAS ABBASI[2], SHAHABODDIN SHAMSHIRBAND[3,4], ANTHONY THEODORE CHRONOPOULOS[5,6], VALERIO PERSICO[7], AND ANTONIO PESCAPÈ[7]

[1]Department of Software Engineering, Foundation University, Islamabad 44000, Pakistan
[2]Department of Computer Science, International Islamic University, Islamabad 44000, Pakistan
[3]Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[4]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[5]Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249, USA
[6]Department of Computer Engineering and Informatics, University of Patras, 26500 Rio, Greece
[7]Electrical Engineering and Information Technology Department, University of Napoli Federico II, 80125 Naples, Italy

Corresponding author: Shahaboddin Shamshirband (shahaboddin.shamshirband@tdtu.edu.vn)

ABSTRACT Cloud computing concepts offer effective and efficient tools for addressing resource-hungry computational problems. While conventional methods, architectures, and processing techniques may limit cloud data center performance, *software-defined cloud computing* (SDCC) is an approach where virtualization services to all network resources in a dc are software-defined and where software-defined networking (SDN) and cloud computing go hand in hand. SDCC-related concepts change the previous state of affairs by promoting the centralized control of networking functions in a data center. A key objective of developing software-driven cloud infrastructure is that the networking hardware, software, storage, security, and network traffic management is open and interoperable. This facilitates easy installation and management of networking functions in the cloud infrastructure. Employing SDCC concepts to cloud data centers can improve resource administration challenges to a greater extent. This paper presents a survey on SDCC. We begin by introducing SDCC environments and explain its main architectural components. We identify the essential contributions of various developments to this field and discuss the implementation challenges and limitations faced in their adoption. We also explore the potential of SDCC in two domains, namely, resource orchestration and application development, as case studies of specific interest. In an attempt to anticipate the future evolution, we discuss the important research opportunities and challenges in this promising field.

INDEX TERMS Cloud computing, data centers, infrastructure management, networking, network functions virtualization, scalability, software defined networking.

I. INTRODUCTION

Cloud computing [1] is an important computing technology built around the concept of reduced investment and pay-per-use billing solutions, with cloud service providers typically employing ''pay-as-you-go'' models. The adoption of cloud-based services has become more and more pervasive, as this paradigm provides a perfect fit for a wide range of applications, e.g. those leveraging the potentialities of the IoT domain [146].

SDN is a concept in computer networking where network administrators can manage network services through flexible software-defined control and functions. The SDN concept suggests separating network control functions from its data plane for ease of administration and allows remote access to the data center (DC) switches for network traffic management [2]–[5]. This distinguishes it from conventional network administration concepts. SDN concepts are pivotal in Software-Defined Cloud Computing (SDCC) because they facilitate multi-tiered applications and ensure that user transactions are being processed within a prescribed time frame under certain Service Level Agreements (SLAs). This has been briefly explained in [6]–[10]. Since their inception, SDCC [11] concepts have addressed several issues related to Network Functions Virtualization (NFV) [11]–[13], DC and

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao Liu.

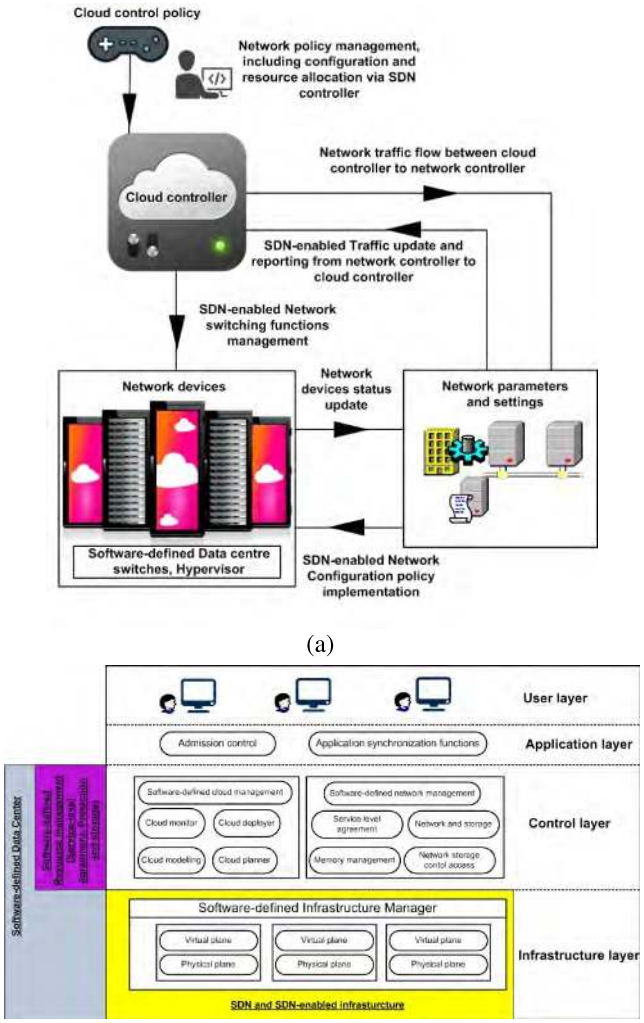

FIGURE 1. **(a) SDDC administration model. (b) Major components of SDDC.**

cloud computing models [14], [15], value-added network services deployment [16] and network management frameworks [17]–[19].

SDCC [22], [147], Software-Defined Cloud (SDC), or Software-Defined Cloud Networking (SDCN) automates data center features by employing virtualization functions to all resources and functions [14]. While SDCC concepts evolved in line with non-standard behavior of switching and routing elements in cloud DCs, the need for SDCC is of profound importance in networked environments where the standard behavior of a switch or a router is not optimized [17], [18]. Indeed, SDCC concepts facilitate processing and dynamic configuration of links and nodes through SDN controllers, removing complications in configuration and management of cloud resources and enabling network administrators to dynamically modify network configurations to uphold incoming service requests from cloud tenants.

Fig. 1(a) illustrates a generalized layout of a SDCC administration model. It presents an architecture consisting of cloud and network controllers where a network administrator can configure network traffic, SDN policies and devices through cloud controller [19]. In Fig. 1(b), we highlight the main SDCC architectural elements. The cloud controller is responsible to manage the underlying physical resources, virtual machine management, and storage allocation functions. A network controller parses the network specification into configuration commands, resulting in sets of policies to be installed on the SDN-enabled switches [20]. This process results in the centralized management of network traffic and is the fundamental concept behind SDDC paradigm.

A. CONTRIBUTION OF THE PAPER

The field of SDCC is quite new [147]. Notably, this work deals with the vast topic of SDCC, Cloud Computing and DCs. The term "SDCC" is itself complicated [21]. We therefore carefully considered in our paper only those works that qualify the definition of SDCC coined in [22] and which provide an opportunity for a complete SDCC employment in future.

1. We begin the paper by introducing the concepts of cloud computing and software-defined cloud computing and discuss the main SDCC architectural elements.

2. We present an overview of related developments of SDCC and discuss the implementation challenges in detail. We confine our discussion about implementation challenges to four major aspects, namely programmability, scalability, security, and interoperability.

3. Assuming that a neat, clear and open interface among networking devices is required in order to get aligned with the growing needs of users, we selected Meridian [23] and Frenetic [24] as our case studies. Meridian framework can be incorporated with multiple cloud controllers for bringing in the SDN advantages to cloud DCs, whereas the latter delivers a consistent way of writing and reasoning with SDN applications. A short description of available SDN controllers is also provided in Table 2.

4. In order to address the limitations faced in large-scale adoption of SDNs, we conduct a study on SDCC platforms and thereafter propose an eleven-point future direction stream.

The paper is organized as follows. Section II presents an extensive study on the building blocks of SDCC infrastructure. Section III presents related work on recent developments made to achieve SDDC benefits. In Section IV, we highlight the implementation challenges necessary to unleash the full potential of SDDCs. Limitations in large scale adoption of SDCC concepts are discussed in Section V. In section VI, we investigate Meridian [23], and Frenetic [24] frameworks as case studies. We also explore their functioning and implementation challenges in detail. The discussion in Section VII explains the current research efforts in the area, future work and available opportunities. Finally, section VIII concludes the paper.

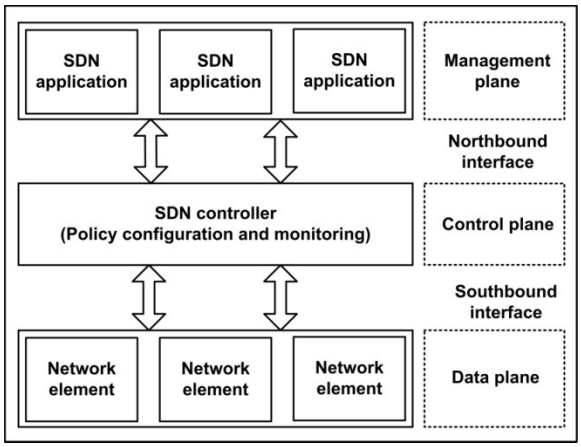

FIGURE 2. System architecture of software defined networks.

II. ARCHITECTURAL ELEMENTS

SDCC can be defined as an approach for developing cloud services where management and monitoring of all the resources (compute, storage, data center, security, SLA, etc.) are software-defined [25]. This concept enables flexible management of hardware and software resources. In Fig. 1(b), we highlight the major elements of SDCC. SDCC encompasses a variety of concepts and infrastructure components, where each component can be provisioned, operated, and managed through an API. In the following, we report a description for the main SDCC architectural elements.

A. SOFTWARE-DEFINED NETWORKING

SDN separates network control and data functions [11]. The sophistication of SDN allows it to cater to the high bandwidth needs of applications. Enterprise networks have to set up new applications and virtual machines on demand to accommodate new processing requests such as those for big data. SDN allows IT managers to experiment with network configuration without impacting the network. In SDNs, network applications running on an operating system can smoothly manage network behavior. It is because all the applications can access the same network information by using the global network view functions. A simplified architecture of SDN concept is provided in Fig. 2.

SDN design principles ensure a flexible and manageable solution to conventional networking problems. A short description of the main attributes of the SDN paradigm is reported in what follows:

1. *Flexible:* SDNs decouple network control and forwarding functions. This enables administrators to manage network functions in a flexible and hassle free way.

2. *Agile:* SDN architecture allows administrators to administer network traffic flow according to their own requirements.

3. *Manageable:* The SDN controller facilitates in providing a global view of network state which eases network management.

4. Centralized: A centralized controller device makes it much easier to access data about real time flows on the network.

5. Configurable: SDN lets network managers configure network resources themselves by using open standard software programs.

6. Neutral: SDNs open standards simplify network design, operations and frees network from vendor-specific devices and protocols.

Administrative efficiency, improvements in server utilization, better control of virtualization, and other benefits should result in operational savings. In SDNs, the major difference with respect to traditional networks is that SDN network elements only cover forwarding functions with no intelligence as the control plane functions are implemented in a distinct (centralized) location called SDN controller [3]. As shown in Fig. 3, the SDN architecture consists of the following major components:

1. *Forwarding Device:* It consists of hardware and software based devices at data plane aimed to perform basic networking operations.

2. *Southbound Interface:* It is a collection of instruction sets used as forwarding device and is defined by a southbound API.

3. *Northbound Interface:* It is the interface provided by Network Operating System (NOS) or SDN controller to develop applications. It enables users to communicate with forwarding devices.

4. *Control Plane:* The control plane carries signaling traffic for routing. It is used for controlling network functionalities and traffic rules through data plane by using southbound interface elements.

5. *Data Plane:* The data plane carries user traffic and consists of interconnected switching elements connected together through wireless radio channels or wired cables.

6. *Management Plane:* Consists of set of applications to leverage Northbound Interface functionalities by implementing network management operations. Its main purpose is to define policies which can be translated as instructions for executing different tasks.

The transformation landscape in SDN (from traditional networks) is expected to evolve in the future. With fewer developments carried out in migrating traditional networks to SDN paradigm, research efforts are underway at ONF [25], [26], ITU-T [27] and IETF [28]. SDN-enabled devices can co-exist with traditional Ethernet devices. In this regard, a solution like ForCES [29] is an approach in traditional network management where control and data planes of a networking device are separate but they exist in the same network element. This approach helps in adopting SDN features without changing the backbone of the network. In the following section, we present a quick review on developments made in switching designs and network hypervisor based solutions for enabling gradual transformation of networks from traditional Ethernet to SDN technology.

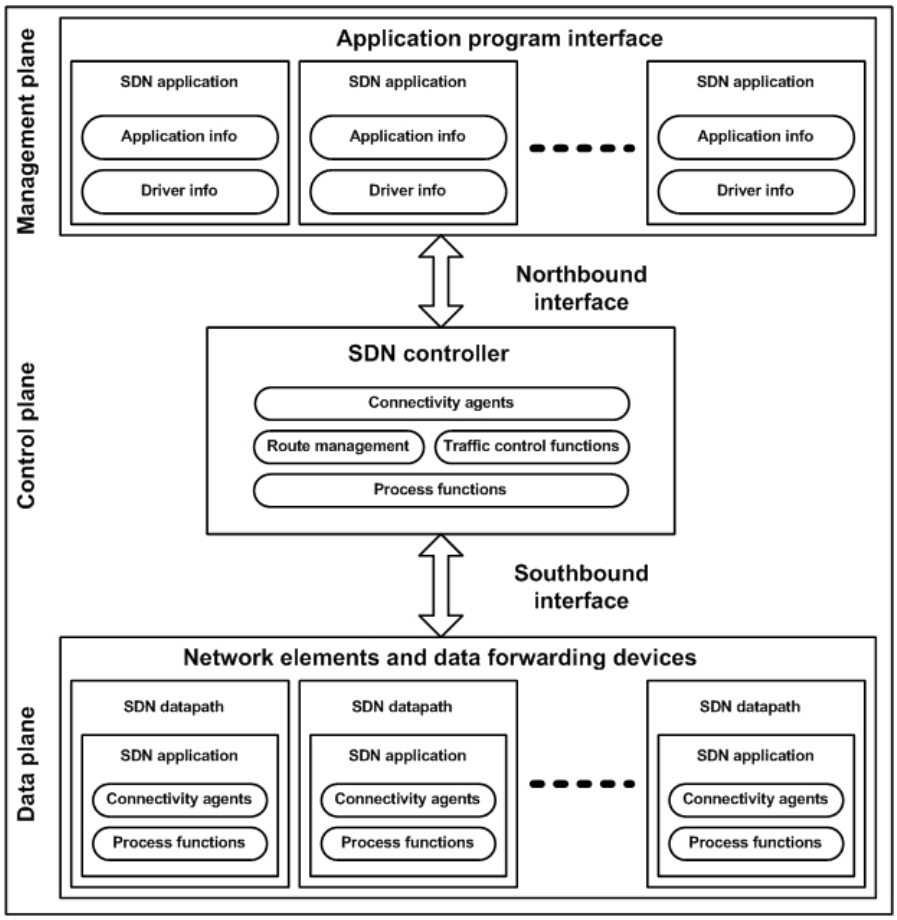

FIGURE 3. SDN planes along with their related role and functions in a network.

Juniper [30], IBM [31] and HP [32] released their SDN based switching and routing devices for DCs. These peripherals have hybrid switching capabilities to support both conventional Ethernet and SDN OpenFlow standards.

Network Hypervisor offers high-level abstractions and enables distinct virtual machines (VMs) to share hardware resources through APIs. Solutions like FlowVisor [33], Network Virtualization Platform (NVP) [34] and IBM SDN VE [35] are few of the available commercial multi-tenant hypervisors which also support conventional cloud environments.

B. SOFTWARE-DEFINED INFRASTRUCTURE

The term Software-defined infrastructure (SDI) can be defined as a technical computing infrastructure entirely managed by software without any operator or legacy software. It refers to a comprehensive, fully integrated hybrid cloud computing environment and provides the enabling ingredients for SDN technologies to work in harmony with cloud functions [35]. In SDI, all components of a DC follow software-defined principles, such as computing, storage, security and data transmission across network nodes and switches. Organizations must develop software-defined infrastructures over time, step-by-step. In order to transform conventional cloud infrastructure to SDI, DCs require SDN-enabled equipment and technologies [41]. This will ultimately give rise to a pure Software-defined environment (SDE), where the whole computing infrastructure would be software defined. Transforming conventional DCs to SDDCs can be realized by adopting a test and trial transformation procedure. A pure SDE insists on a single point of control and orchestration for cloud based services and applications [42]. This helps in an easy management and administration of DC management function. The following sections review the developments and implications of SDEs over existing clouds infrastructures. In Fig. 4, we illustrate a SDE which uses application-aware techniques to manage DC issues in real time.

SDN-enabled switches help users to meet the scalability demands for implementing private and hybrid clouds. This reduces network traffic congestion issues. They also ease the deployment of on-demand applications by using isolated virtual networks. SDN-enabled switching schemes facilitate users to program and manage issues related to network visibility, availability and changing workloads.

Novel SDN switch designs proposed in [36], [37] are appearing in numerous hardware combinations. Similarly, design solutions like the parallel lookup model [38], [39] can

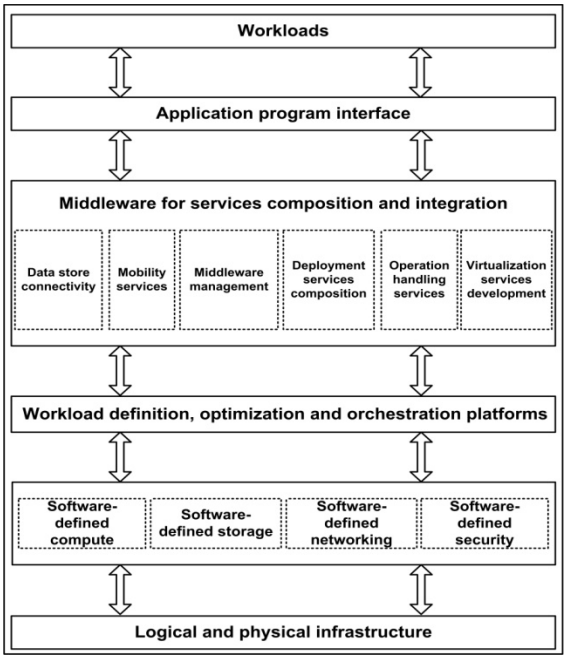

FIGURE 4. Workflow of services in a SDE.

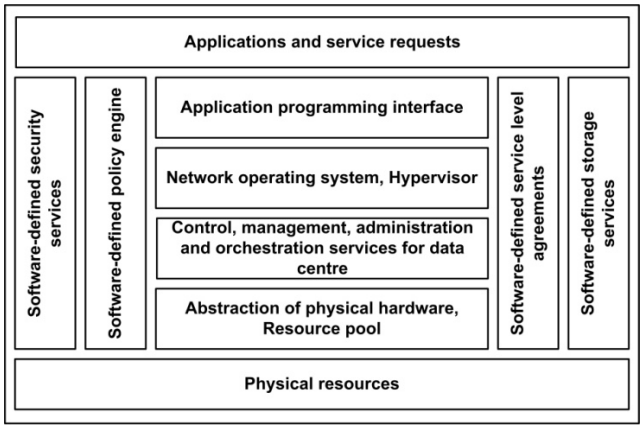

FIGURE 5. Architecture, services and roles performed in a SDDC.

also be applied to conventional cloud environments for reducing data center equipment costs. OpenFlow switch arrangements described in [40] elaborate on the ways to overcome the shortcomings of flow table sizes by using new switching design solutions.

SDEs push a data center networking approach to a level of completely virtualized environment which is based on open standards. SDEs consist of a cloud infrastructure enabled with software-defined principles and concepts. In SDEs, workloads are managed without considering the restrictions of underlying networking infrastructure, i.e. they are not technology or vendor specific [35], [41]. This approach helps in simplifying IT operations and management.

The IBM SmartCloud Orchestrator is a practical implementation of SDE [42], [43]. It facilitates existing cloud infrastructures to follow limited software-defined concepts using APIs. It also enables the resulting cloud architecture to deliver cloud services on OpenStack and Amazon EC2 platforms. More in general, in SDEs, workloads and network services are assigned to the most appropriate IT resources. Resources are selected on the basis of an application's characteristics and security. They also reduce the number of steps involved in managing public, private and hybrid cloud services by using a centralized easy-to-use interface. A comparison between the traditional approaches and SDE supported features is provided in [42], [43].

C. SOFTWARE-DEFINED DATA CENTERS

In software-defined data centers, the control of the data center is fully automated by software. It means that the hardware configuration is maintained through intelligent software systems. SDDCs extend virtualization concepts to all DC services. It also enables communication between legacy and software-defined DC peripherals. This enables network administrator to control network services, thereby shrinking or expanding network resource usage to meet desired level of service assurance.

SDDCs constitute a vision where all aspects of a DC (i.e. compute, networking, storage, security etc.) are managed through hardware independent management and virtualization system [46]. Applications running on top of SDDCs can define their own resource requirements which help in reducing operational expenditures. Architecting SDDC applications leads professionals to rethink the design, automation, orchestration and billing processes [61]. In Fig. 5, we highlight some major roles and services in a SDDC. SDDCs are still in their infancy, and will witness a great deal of innovation over the next few years. The true benefit of software definition can only be delivered through re-imagining the way DC resources are managed and controlled that transcends simple virtualization.

Presently, cloud DCs are being transformed to SDDCs on an experimental basis. This includes partial transformation, migration and integration processes where switching hardware is replaced with software-defined hardware equipment [63], [64]. This also includes integration of communication layers between legacy hardware equipment and DC facilities. Below we provide a 3 points description of SDDC infrastructure architecture:

1. *Physical hardware and legacy infrastructure:* It consists of hardware equipment that can be used for delivering virtualization services across physical or legacy systems.

2. *Management layer:* The layer comprises of a collection of development, management, monitoring and performance tuning applications which can help in administering hardware resources.

3. *Infrastructure bridging elements:* These elements integrate management applications with DC components using various SDN-enabled hardware devices and APIs.

In order to commission true SDDCs, complete transformation of all DC functions must be ensured. As SDDCs are in their trial stages, information regarding implementation of a pure SDDC is not available in literature. Thus available

literature only discusses implementation scenarios where legacy equipment is in complete harmony with SDC equipment for limited and specific range of interests. To the best of our knowledge, there is no SDDC standard currently available in the market. DMTF [44], is an association dedicated to promoting enterprise and systems suggested to develop an Open Software Defined Data Center (OSDDC) incubator [45], [46]. The OSDDC aims to develop real world based architectural specifications for SDDCs which can provide clear definitions to scope of SDDC concepts. The specific advantages of SDDC will vary from network to network, but there are benefits from network abstraction and the agility it offers for network administration and automation.

D. SOFTWARE-DEFINED SERVICE LEVEL AGREEMENT

SLAs are used to identify enterprise level service-level requirements [13]. An SLA includes penalties for non-compliance. In order to check whether an SLA is being implemented, various audit mechanisms are implemented such as the service level objectives (SLOs) [13], [23]. Concrete and measurable SLOs are often used to test that an SLA is being implemented properly. Distributed systems such as cloud data centers are difficult to design and operate. Keeping in view their complexity, SLAs must be designed to reduce service delivery constraints.

Software-defined SLAs (SD-SLAs) constitute an important part of the SDCs and their importance is expected to increase due to highly optimized service deliverance requirements in SDDCs [53]. They provide novel methods to formalize SLAs and SLOs. In SD-SLAs, a SD-SLA-aware resource manager can treat SLO configuration regardless of the vendor-specific traffic flow rules. This helps in automatic reconfiguration without further complicating the system. In Fig. 6, we illustrate an SD-SLA resource manager administering VM services through a SD-SLA orchestrator on a platform of shared network devices.

Service-specific implementation of SLAs is important to meet changing SLO requirements. In [47], [48], authors discussed guidelines for SD-SLAs in public clouds. SD-SLAs must be implemented to utilize available resources in the best possible way. SD-SLAs are vendor and technology-independent. Due to design challenges in the development of distributed systems, a variety of approaches can be implemented to manage changing requirements of DCs. In the absence of meaningful data, the original service-provider SLA is the only way to gain basic insight into network performance. With in-depth reporting, an enterprise can create its own internal SLA based on a configuration, rather than a commercial agreement. By using reporting statistics to visualize how the network is being used and by creating a user profile based on more than just an IP address, IT teams can really start to nail network performance.

E. SOFTWARE-DEFINED PROTECTION

In physical data centers, security architecture is complex. It often requires multiple servers, specialized hardware

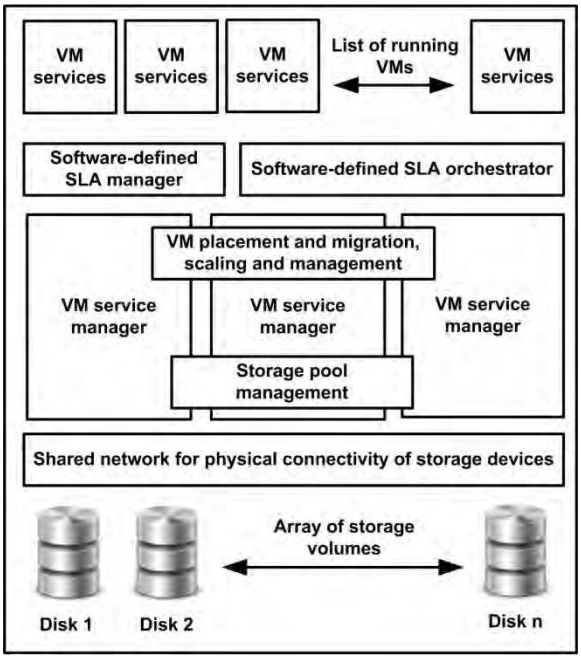

FIGURE 6. SLA provisioning in a SDDC.

devices, network identities, and more. In a SDCC, security is based on logical policies. It can be related to the concept of mathematical walls replacing the physical walls of a data center. SDP does not rely on physical location of data; information may be protected anywhere it resides. It is a practical security methodology to cope with the continuously changing security needs. It provides a multi-level security in data centers. The SDP security is managed through three virtual layers namely enforcement layer, control layer and management layer. The enforcement layer provides security functions on different segments of the network whereas control layer and management layer provide access control and interface functions, respectively. SDP offers a hassle free secure infrastructure to protect organizations using cloud services [49].

As cloud-based security provisioning systems are moving security controlling functions from network administrators to cloud data centers, SDP ensures the use of trusted channels to validate and safeguard all communications to and from the cloud. In following sections, we discuss SDP related concepts and concerns in detail.

SDP introduces simplicity in security management by bringing in logic based policies. These policies are independent of any security device. This adaptive, virtualization security is achieved by abstraction of security resources across network boundaries. SDP is also independent of data locality. It integrates a range of network security controls into a single coordinated engine [50] for intelligent analysis. This practice action is unique to SDPs and is difficult to be achieved in traditional security systems.

SDP also improves the visibility and tracking of network activity. It enables network administrators to detect

anomalous behavior of processes that would be invisible to them with physical devices. This brings a greater degree of control in network management and helps in mapping security policies for the network in a convenient way.

The use of one security policy language to manage security infrastructure in SDP enables network administrators to automate the policy execution process from a centralized location [115]. This reduces chances of human intervention errors in network and brings significant ease of administration. By offering unique function-based security architecture, SDP gives organizations an agile protection solution.

Indeed, traditional physical DC security architectures are usually rigid with conventional network security measurements relying on static machines and network identities [49]. This problem is further complicated when one-solution-fits-all approach is adopted for all applications. SDP is aimed at providing defense-in-depth protection plans. A flexible solution like SDP is best suited for DCs where a wide range of network security controls merge up into a single coordinated engine for analysis and response. As cloud based enterprise information systems are located in multiple physical sites, SDP solutions like [49], [50] tend to provide maximum security services in SDEs.

F. SOFTWARE-DEFINED STORAGE

Software-defined storage (SDS) is gaining wide attention in cloud and DC industry. It is a recent trend in the software-defined paradigm that enables enterprises & cloud providers to create shared, distributed storage resources. As a result, IDC forecasts the worldwide software-defined storage (SDS) market will see a compound annual growth rate of 13.5 percent over the 2017-2021 forecast period, with revenues of nearly $16.2 billion in 2021 [1], [18]. In SDS, storage-related controls are decoupled from the physical storage hardware. SDS is sometimes referred to as a storage hypervisor [46]. Although the two concepts are somewhat similar, yet the biggest difference is the flexibility of hosting storage control functions from any server hardware in the network.

While the concept of storage virtualization allows multiple storage devices or disk arrays to be pooled together. On the other hand, SDS is not about separating capacity from a storage device but about separating the storage control functions from the storage device.

VMWare [51] defines SDS as a fundamental component of the SDDC. With SDS, resources are abstracted to enable pooling, replication, and on demand distribution. With the emergence of SDS technology, the demarcation between network hardware and software layers will eventually disappear [52]. This centrally managed storage philosophy allows all physical and virtual resources to be visible and supports devices from different storage vendors.

SDCC can be defined as an approach for developing cloud services where management and monitoring of all the resources (compute, storage, data center, security, SLA, etc.) are software-defined [25]. This concept enables flexible management of hardware and software resources. SDCC encompasses a variety of concepts and infrastructure components, where each component can be provisioned, operated, and managed through an API. In the following, we report a description for the main SDCC architectural elements.

III. RECENT DEVELOPMENTS IN SDCC MODEL-BASED SOLUTIONS

Current SDCC environments are in an early stage of development. A full-fledged SDC environment might take years to come into existence. Current implementations involve partial deployment of SDC features or functions to only a limited area of service. In many cases, a sudden transformation to a new stream is considered risky. In this regard, the concept to outsource enterprise middlebox processing in clouds is proposed in [53]. The developed system named APLOMB (Appliance for Outsourcing Middleboxes), outsources middlebox functionalities to a third party for ease of management and reduced price. In another development, a cloud computing architecture based on SDCC concepts is presented in [22] which focuses on improving services delivery features for data-intensive applications and suggests software-defined enhancements for Cloudsim [54] simulation software. This work also provides a flexible guideline for improving existing cloud models with enhanced software-defined administration features.

A concept termed as ''Operational excellence'' [55] is presented to leverage SDCC concepts for achieving fault tolerance and recovery. The presented concept uses various configuration states to determine the sequence of system events using OpenStack. Harmony [56] presents a vision to develop an architecture which can coordinate different network services in SDDCs. By exploiting virtualization of network functions in conjunction with SDNs, NetVM [57] facilitates the development of customizable data-plane processing capabilities for VM administration in SDDCs.

Considering that DC network configurations must not change during a workload migration process, NVP based prototype approaches are proposed in [34], [58] which deliver convenient ways to handle SDC related applications.

SDCCs can provide an interface to control data organization across huge storage platforms. They also ensure that the storage infrastructure is following software-defined principles. A novel software-defined cooperative caching framework is presented in [59] which manage data placement concerns for multi-tier servers and storage applications in a coherent way. Similarly, a vision for service modularization based on real-world customer requirements is provided in [60] which describes the pros and cons of different classes of APIs from customers and cloud service provider perspective. Insights for improvement of designs of SDE and DCs are presented in [61]. The presented work also proposes an approach to honor consumer requirements in view of SLAs.

A complete SDC solution centralizes control of network infrastructure and operates across virtual- and physical-device layers. This maximizes operational efficiency with

TABLE 1. A list of past programmability efforts.

Name	Innovation factor	Short description
SOFTNET [76]	Data packet tracing	SOFTNET was capable of adding content details to network packets.
OPENSIG [77]	Network switching management for signaling	OPENSIGN led to the development of label switching functions.
Active Networks [78]	Distributed network resource management	A programming interface that exposed network resources on individual network nodes.
DCAN [79]	Centralized network traffic management	It provided infrastructure for scalable control of ATM Networks.
4D Project [80]	Uniform network behavior for light traffic	It used dedicated protocols to govern the interaction of network elements.
NETCONF [81]	Parallel and distributed traffic control	Provided parallelism implementation in hybrid switches.
Ethane [82]	Network policy enforcement	It uses a centralized controller for managing policies and security functions in networks.

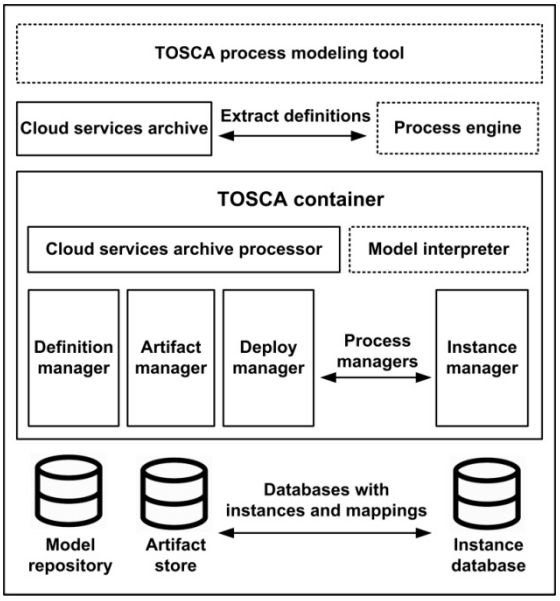

FIGURE 7. Sample architecture of a TOSCA cloud environment.

automated network configuration and management. In [55], authors describe software-defined technologies benefiting cloud service providers and end-users. They present IBM's vision for SDN, virtualization and underlying physical network infrastructure.

The easy creation and manipulation of SDEs enable model-based deployment of various market-oriented standards. One of these upcoming standards is TOSCA [62], [63]. A simplified architecture of TOSCA cloud environment is presented in Fig. 7 where TOSCA modeling container coordinates with cloud services allow processors to perform network application management.

In order to create, allocate and control heterogeneous infrastructure resources, a hyper-converged computing architecture is presented in [64]. The proposed system supports workload processing for basic DC infrastructure applications to resource-hungry analytics applications.

The existing work on SDCC is mostly related to areas like SLA management, middlebox configuration, and SDN controls for DCs. By presenting a comprehensive view on the potential of SDCs, we discuss their implementation challenges and opportunities in the following sections. We believe that the explosive growth witnessed in mission-critical and performance-sensitive applications in SDCC paradigm is an encouraging move for its future development.

IV. IMPLEMENTATION CHALLENGES

In this section we highlight the implementation challenges. The discussion has been restricted to four major domains i.e. programmability, scalability, interoperability and security for SDCC.

A. PROGRAMMABILITY

Programmability in a network enables it to accommodate a higher level of network services. In Table 1, we give a short outline of past programmability efforts in networking. Current DC hardware and management architectures are often rigid. This restricts administrators to follow vendor-defined and hardware-specific rules. A series of research efforts are related to make the programmability of existing network technologies a possibility [2], [24], [65]. These efforts will help in developing better tools for debugging and testing networks [66], [67]. Following sections provide a detailed insight on contributions made by SDCs in achieving programmability for cloud networking infrastructures.

Conventional DCs face the challenges of handling high-performance packet flows to provide connectivity between servers. They use scaling and server overbooking methods to reduce service costs. C-Through [68] and Helios [69] present two major hybrid packet and circuit switched DC network architectures. These architectures improve

circuit-switched network performance by enabling them to supply higher bandwidth to applications. This helps in achieving the balance between inter and intra-DC network traffic demands. Intelligent switching schemes are considered to be highly successful as they bring switching information to a single logical file system for analysis [70], [71]. Intelligent switching schemes also lower network asset utilization by using best-fit switching hardware devices.

Self-adaptable services and their composition help network administrators to manage load shifts, traffic routing and control features in a SDDC. Self-adaptability concepts are gaining importance in clouds and distributed systems. It is because the middleware infrastructure plays an important role in self-adaptive services composition. SDCC concepts open network to application developers, who may write applications to manage network elements and data flows according to their requirements e.g. Google uses a virtual network overlay and switching fabric to connect cluster routers [72]. Similarly, SDN interface to routing system (SDN I2RS) [2] describes a number of possible uses for developing an interface between SDN and its controlled applications. DCs worldwide energy consumption from 2005 to 2010 has risen by 56% [73].

A novel novel energy efficient flow scheduling and routing algorithm for SDN-enabled DC networks is proposed in [148]. By bringing in programmability features to control management interface, OpenStack Neat [74] can help DC administrators to reduce energy consumption in conventional as well as pure SDDCs prototypes. It is an extended version of OpenStack that helps in reducing energy consumption by reallocating VMs using live migration schemes. Extending the same approach to SDDCs enables them to save power usage to a greater extent.

SLAs must be monitored carefully to ensure that no service operations and rules are violated. A programmable management framework has been presented in [75] which use SDN principles to address multiple SLA monitoring and compliance concerns.

B. SCALABILITY

SDDC promises to deliver easier design, operation and administration of cloud infrastructures. This enables them to develop networks that can accept the changing system requirements [84], [85], [87], [92]. Cloud operators like Google and Yahoo employ large scale parallel processing algorithms to manage scalability challenges. These companies also use high-performance network services to provide efficient connectivity between physical servers. SDCC architectures can resolve these problems by providing hyper-scalability in DCs which may result in improved DC performance. In the following sections, we examine that how SDCC scalability enhancements are influencing conventional cloud DC architectures.

Applications running on VMs in a DC must be scalable. When scalability requirements increase, additional VMs are required to process the workload [145]. SDN controllers [83] manage workload related challenges to ensure seamless integration and processing of these applications. SDN controllers also facilitate network administrators to develop and customize network environments where network traffic engineering schemes can be developed and tested.

SDC concepts push network control functions to a centralized controller. This brings up some scalability concerns. As networks expand, more network requests are sent to the controller, until a point comes when it can no longer handle all incoming requests. This issue can be resolved by using the concepts of parallelism [84], [85].

SDDC functions when processed on a high scale, can overload a centralized controller very easily. This problem can be resolved by installing rules on network switches. Implementing rules on switching eliminates scalability bottleneck issues before they can affect the overall system performance.

Frameworks such as Onix [17], Hyperflow [86] and Kandoo [87] provide solutions to these problems. Onix [17] is a network-wide control platform running on multiple devices to oversee a set of switches. It is often used to help in scalability issues. Hyperflow [86] is a logically centralized and physically distributed framework. It gives applications a consistent and durable control over scalability issues. Kandoo [87] framework employs a tier-based method to manage controller traffic. Table 2 contains a list of famous SDN controllers with a short description.

Cloud architectures support the deployment and migration of applications between different cloud tenants. This is made possible by the cloud manager. AutoSlice [104] is a software-defined virtualization proposal that automates the deployment of SDN features for intermediate mediation services. This helps cloud managers to accommodate and facilitate a large number of tenants.

C. INTEROPERABILITY

SDCCs interoperability concerns still remain a challenge for the IT industry. Standardization efforts support a smooth transition from traditional cloud environment to SDE. It is simple and convenient to deploy a completely new SDDC because all its elements and devices would be software-defined [101]. However, it is not the same when it comes to transforming existing DCs to SDDC pattern. There is a large number of conventional networking equipment and swapping them with SDN-enabled infrastructure is not possible. Swapping out option is often suitable for closed environments such as a testbed or a campus network etc.

Keeping in view the current scenario, it is difficult to collect any live data from SDN related networks on the internet [105].

Open Data Center Alliance (OCDA): OCDA is unifying DCs migration to cloud computing environments. By using interoperable solutions, it proposed a detailed documentary on SDN use case models for cloud DCs [106]. It also suggested standard SDN requirements for deploying cloud datacenters.

TABLE 2. List of SDN controllers with short description.

Controller	Language	Latest version	Control architecture	Reliability
Onix [17]	C, Python	v1.0	Physically Distributed Logically Centralized	Good
Meridian [23]	Java	v1.0	Physically Distributed Logically Centralized	Very Good
Hyperflow [86]	C++	v1.0	Physically Distributed Logically Centralized	Good
Kandoo [87]	C, Python	v1.0	Physically Distributed Logically Centralized	Limited
Beacon [88]	Java	v1.0	Physically centralized	Good
NOX [89]	C++	v1.0	Physically centralized	Limited
Floodlight [90]	Java	v1.1	Physically centralized	Limited
MobileFlow [91]	Java	v1.2	Physically centralized	Limited
Maestro [92]	Java	v1.0	Physically centralized	Very Good
MUL [93]	C	v1.0	Physically Distributed Logically Centralized	Limited
ONOS [94]	Java	v2.0	Physically Distributed Logically Centralized	Good
DISCO [95]	Java	v1.1	Physically Distributed Logically Distributed	Limited
SmartLight [96]	Java	v1.0	Physically Centralized	Very Good
Prog. Flow [98]	Java	v1.0	Physically Centralized	Very Good
POX [98]	Python	v1.0	Physically Centralized	Limited
Ryu [99]	Python	v1.3	Logically Centralized	Limited
SNAC [100]	C++	v1.0	Physically centralized	Good
Trema [101]	C, Ruby	v1.0	Physically Distributed Logically Centralized	Good
NOX-MT [102]	C++	v1.0	Physically centralized	Good
OpenDaylight [103]	Java	v1.3	Physically distributed	Good

Alliance for Telecommunication Industry Solutions (ATIS): It addresses programmability standards for interoperable telecommunication devices.

European Telecommunication Standards Institute (ETSI): Its focus is on developing standards for key enabling technologies like the SDN and NFV for IT and Telecommunication industry.

In order to ensure quick transformation from conventional DCs to SDDCs, organizations should recognize the need of inter-DC traffic architectures and rules. These rules can be implemented on DCs after making regulatory changes in enterprise SLAs. In order to accomplish a complete transformation solution, a company needs to develop a product development agenda. In Table 3, we present a list of software-defined cloud orchestration platforms with a short description of their components and applications.

Due to lack of standardization, SDN controllers developed by different vendors may exhibit contrary behaviors. Using different controllers also leads to network traffic bottlenecks. An interoperability standard solution may also reduce these complexities [112]. A standard or multi-vendor supported SDN controller will play a key role in answering controller interoperability related issues in SDDCs.

Rapid advancements in cloud and mobile cloud technologies resulted in development of resource intensive applications [53]. The Telco cloud network is evolving towards an orchestrated ecosystem of vendor agnostic hardware, multi-vendor virtualized network functions and orchestration platforms. Organizations like ATIS and ETSI are developing standards for a unified multi-services orchestration platform that can manage Telco cloud related functions [106]. An ideal environment to operate these multi-vendor orchestration platforms is possible once they follow standard operating instructions. As SDCC is in its early phase of development, its interoperability issues still need to be addressed in all domains. Several proposals and techniques have been presented [53], [113], [114] to reduce interoperability concerns and costly migration to SDC environments.

D. SECURITY

SDCC integrates a range of network security control services into a single pane of glass view for analysis and control [115], [117]. This orchestration is critical for compliance requirements. It is because SDC security policies and their related security events can be integrated into a real-time policy driven system. This will result in minimizing security overheads on shared resources. It is noteworthy that achieving the same results over traditional DC based approaches is expensive and complex [122]. This dynamism of SDDCs is mostly due to their flexibility, virtualization, and on demand resources provisioning capability. SDP mechanisms introduce simplicity by delivering protection models. They improve DC security in the following ways,

1. Assist administrators in providing easy ways of detecting activity using previous logs, policy matching, and result-based optimization approaches.

2. Establish measures to determine and minimize policy conflicts among SDN controllers.

3. Facilitate services management for bandwidth allocation under predefined criteria.

TABLE 3. List of Software-defined cloud orchestration platforms with small description.

Platform	Short Description	Major Components	Implementation	Applications
CloudStack [107]	Deploy VMs as highly scalable IaaS	Clusters, Pods, Zone	Plugin-based	Open Contrail
OpenStack (Neutron) [108]	Provide fine-grained control by authentication	Plugins, RPC agents and Task schedulers,	Plugin-based	Open Daylight, HP VAN SDN, Open Contrail
OpenNebula [109]	Provide flawless adaption in heterogeneous networks	H/W and Cloud Plug-ins	Through OpenNebula or One NOX	Apache IaaS, EC2
Virtualized Services Platform [110]	Break restricted constrain responsiveness	Virtual services directory with routing functions	By using Virtual services gateway	Built-in DC optimization tool
Eucalyptus [111]	AWS compatible platform offering Virtual private cloud	Load balancer for auto scaling of workloads	By using Amazon Web Service API	EC2, EBS, S3
Meridian [23]	Provide global annotated view of DC topology	A middlebox with high-performance task scheduler	By using IBM Smart Cloud Provisioning (SCP) API	OpenStack Quantum Network Manager, IBM Tivoli

In the following sections, we discuss various aspects of SDP in delivering enhanced security services to cloud tenants.

Role based authorization services restrict system access to authorized users. They are attracting increasing attention in large network applications. Although these authorization services have often been criticized for their complexity in setting up an initial role structure, yet they have been successful in distributed environments. Role based authorization features are now being extended in DevoFlow [84], Beacon [88], NOX [115], and Maestro [92] controllers to address runtime authorization based security threats.

Threat modeling can help in eradicating major security threats. As major threat modeling methodologies lack automation, it is very difficult to scale them in large DCs. Techniques like STRIDE and P.A.S.T.A. (Process for Attack Simulation and Threat Analysis) perform automated security functions and can be implemented in SDDCs. Employing P.A.S.T.A. [116] in SDDC can reduce the high costs of security vulnerability. Its methodical approach makes threat identification very smooth and convenient. STRIDE on the other hand analyzes a network for its susceptibility to threats. STRIDE has also been employed in [117] to enumerate potential vulnerabilities of OpenFlow on widely used virtual switch and controller applications.

A security policy implementation plan consists of two major parts. The first part prevents external threats and maintains the integrity of the network, while the second part reduces internal risks by defining appropriate measures for utilizing network resources.

Implementing an Acceptable Use Policy (AUP) on basis of SLAs require extensive planning. It is because rule conflict detection in conventional cloud technology is an exhaustive process. In this regard, FortNOX [115], a role-based security enforcement kernel for NOX OpenFlow controller is presented. It checks flow rule contradictions in real time. Its security kernel detects security policy violations. In another relevant approach, VeriFlow [118] considers slicing of network functions into classes for checking invariant property violations by implementing security algorithms.

While SDDCs promise an enhanced and agile security solution through a trusted, automated and multi-vendor management platform, Unified Security Policy (USP) [50] implementation can be the right solution. USP ease the management of complicated network policies by delivering information about policy restrictions. In SDDCs, introducing USP capabilities can also ensure the implementation of consolidated security policy updates.

SDDCs can facilitate the enforcement of appropriate multi-functional security policies. These policies can be implemented by using the concept of trust zones [50], where each trust zone can execute security policies through attached hypervisor interfaces. This concept enables an automated enforcement of security policies to alarm on potential security provocations and provides instant compliance reporting for major standards and mandates. We present a comparison

TABLE 4. A comparative view of SDP functions in SDDCs.

Platform	Conventional approach	SDDC approach	Limitations	Cost effectiveness
Security policy management	Policies are implemented using code groups, set of permissions and memberships	Policies are implemented under unified programmatic control with global network mapping	Policy management at different level restricts certain users from accessing required information.	SDDC policy enforcements in conventional DC's are cheaper.
Implementing Security functions	Automated Malware Quarantine (AMQ) isolates insecure network devices	Security functions only isolate suspicious flows from others for monitoring purposes	Fine grain security policy implementation is difficult.	A one-solution-fits-all approach for SDDC makes security implementation cost effective
Security breach	A user is denied from accessing network resources	User access to information and resources is limited	SDDC has improved solutions to handle security issues however it needs consistent supervision.	Security breach violation solution for SDDCs is expensive yet more effective.
Threat Monitoring	Security solutions are employed by keeping in view DC's hardware equipment and standards	A logically centralized control allows effective threat monitoring across the network	Implementing too many security monitoring policies might slower down the system performance.	Threat monitoring in SDDC is costly as it requires dynamic threat monitoring solutions for user.
Upgrading security features	Finite resources in embedded devices make security upgrade difficult for conventional DCs	A centralized Virtual Execution Environment (VEE) enables rapid upgrading of security features	Rapid upgradation of security features makes SDDC systems slows down system response time.	SDDCs security upgradation is cost effective as compared to conventional DCs.

between conventional cloud and SDDCs security functions in Table 4.

Complete data center transformation to software-defined principles will take years. The existing data center transformation initiatives include standardization/consolidation, virtualization, automation and security enhancements. During to current rigid state of data centers, the transformation process is facing numerous challenges. In Table 5, we summarize the research work on implementation challenges in transforming data centers mainly in terms of programmability, scalability, interoperability and security.

V. LIMITATIONS IN LARGE SCALE ADOPTION

SDCC concepts are at very early stage of development. However, many software and hardware vendors have already started selling products to enable the SDCC paradigm. These products include a wide range of virtualization services, management and orchestration platforms, storage resource managers, and hybrid-cloud deployment solutions. Considering the expected developing pace of SDDCs, enterprises should adopt these infrastructures.

Considering the current hypothetical state of SDDCs implementation, organizations can initially start by benefiting from basic SDN control features in their DCs [41]. To translate the SDCC vision into reality, the industry has to overcome several challenges, including but not limited to the following:

1. Standardization: the rise of software-centric cloud networking will shift the burden of innovation from equipment vendors to developer communities, but for that to happen, developers and users will have to ensure that SDN standards are completely open and interoperable. A number of proposals address intra- and inter-networking challenges problems for open source systems and DCs [119], [120], [121], but no explicit solutions for SDCC has been reported at this stage. A common open standards-based framework for SDDC to leverage the implementation and interoperability concerns is therefore necessary.

2. Multivendor coordination: due to multivendor coordination, SDN-enabled switches support traffic control, but to ensure their continued support, consistent coordination among vendors is required. The need of specific traffic control policies for centralized network environments and lack of explicit service models for service deliverance are major obstacles in their large scale adoption.

3. Data center communication: there is an emerging understanding that transformation of conventional DCs to SDDCs can provide a number of new opportunities for network service providers. However, these opportunities also face a lot of challenges. SDN adoption challenges and their solutions have been addressed in [122]. Other challenges include inter-DC communication limitations over large scale-cloud environments as its performance may heavily impact the QoS of deployed services and requires purposely-designed approaches to be monitored [138], [139].

4. Orchestrating virtualization functions: *i*n recent years, cloud systems (e.g. DC hardware and NFV functions)

TABLE 5. Summary of implementation challenges.

Paper	Driver																
	Programmability						Scalability			Interoperability			Security				
	Hardware-specification rules	Data packet flow	Scaling and overbooking	Intelligent switching	Self-adaptable services	SLA monitoring and compliance	Hyper scalability	Parallelism	Scalability bottleneck issues	Standardization	Multi-vendor eco-system	Vendor agnostic orchestration	Role based authorization	Threat modeling	Acceptable use policy	Agile security protection	Unified security solution
Bannour et al. [2]				✓				✓			✓						
Son et al. [17]		✓	✓				✓	✓									
Foster et al [24]	✓	✓		✓					✓		✓	✓	✓				
Griffioen et al. [53]	✓	✓				✓		✓			✓	✓					
Garcia et al. [65]		✓					✓	✓		✓	✓						
Gao et al. [66]		✓		✓	✓	✓					✓						
Shrivastava et al. [67]				✓											✓		
Foerster et al. [68]	✓								✓								
Shao et al. [69]				✓			✓	✓									
Wu et al. [70]			✓								✓			✓			✓
Calcaterra et al. [71]		✓					✓	✓	✓			✓					
Kaur et al. [72]											✓						
Cao et al. [73]		✓		✓				✓									
Hawilo et al. [75]	✓			✓			✓	✓				✓					
Cokic et al. [84]			✓	✓							✓						
Sultana et al. [85]									✓								
Nobre et al. [86]			✓					✓				✓					
Nayyer et al. [87]	✓	✓			✓		✓	✓									
Chuang et al. [92]							✓	✓									
Oliveira et al. [105]		✓												✓	✓	✓	
Montero et al. [112]										✓		✓					
Garg et al. [113]				✓								✓	✓				
Liu et. al. [114]							✓										
Birkinshaw et al. [115]														✓	✓	✓	
Lee et al. [117]								✓						✓		✓	
Farris et al. [122]		✓											✓				✓

witnessed a series of dynamic changes. One of the major reasons behind these rapid changes is the increasing variation in user requirements [111]. Virtualization concepts when intertwined with SDN concepts bring mutual benefits to network applications. At present, SDDC architectures provide support for both conventional and software-defined cloud concepts. Currently, SDDCs implementations employ conventional cloud orchestration platforms to manage routine DC virtualization functions while software-defined concepts are used to control switching and routing functions. A plug-in based approach [123] can also be used to put cloud orchestration and virtualization functions under one umbrella. This will ease services delivery features to greater extent.

5. Network monitoring: the role of monitoring activities in high performance networks is critical for their management. In the short term, the SDDC operations can be supported by the already existing network management protocols such as SNMP, NETCONF, etc. [60], [100], [111]. Unfortunately due to the absence of proper management interface standards, it is hard for software-defined infrastructure to use third-party management solutions. A proper information exchange between these platforms is therefore desired and can be provided by user-friendly interfaces. In this regard, monitoring solutions leveraging the mandatory messages of the protocols adopted in SDN architectures (e.g. OpenFlow) have been also proposed to monitor QoS parameters [140]–[142]. Finally, techniques leveraging *non-cooperative* approaches have been also proposed to obtain additional knowledge about cloud performance [139], [143], [144]. These techniques allow network administrators to allocate network traffic resources on basis of user profiles.

VI. CASE STUDIES - SOFTWARE-DEFINED CLOUD ORCHESTRATION FUNCTIONS: MERIDIAN AND FRENETIC

SDC platforms support a variety of services. With cloud applications demanding greater flexibility and access rights over network, SDN concept seems a natural approach. The emergence of software defined paradigm in cloud computing also provides opportunities to seamlessly integrate applications through user-friendly interfaces and automation.

In this section, we explore the Meridian [23] and the Frenetic projects [24] as our case studies to understand the real time threats and challenges concerning SDEs. Meridian cloud platform architecture is inspired by SDN model and can support several cloud orchestration platforms. On the other hand, Frenetic replaces the available low-level imperative interface by delivering intuitive abstractions for programming. These platforms have been studied in the following sections.

A. MERIDIAN – THE SDN PLATFORM FOR CLOUD COMPUTING

Meridian is a SDC framework which supports service-level model to deliver cloud services. Meridian also supports services related to topology views which are used to gather performance metrics and statistics for various functions of the cloud network.

Meridian service model functions in terms of logical topologies. It delivers a service-level network model which specifies services associated with the VMs. Meridian components provide associated APIs with information to interact with a network through the cloud controller. Following is a short description of its architectural components.

Meridian employs *entities* to identify virtual links and construct connectivity topology among VMs. The *Planner* maintains a flow of scheduled tasks. It decides whether to execute these tasks in a parallel or a sequential mode.

The *Deployer* acts as a central point in Meridian's architectural hierarchy from where network commands are sent.

The Quantum plug-in [124] was developed for mapping basic Quantum constructs to Meridian network model. Afterwards, a Meridian virtual network for enabling all-to-all communication was developed which enabled Quantum network manager to function by using Meridian standard APIs. This integration enabled Meridian to offer high-level connectivity and policy abstractions for cloud applications.

IBM SCP [125] combines infrastructure and platform management capabilities to deliver virtualization services in cloud data centers. Meridian's integration with SCP uses network robots (or bots) to manage image, volume and computing resources.

A new network bot was created to support Meridian's network service features. This enables Meridian to communicate with SCP through the network leader bot.

A major challenge in Meridian implementation is its capacity for supporting large number of network requests. Improvements are underway to improve its topology discovery services. Our selection of Meridian as a case study is its flexible virtual network support architecture.

This supports a large variety of application topologies and cloud controllers. A short description of Meridian's architectural components and functions is given in Table 6.

B. FRENETIC – PROGRAMMING THE SDN APPLICATIONS

SDCs allow to deploy existing applications and new ones. They ensure a neat and clean environment between networking devices and their applications. Overall, application development in today's SDN-enabled controllers is a difficult task [82], [127]. Indeed, for load balancing among back-end servers, a controller can split flows over several server replicas which are too difficult to implement. Protocols such as OpenFlow [11] directly communicate with the underlying switching hardware. Similarly, controllers like Beacon [88], NOX [89] and Floodlight [90] also support the same low-level interface. It is therefore necessary to develop a mechanism which can support multiple-level coordination simultaneously.

The goal of Frenetic project [24] (also known as Pyretic) is to facilitate the creation of an environment for developing software-defined applications [25].

Frenetic is used to query network state and define policies. This process is completed in two steps: *policy composition* and *packet flow update.*

TABLE 6. Simplified overview of meridian [23] architecture and functions with short description.

Architectural components, layers	
API layer	Provides APIs with information to interact with a network
Orchestration layer	Performs API calls conversion, QoS, service insertion
Application layer	Manages network integration points
Driver layer	Enables controller to interface with the network
Implementation schemes	
Endpoint	Represents a virtual network interface on a VM
Group	Simplifies and share policy implementation
Service	Describes service roles during connectivity
Segment	Provides connectivity path among communicating groups
Virtnet	Creates a logical network domain for every single cloud tenant
Plan execution strategies	
High level plans, Low level plans	Execute plans using virtual networks, and network groups
Network deployment commands	
Validate ( ):	Perform error checking, network creation
Install ( ):	Installs commands
Undo ( ):	Performs reverse install operations.
Resume ( ): Suspend ( ):	Resumes or halt service execution plans

During the policy composition step, policies are associated with predefined criteria. The aim of performing this function is to ensure policy enforcement. On the other hand, during the packet flow update, a per-packet consistent update policy is used. This policy guarantees that all network packets are forwarded and processed by using the same policy.

In practice, implementing policy based approaches and optimizations may slow down the network performance [126]. If a network topology modification is made during run time, Frenetic can update the switching information to improve network performance.

We present a simplified overview of Frenetic functions in Table 7.

Orchestration is often known as the automated configuration and management of computing systems. In data centers, cloud orchestration platforms function as a tool for management of interconnections and interactions among workloads. In Table 8, we present a checklist of important software-defined cloud orchestration platforms on basis of cloud-services orchestration and other specific functions.

VII. RESEARCH EFFORTS AND FUTURE DIRECTIONS

In this section, we highlight some important research streams in SDCC.

1. Network management system: Network management systems like Procera [128] and network configuration languages like Frenetic [24] are required to deliver enhanced network administration features. To further improve support for network management systems, solutions pertaining to specialized hardware, operating systems, and networking applications can also be presented.

2. Load balancing and route optimization: Customized APIs can be developed to improve performance optimization features in SDDCs. As we witness major contributions in path exploration [129], route withdrawal [130], latency convergence [131] and network views optimization [24], there is a huge scope for developing performance tuning related applications for SDDCs.

3. Content delivery: Increased support for content delivery services in SDNs is presented in [132]. Efforts can be made to realize Information Centric Networking (ICN) development through SDCC concepts.

4. Policy enforcement and validation: Floodlight [90], Procera [128] and Mirage [133] present scalable solutions for policy isolation and validation features using SDNs. In order to enable SDDCs to access required NFV features, policy enforcement and validation schemes can be developed to administer real-time traffic challenges.

5. Autonomous system: Autonomous system concepts and their functional roles can be incorporated in SDDCs. This will help SDDCs to deliver self-healing and self-management capabilities to enhance users' service quality.

6, SDN Controller design and network traffic distribution: The controller placement design issues presented in [134], [135] can be resolved by improving scalability related concerns of SDN controllers. Proposals can also be made to

TABLE 7. Simplified overview of frenetic functions.

Design modules	Functions	Short description
Control loop architecture	Querying network state	Query network state and handle switching counters.
	Expressing policies	Specify packet-forwarding behavior, topology discovery and load balancing features.
	Reconfiguring network	Updates global network configuration.
Querying design parameters	High-Level predicates	Invoke and update packets over standard OpenFlow headers.
	Dynamic unfolding	Processes future traffic from hosts.
	Limiting traffic	Handles extra packets by applying forwarding policies.
	Polling	Performs queuing operations.
Policy composition	High-level predicates	Measures the traffic volume on a particular link.
	Dynamic unfolding	Performs MAC learning to identify interfaces for communicating hosts.
Policy update	Per-Packet- Per-Flow Consistency	Guarantees packet flowing processing through one forwarding policy.

TABLE 8. SDC functional classification.

	Cloud services orchestration	Querying network state	Network policy implementation	Packet-based traffic flow consistency	Polling (Queuing)	Interface and topology based optimization	Predicate-based policy implementation
Meridian	✓	Partly	✓	✓	✓	✓	Partly
Frenetic	Partly	✓	✓	✓	n/a	✓	✓
CloudStack	✓	Partly	✓	n/a	✓		✓
OpenStack	Partly	✓	n/a	Partly	✓	✓	Partly
OpenNebula	✓	n/a	Partly	n/a	✓	✓	n/a
Virtualized Services Platform	✓	n/a	✓	✓	✓	✓	✓
Eucalyptus	Partly	✓	n/a	✓	✓	n/a	n/a

deliver efficient solution for cloud controller related issues like flow delays, overheads and data modeling in near future.

7. Heterogeneous deployment solutions: SDDC deployment needs a shared, compounded, and well-managed physical medium which can ensure a decentralized environment that is free of disruptions and delays [7], [57], [64]. Efforts can be made to develop solutions to deploy a SDE alongside legacy infrastructure. These solutions can unify legacy and latest DC peripherals, allowing users to experience an accelerated service innovation. This can also help in reducing costs and will protect the current investments in cloud business.

8. Internet Exchange Points (IXPs): Deploying SDN schemes at IXPs offers new solutions for solving inter-domain routing challenges. Thus, SDX controller [136] is a vital development in this regard as it provides sequential

composition of policies that change inter-domain routing. The flexibility provided by the SDNs enabling their usage in DCs and IXPs can be further explored in the future.

9. Development of migration schemes: Complete migration from conventional cloud architectures to SDDCs is a gradual and step-by-step task [39]. This may also require rewriting network policy configurations from the beginning, which is one of the reasons why administrators are reluctant to transfer their systems to software-defined environments. Exodus [137] suggests generating network topologies that are functionally similar to the original networks. This methodology helps to identify the network topology related changes made during a migration process. Applications can be developed in the future to disintegrate and reduce dependencies in cloud migration processes.

10. Power management models: Power management techniques for reducing DC energy consumption can deliver significant opportunities for operational cost savings and other business values [38], [42], [74]. In many areas, energy reduction initiatives can actually be used for generating revenue. These power management and efficient energy model proposals can be further pursued to develop an all together green DC.

11. Language for developing function models: SDCC concepts enable network programmers to design simple abstractions for monitoring data traffic and update network policies [24]. Research efforts are required to develop policy based languages that makes it convenient for network administrators to manage network traffic functions in DCs.

VIII. CONCLUSION

Traditional DC architectures are rigid and complex, giving rise to vendor lock-in related problems for network infrastructures. Vendor- and hardware-specific restrictions are the long standing problems in conventional DCs. SDDCs address these issues by providing an open environment for users to manage data centers according to their requirements. The provision of global view and consistent policies in SDDCs make them the best option for users, service providers, administrators, and developers. Indeed, SDDCs are able to accommodate new and existing applications on multiple cloud platforms and their enhanced control over security and power usage levels. SDDCs are likely to reduce management costs as well, with no need of specific skills to operate network devices through vendor-specific interfaces. SDDCs are also expected to pave the way for development of new applications to fulfill user demands.

In spite of some recent interesting attempts that address SDCC features, the literature on the vast topic concerning their role in cloud was still limited. In this paper, we tried to fill this gap. The topics, the content, and the ideas presented in this paper will help in adopting a unified approach towards implementing SDCC concepts in future.

We began our paper by explaining basic SDCC architectural elements. We discussed its architectural challenges and limitations and surveyed various developments occurring in this domain. The main motivation to perform a comprehensive survey is to develop a consensus among the research community and to promote the idea of SDEs in cloud environments. A number of vendors are actively developing components and standards for adopting SDCC approach. Ultra large scale service providers (e.g. Google, Yahoo and Amazon) can potentially rip huge benefits from this.

SDCC concepts are promoting advancements in several areas including physical hardware and legacy infrastructures, network management and infrastructure bridging elements. We strongly believe that SDCC will continue to witness enormous growth in the near future and their adoption would add new levels of flexibility in cloud network programming and management.

REFERENCES

[1] R. Buyya, S. N. Srirama, G. Casale, R. Calheiros, Y. Simmhan, B. Varghese, E. Gelenbe, B. Javadi, L. M. Vaquero, M. A. Netto, and A. N. Toosi, "A manifesto for future generation cloud computing: Research directions for the next decade," *ACM Comput. Surv.*, vol. 51, no. 5, p. 105, 2018.

[2] F. Bannour, S. Souihi, and A. Mellouk, "Distributed SDN control: Survey, taxonomy, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 333–354, 1rd Quart., 2018.

[3] T. He, A. N. Toosi, and R. Buyya, "Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers," *J. Parallel Distrib. Comput.*, vol. 131, pp. 55–68, Sep. 2019.

[4] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Efficient NFV-enabled multicasting in SDNs," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2052–2070, Nov. 2019.

[5] M. Paliwal, D. Shrimankar, and O. Tembhurne, "Controllers in SDN: A review report," *IEEE Access*, vol. 6, pp. 36256–36270, 2018.

[6] Q. Qin, K. Poularakis, G. Iosifidis, and L. Tassiulas, "SDN controller placement at the edge: Optimizing delay and overheads," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 684–692.

[7] J. Xie, D. Guo, C. Qian, L. Liu, B. Ren, and H. Chen, "Validation of distributed SDN control plane under uncertain failures," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1234–1247, Jun. 2019.

[8] Y.-C. Wang and S.-Y. You, "An efficient route management framework for load balance and overhead reduction in SDN-based data center networks," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1422–1434, Sep. 2018.

[9] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3259–3306, 4th Quart., 2018.

[10] S.-Y. Chang, Y. Park, and B. B. A. Babu, "Fast IP Hopping Randomization to Secure Hop-by-Hop Access in SDN," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 308–320, Dec. 2019.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[12] B.-H. Oh, S. Vural, N. Wang, and R. Tafazolli, "Priority-based flow control for dynamic and reliable flow management in SDN," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1720–1732, Nov. 2018.

[13] D. Wu, X. Nie, E. Asmare, D. Arkhipov, Z. Qin, R. Li, J. McCann, and K. Li, "Towards distributed SDN: Mobility management and flow scheduling in software defined urban IOT," *IEEE Trans. Parallel Distrib. Syst.*, to be published.

[14] Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, "Green and sustainable cloud of things: Enabling collaborative edge computing," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 72–78, Jan. 2019.

[15] R. Chaudhary, G. S. Aujla, N. Kumar, and J. J. P. C. Rodrigues, "Optimized big data management across multi-cloud data centers: Software-defined-network-based analysis," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 118–126, Feb. 2018.

[16] A. N. Toosi, J. Son, and R. Buyya, "CLOUDS-Pi: A low-cost raspberry-pi based micro data center for software-defined cloud computing," *IEEE Cloud Comput.*, vol. 5, no. 5, pp. 81–91, Oct. 2018.

[17] J. Son and R. Buyya, "SDCon: Integrated control platform for software-defined clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 1, pp. 230–244, Jul. 2019.

[18] C.-T. Yang, S.-T. Chen, J.-C. Liu, Y.-Y. Yang, K. Mitra, and R. Ranjan, "Implementation of a real-time network traffic monitoring service with network functions virtualization," *Future Gener. Comput. Syst.*, vol. 93, pp. 687–701, Apr. 2019.

[19] B. Mao, Y. Yang, S. Wu, H. Jiang, and K.-C. Li, "IOFollow: Improving the performance of VM live storage migration with IO following in the cloud," *Future Gener. Comput. Syst.*, vol. 91, pp. 167–176, Feb. 2019.

[20] L. P. Priego, D. Osimo, and J. D. Wareham, "Data sharing practice in big data ecosystems," Escola Superior d'Administració i Direcció d'Empreses, Barcelona, Spain, Res. Paper 273, 2019.

[21] N. F. S. de Sousa, D. A. L. Perez, R. V. Rosa, M. A. S. Santos, and C. E. Rothenberg, "Network service orchestration: A survey," *Comput. Commun.*, to be published.

[22] F. A. Zaman and A. J. A. Karmouch, "Software defined network-based edge cloud resource allocation framework," *IEEE Access*, vol. 7, pp. 10672–10690, 2019.

[23] M. Banikazemi, D. Olshefski, A. Shaikh, J. Tracey, and G. Wang, "Meridian: An SDN platform for cloud network services," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 120–127, Feb. 2013.

[24] N. Foster, A. Guha, M. Reitblatt, A. Story, M. J. Freedman, N. P. Katta, C. Monsanto, J. Reich, J. Rexford, C. Schlesinger, D. Walker, and R. Harrison, "Languages for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 128–134, Feb. 2013.

[25] H. Parzyjegla, C. Wernecke, G. Mühl, E. Schweissguth, and D. Timmermann, "Implementing content-based publish/subscribe with OpenFlow," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 1392–1395.

[26] J. L. G. Gomez, T. C. Chang, C. F. Chou, L. Golubchik, "On improving the performance of software-defined networking through middlebox policies," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–4.

[27] R. M. Aileni, G. Suciu, C. M. Balaceanu, C. Beceanu, P. A. Lavinia, C.-V. Nadrag, S. Pasca, C. A. V. Sakuyama, and A. Vulpe, "Body area network (BAN) for healthcare by wireless mesh network (WMN)," in *Body Area Network Challenges and Solutions*. Cham, Switzerland: Springer, 2019, pp. 1–17.

[28] K. Košt'ál, R. Bencel, M. Ries, P. Trúchly, and I. Kotuliak, "High performance SDN WLAN architecture," *Sensors*, vol. 19, no. 8, p. 1880, 2019.

[29] N. S. Pawar, A. Arunvel, G. N. Kumar, and A. K. Sinha, "Securing network using software-defined networking in control and data planes," in *Computing and Network Sustainability*. Singapore: Springer, 2019, pp. 433–443.

[30] D. Singh, B. Ng, Y.-C. Lai, Y.-D. Lin, and W. K. G. Seah, "Analytical modelling of software and hardware switches with internal buffer in software-defined networks," *J. Netw. Comput. Appl.*, vol. 136, pp. 22–37, Jun. 2019.

[31] K. S. Sahoo, S. K. Panda, S. Sahoo, B. Sahoo, and R. Dash, "Toward secure software-defined networks against distributed denial of service attack," *J. Supercomput.*, to be published.

[32] HP. *HPE FlexFabric 12900E Switch Series*. Accessed: Jun. 17, 2019. [Online]. Available: http://h17007.www1.hp.com/docs/interop/2013/4AA4-6499ENW.pdf

[33] L. Liao, C.-F. Lai, J. Wan, V. C. M. Leung, and T.-C. Huang, "Scalable distributed control plane for On-line social networks support cognitive neural computing in software defined networks," *Future Gener. Comput. Syst.*, vol. 93, pp. 993–1001, Apr. 2019.

[34] Z. Liu, Y. Cao, and X. Zhang, "Managing recurrent virtual network updates in multi-tenant datacenters: A system perspective," *IEEE Trans. Parallel Distrib. Syst.*, to be published.

[35] C. Qiu, S. Cui, H. Yao, F. Xu, F. R. Yu, and C. Zhao, "A novel QoS-enabled load scheduling algorithm based on reinforcement learning in software-defined energy Internet," *Future Gener. Comput. Syst.*, vol. 92, pp. 43–51, Mar. 2019.

[36] K. Qiu, J. Yuan, J. Zhao, X. Wang, S. Secci, and X. Fu, "FastRule: Efficient flow entry updates for TCAM-based openflow switches," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 484–498, Mar. 2019.

[37] A. J. Ferrer, J. M. Marquéés, and J. Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Comput. Surv.*, vol. 51, no. 6, p. 111, 2019.

[38] W. Li, D. Li, Y. Bai, W. Le, and H. Li, "Memory-efficient recursive scheme for multi-field packet classification," *IET Commun.*, vol. 13, no. 9, pp. 1319–1325, 2019.

[39] Open Networking Foundation. (2014). *Migration Use Cases and Methods*. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/use-cases/Migration-WG-Use-Cases.pdf

[40] R. Ying, W.-K. Jia, Y. Zheng, and Y. Wu, "Fast invalid TCP flow removal scheme for improving SDN scalability," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–5.

[41] D. Cain, S. Racherla, P. Patil, A. M. Tarenzio, S. Irwin, and P. Ljungstrøm, *Implementing IBM Software Defined Network for Virtual Environments*. Armonk, NY, USA: IBM Redbooks, 2014.

[42] R. Gracia-Tinedo, J. Sampé, G. París, M. Sánchez-Artigas, P. García-López, and Y. Moatti, "Software-defined object storage in multi-tenant environments," *Future Gener. Comput. Syst.*, vol. 99, pp. 54–72, Oct. 2019.

[43] P. Raj and A. Raman, "Demystifying software-defined cloud environments," in *Software-Defined Cloud Centers*. Cham, Switzerland: Springer, 2018, pp. 13–34.

[44] D. Sanvito, I. Filippini, A. Capone, S. Paris, and J. Leguay, "Clustered robust routing for traffic engineering in software-defined networks," *Comput. Commun.*, vol. 144, pp. 175–187, Aug. 2019.

[45] K.-S. Lim, S. H. Lee, J. W. Han, and G. W. Kim, "Design considerations for an intelligent video surveillance system using cloud computing," in *Proc. Int. Conf. Parallel Distrib. Comput., Appl. Technol.* Singapore, Springer, 2018, pp. 84–89.

[46] U. Paščinski, J. Trnkoczy, V. Stankovski, M. Cigale, and S. Gec, "QoS-aware orchestration of network intensive software utilities within software defined data centres," *J. Grid Comput.*, vol. 16, no. 1, pp. 85–112, 2018.

[47] F. Kelbert and A. Pretschner, "Data usage control for distributed systems," *ACM Trans. Privacy Secur.*, vol. 21, no. 3, p. 12, 2018.

[48] J. Lango, "Toward software-defined SLAs," *Commun. ACM*, vol. 57, no. 1, pp. 54–60, 2014.

[49] A. Berns, J. Curbow, J. Hilliard, S. Jorkeh, and M. Sanders, "Lightweight formal methods for improving software security," Univ. Northern Iowa, Cedar Falls, IA, USA, Tech. Rep. 11, 2019.

[50] D. Yang, H. Wei, Y. Zhu, P. Li, and J.-C. Tan, "Virtual private cloud based power-dispatching automation system—Architecture and application," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1756–1766, Jun. 2019.

[51] M. H. Rehmani, A. Davy, B. Jennings, and C. Assi, "Software defined networks based smart grid communication: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, to be published.

[52] A. Demirpolat, D. Ergenç, E. Ozturk, Y. Ayar, and E. Onur, "Software-defined network security," in *Enabling Technologies and Architectures for Next-Generation Networking Capabilities*. Hershey, PA, USA: IGI Global, 2019, pp. 232–253.

[53] J. Griffioen, Z. Fei, S. Rivera, J. Chappell, M. Hayashida, P. Shi, C. Carpenter, Y. Song, B. Chitre, H. Nasir, and K. L. Calvert, "Leveraging SDN to enable short-term on-demand security exceptions," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 13–18.

[54] E. Barbierato, M. Gribaudo, M. Iacono, and A. Jakóbik, "Exploiting CloudSim in a multiformalism modeling approach for cloud based systems," *Simul. Model. Pract. Theory*, vol. 93, pp. 133–147, May 2019.

[55] Y. Roumani and J. K. Nwankpa, "An empirical study on predicting cloud incidents," *Int. J. Inf. Manage.*, vol. 47, pp. 131–139, Aug. 2019.

[56] B. Yin, L. Mei, Z. Jiang, and K. Wang, "Joint cloud collaboration mechanism between vehicle clouds based on blockchain," in *Proc. IEEE Int. Conf. Service-Oriented Syst. Eng. (SOSE)*, Apr. 2019, pp. 2227–2275.

[57] J. Duan, X. Yi, S. Zhao, C. Wu, H. Cui, and F. Le, "NFVactor: A resilient NFV system using the distributed actor model," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 586–599, Feb. 2019.

[58] H. Fontes, T. Cardoso, R. Campos, and M. Campos, "Improving ns-3 emulation performance for fast prototyping of routing and SDN protocols: Moving data plane operations to outside of ns-3," *Simul. Model. Pract. Theory*, vol. 96, Nov. 2019, Art. no. 101931.

[59] H. Cao, H. Zhu, and L. Yang, "Dynamic embedding and scheduling of service function chains for future SDN/NFV-enabled networks," *IEEE Access*, vol. 7, pp. 39721–39730, 2019.

[60] S. Jeong, J.-H. You, and J. W.-K. Hong, "Design and implementation of virtual TAP for SDN-based openstack networking," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 233–241.

[61] J. Chung, R. Kettimuthu, N. Pho, R. Clark, and H. Owen, "Orchestrating intercontinental advance reservations with software-defined exchanges," *Future Gener. Comput. Syst.*, vol. 95, pp. 534–547, Jun. 2019.
[62] A. Roozbeh, J. Soares, G. Q. Maguire, F. Wuhib, C. Padala, M. Mahloo, D. Turull, V. Yadhav, and D. Kostic, "Software-defined 'hardware' infrastructures: A survey on enabling technologies and open research directions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2454–2485, 3rd Quart., 2018.
[63] M. Usman, A. C. Risdianto, J. Han, and J. Kim, "Interactive visualization of SDN-enabled multisite cloud playgrounds leveraging smartx multi-view visibility framework," *Comput. J.*, vol. 62, no. 6, pp. 838–854, 2018.
[64] A. Khelaifa, S. Benharzallah, L. Kahloul, R. Euler, A. Laouid, and A. Bounceur, "A comparative analysis of adaptive consistency approaches in cloud storage," *J. Parallel Distrib. Comput.*, vol. 129, pp. 36–49, Jul. 2019.
[65] D. Garcia, J. Astorga, and E. Jacob, "Innovating at the connected industry: SDN and NFV experiences and lessons learned," in *Proc. IEEE 26th Int. Conf. Netw. Protocols (ICNP)*, Sep. 2018, pp. 245–246.
[66] Y. Gao, Y. Jing, and W. Dong, "UniROPE: Universal and robust packet trajectory tracing for software-defined networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2515–2527, Dec. 2018.
[67] R. Shrivastava, K. Samdanis, and V. Sciancalepore, "Towards service-oriented soft spectrum slicing for 5G TDD networks," *J. Netw. Comput. Appl.*, vol. 137, pp. 78–90, Jul. 2019.
[68] K.-T. Foerster, M. Pacut, and S. Schmid, "On the complexity of non-segregated routing in reconfigurable data center architectures," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 49, no. 2, pp. 2–8, 2019.
[69] E. Shao, Z. Wang, G. Yuan, G. Tan, and N. Sun, "Wormhole optical network: A new architecture to solve long diameter problem in exascale computer," *CCF Trans. High Perform. Comput.*, to be published.
[70] R. Wu, L. Huang, and H. Zhou, "RHKV: An RDMA and HTM friendly key–value store for data-intensive computing," *Future Gener. Comput. Syst.*, vol. 92, pp. 162–177, Mar. 2019.
[71] C. Calcaterra, A. Carmenini, A. Marotta, and D. Cassioli, "Hadoop performance evaluation in software defined data center networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2019, pp. 56–61.
[72] G. Kaur, A. Bala, and I. Chana, "An intelligent regressive ensemble approach for predicting resource usage in cloud computing," *J. Parallel Distrib. Comput.*, vol. 123, pp. 1–12, Jan. 2019.
[73] G. Cao, "Topology-aware multi-objective virtual machine dynamic consolidation for cloud datacenter," *Sustain. Comput. Inform. Syst.*, vol. 21, pp. 179–188, Mar. 2019.
[74] A. Ponraj, "Optimistic virtual machine placement in cloud data centers using queuing approach," *Future Gener. Comput. Syst.*, vol. 93, pp. 338–344, Apr. 2019.
[75] H. Hawilo, M. Jammal, and A. Shami, "Network function virtualization-aware orchestrator for service function chaining placement in the cloud," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 643–655, Mar. 2019.
[76] J. Zander and R. Forchheimer, "The SOFTNET project: A retrospect," in *Proc. IEEE EUROCON*, Jun. 1988, pp. 343–345.
[77] A. Doria, *General Switch Management Protocol (GSMP) V3*, document RFC 3292, 2002.
[78] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommun. Syst.*, vol. 70, no. 3, pp. 447–489, 2019.
[79] M. Rezaee and M. H. Y. Moghaddam, "SDN-based quality of service networking for wide area measurement system," *IEEE Trans. Ind. Informat.*, to be published.
[80] K. Poularakis, Q. Qin, E. M. Nahum, M. Rio, and L. Tassiulas,"Flexible SDN control in tactical ad hoc networks," *Ad Hoc Netw.*, vol. 85, pp. 71–80, Mar. 2019.
[81] R. Enns, *Network Configuration Protocol (NETCONF)*, document RFC 6241, 2011.
[82] A. Kannan, S. Vijayan, M. Narayanan, and M. Reddiar, "Adaptive routing mechanism in SDN to limit congestion," *Information Systems Design and Intelligent Applications*. Singapore: Springer, 2019, pp. 245–253.
[83] W. Liu, Y. Wang, J. Zhang, H. Liao, Z. Liang, and X. Liu, "AAMcon: An adaptively distributed SDN controller in data center networks," *Frontiers Comput. Sci.*, to be published.
[84] M. Cokic and I. Seskar, "Software defined network management for dynamic smart GRID traffic," *Future Gener. Comput. Syst.*, vol. 96, pp. 270–282, Jul. 2019.
[85] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-Peer Netw. Appl.*, vol. 12, no. 2, pp. 493–501, 2019.
[86] J. C. Nobre, A. M. de Souza, D. Rosario, C. Both, L. A. Villas, E. Cerqueira, T. Braun, and M. Gerla, "Vehicular software-defined networking and fog computing: Integration and design principles," *Ad Hoc Netw.*, vol. 82, pp. 172–181, Jan. 2019.
[87] A. Nayyer, A. K. Sharma, and L. K. Awasthi, "Laman: A supervisor controller based scalable framework for software defined networks," *Comput. Netw.*, vol. 159, pp. 125–134, Aug. 2019.
[88] S. Chaipet and W. Putthividhya, "On studying of scalability in single-controller software-defined networks," in *Proc. 11th Int. Conf. Knowl. Smart Technol. (KST)*, Jan. 2019, pp. 158–163.
[89] H. Yu, H. Qi, and K. Li, "WECAN: An efficient west-east control associated network for large-scale SDN systems," *Mobile Netw. Appl.*, to be published. [Online]. Available: http://www.projectfloodlight.org/floodlight/
[90] *Floodlight Controller*. Accessed: Jun. 17, 2019. [Online]. Available: http://www.projectfloodlight.org/floodlight/
[91] F. Chahlaoui and H. Dahmouni, "Towards QoS-enabled SDN networks," in *Proc. Int. Conf. Adv. Commun. Technol. Netw. (CommNet)*, Apr. 2018, pp. 1–7.
[92] C.-C. Chuang, Y.-J. Yu, and A.-C. Pang, "Flow-aware routing and forwarding for SDN scalability in wireless data centers," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1676–1691, Dec. 2018.
[93] *OpenMUL Controller*. Accessed: Jun. 17, 2019. [Online]. Available: http://www.openmul.org/openmul-controller.html
[94] Y. Shi, Y. Cao, J. Liu, and N. Kato, "A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks," *IEEE Netw.*, vol. 33, no. 1, pp. 29–35, Feb. 2018.
[95] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, "Multi-controller based software-defined networking: A survey," *IEEE Access*, vol. 6, pp. 15980–15996, 2018.
[96] Y.-D. Lin, C.-C. Wang, C.-Y. Huang, and Y.-C. Lai, "Hierarchical CORD for NFV datacenters: Resource allocation with cost-latency tradeoff," *IEEE Netw.*, vol. 32, no. 5, pp. 124–130, Apr. 2018.
[97] M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Comput. Surv.*, vol. 51, no. 6, p. 114, 2019.
[98] J. Mccauley. *POX: A Python-based OpenFlow Controller*. Accessed: Jun. 17, 2019. [Online]. Available: https://github.com/noxrepo/pox
[99] J. Ali, S. Lee, and B.-H. Roh, "Performance analysis of POX and Ryu with different SDN topologies," in *Proc. Int. Conf. Inf. Sci. Syst.*, 2018, pp. 244–249.
[100] Y. Li, Z. Wang, J. Yao, X. Yin, X. Shi, J. Wu, and H. Zhang, "MSAID: Automated detection of interference in multiple SDN applications," *Comput. Netw.*, vol. 153, pp. 49–62, Apr. 2019.
[101] *Trema Openflow Controller Framework*. [Online]. Available: https://github.com/trema/trema
[102] Y. Xu, M. Cello, I.-C. Wang, A. Walid, G. Wilfong, C. H.-P. Wen, M. Marchese, and H. Jonathan Chao. "Dynamic switch migration in distributed software-defined networks to achieve controller load balance," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 515–529, Feb. 2019.
[103] *OpenDaylight Controller*. [Online]. Available: https://www.opendaylight.org/
[104] V. Jain, V. Yatri, and C. Kapoor, "Software defined networking: State-of-the-art," *J. High Speed Netw.*, vol. 25, no. 1, pp. 1–40, 2019.
[105] D. Oliveira, N. Ghani, M. Hayat, J. Crichigno, and E. Bou-Harb, "SDN testbed for evaluation of large exo-atmospheric EMP attacks," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 88–97, Dec. 2019.
[106] C. M. Zwölf, N. Moreau, Y.-A. Ba, and M.-L. Dubernet, "Implementing in the VAMDC the new paradigms for data citation from the research data alliance," *Data Sci. J.*, vol. 18, no. 1, pp. 1–13, 2019.
[107] *Apache CloudStack: Open Source Cloud Computing*. [Online]. Available: https://cloudstack.apache.org/
[108] N. Joshi and S. Shah, "A comprehensive survey of services provided by prevalent cloud computing environments," in *Smart Intelligent Computing and Applications*. Singapore: Springer, 2019, pp. 413–424.
[109] F. Zalila, S. Challita, and P. Merle, "Model-driven cloud resource management with OCCIware," *Future Gener. Comput. Syst.*, vol. 99, pp. 260–277, Oct. 2019.
[110] *Virtualized Services Platform (VSP)*. [Online]. Available: http://www.nuagenetworks.net/products/#vsp
[111] *Eucalyptus Datasheet*. [Online]. Available: https://www.eucalyptus.com

[112] R. Montero, F. Agraz, A. Pagès, J. Perelló, and S. Spadaro, "SDN-based parallel link discovery in optical transport networks," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 1, p. e3512, 2019.
[113] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 566–578, Mar. 2019.
[114] T. Liu, J. C. S. Lui, D. Lin, and D. Hui, "On the feasibility of inter-domain routing via a small broker set," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 2, pp. 415–427, Aug. 2019.
[115] C. Birkinshaw, E. Rouka, and V. G. Vassilakis, "Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks," *J. Netw. Comput. Appl.*, vol. 136, pp. 71–85, Jun. 2019.
[116] T. UcedaVelez and M. M. Morana. *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. Hoboken, NJ, USA: Wiley, 2015.
[117] D.-Y. Lee, C.-C. Wang, and A.-Y. Wu, "Bundle-updatable SRAM-based TCAM design for openflow-compliant packet processor," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 6, pp. 1450–1454, Jan. 2019.
[118] D. Dumitrescu, R. Stoenescu, M. Popovici, L. Negreanu, and C. Raiciu, "Dataplane equivalence and its applications," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement. NSDI*, 2019, pp. 683–698.
[119] Z. Hu, B. Li, Q. Zheng, and R. S. M. Goh, "Low latency big data processing without prior information," *IEEE Trans. Cloud Comput.*, to be published.
[120] A. Aydeger, N. Saputro, K. Akkaya, and S. Uluagac, "SDN-enabled recovery for Smart Grid teleprotection applications in post-disaster scenarios," *J. Netw. Comput. Appl.*, vol. 138, pp. 39–50, Jul. 2019.
[121] C. Zhang, H. Yang, and G. F. Riley, "Admission control in software-defined datacenter network in view of flow table capacity," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 871–876.
[122] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 812–837, 1st Quart., 2019.
[123] A. N. Al-Quzweeni, A. Q. Lawey, T. E. H. Elgorashi, and J. M. H. Elmirghani, "Optimized energy aware 5G network function virtualization," *IEEE Access*, vol. 7, pp. 44939–44958, 2019.
[124] P. Peresini, M. Kuzniar, and D. Kostic, "Dynamic, fine-grained data plane monitoring with monocle," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 534–547, Feb. 2018.
[125] K. Figiela, A. Gajek, A. Zima, B. Obrok, and M. Malawski, "Performance evaluation of heterogeneous cloud functions," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 23, p. e4792, 2018.
[126] Y. Chen, H. Zheng, and J. Wu, "Consistency, feasibility, and optimality of network update in SDNs," *IEEE Trans. Netw. Sci. Eng.*, to be published.
[127] Q. T. Minh, T. K. Dang, T. Nam, and T. Kitahara, "Flow aggregation for SDN-based delay-insensitive traffic control in mobile core networks," *IET Commun.*, vol. 13, no. 8, pp. 1051–1060, 2019.
[128] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, Feb. 2013.
[129] H. Jin, A. A. Abbasi, and S. Wu, "Pathfinder: Application-aware distributed path computation in clouds," *Int. J. Parallel Program.*, vol. 45, no. 6, pp. 1273–1284, 2017.
[130] S.-H. Tseng, A. Tang, G. L. Choudhury, and S. Tse, "Routing stability in hybrid software-defined networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 790–804, Apr. 2019.
[131] X. Wang, M. Malboubi, Z. Pan, J. Ren, S. Wang, S. Xu, and C.-N. Chuah, "ProgLIMI: Programmable link metric identification in software-defined networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2376–2389, Oct. 2018.
[132] J. Badshah, M. Kamran, N. Shah, and S. A. Abid, "An improved method to deploy cache servers in software defined network-based information centric networking for big data," *J. Grid Comput.*, vol. 17, no. 2, pp. 255–277, 2019.
[133] *Connected Cloud Control: Openflow in Mirage*. [Online]. Available: http://www.openmirage.org/blog/announcing-mirage-openflow
[134] W. Kim, J. W.-K. Hong, and Y.-J. Suh, "T-DCORAL: A threshold-based dynamic controller resource allocation for elastic control plane in software-defined data center networks," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 198–201, Nov. 2019.
[135] V. Kaushik, A. Sharma, and R. Tomar, "Virtualizing network functions in software-defined networks," in *Innovations in Software-Defined Networking and Network Functions Virtualization*. Hershey, PA, USA: IGI Global, 2018, pp. 26–51.
[136] P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, and C.-S. Yang, "Network monitoring in software-defined networking: A review," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3958–3969, Feb. 2018.
[137] D. Basu, X. Wang, Y. Hong, H. Chen, and S. Bressan "Learn-as-you-go with megh: Efficient live migration of virtual machines," *IEEE Trans. Parallel Distrib. Syst.*, to be published.
[138] V. Persico, A. Botta, A. Montieri, and A. Pescapé, "A first look at public-cloud inter-datacenter network performance," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.
[139] V. Persico, A. Botta, P. Marchetta, A. Montieri, and A. Pescape, "On the performance of the wide-area networks interconnecting public-cloud datacenters around the globe," *Comput. Netw.*, vol. 112, pp. 67–83, Jan. 2017.
[140] P. Megyesi, A. Botta, G. Aceto, A. Pescapé, and S. Molnár, "Challenges and solution for measuring available bandwidth in software defined networks," *Comput. Commun.*, vol. 99, pp. 48–61, Feb. 2017.
[141] M. Aldossary, K. Djemame, I. Alzamil, A. Kostopoulos, A. Dimakis, and E. Agiatzidou, "Energy-aware cost prediction and pricing of virtual machines in cloud computing environments," *Future Gener. Comput. Syst.*, vol. 93, pp. 442–459, Apr. 2019.
[142] S.-H. Shen, "An efficient network monitor for SDN networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 2, pp. 95–96, 2019.
[143] V. Persico, P. Marchetta, A. Botta, and A. Pescapè, "Measuring network throughput in the cloud: The case of Amazon EC2," *Comput. Netw.*, vol. 93, pp. 408–422, Dec. 2015.
[144] V. Persico, A. Montieri, and A. Pescapé, "On the network performance of amazon S3 cloud-storage service," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Oct. 2016, pp. 113–118.
[145] V. Persico, D. Grimaldi, A. Pescape, A. Salvi, and S. Santini, "A fuzzy approach based on heterogeneous metrics for scaling out public clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 8, pp. 2117–2130, Jan. 2017.
[146] A. Botta, W. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Generat. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.
[147] Y. Jararweh, M. Al-Ayyoub, E. Benkhelifa, M. Vouk, and A. Rindos, "Software defined cloud: Survey, system and evaluation," *Future Gener. Comput. Syst.*, vol. 58, pp. 56–74, May 2016.
[148] G. Xu, B. Dai, B. Huang, J. Yang, and S. Wen, "Bandwidth-aware energy efficient flow scheduling with SDN in data center networks," *Future Gener. Comput. Syst.*, vol. 68, pp. 163–174, Mar. 2017.

AAQIF AFZAAL ABBASI received the Ph.D. degree in computer engineering from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He is currently an Assistant Professor with the Department of Software Engineering, Foundation University, Islamabad, Pakistan. His current research interests include parallel and distributed systems, cloud computing, data-intensive computing, and data center performance optimization. He is a member of the ACM.

ALMAS ABBASI received the Ph.D. degree in computer science from the School of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia. She is currently an Assistant Professor with the Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan. Her current research interests include image processing, information hiding, and mobile applications.

SHAHABODDIN SHAMSHIRBAND received the M.Sc. degree in artificial intelligence from Iran, and the Ph.D. degree in computer science from the University of Malaya (UM), Malaysia, in 2014. He was an Adjunct Assistant Professor with the Department of Computer Science, Iran University of Science and Technology. He also severed as a Senior Lecturer with UM, Malaysia, and with Islamic Azad University, Iran. He participated in many research programs within the Center of Big Data Analysis, IUST and IAU. He has been associated with young researchers and elite club, since 2009. He supervised or co-supervised undergraduate and postgraduate students (master's and Ph.D.) by research and training. He has also authored or coauthored papers published in IF journals and attended to high-rank A and B conferences. He is an Associate Editor, a Guest Editor, and a Reviewer of high-quality journals and conferences. He is a professional member of the ACM.

ANTHONY THEODORE CHRONOPOULOS received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, in 1987. He is currently a Full Professor with the Department of Computer Science, University of Texas, San Antonio, TX, USA, and a Visiting Professor with the Department of Computer Engineering and Informatics, University of Patras, Rio, Greece. He has coauthored 82 journals and 73 peer-reviewed conference proceedings publications in the areas of distributed and parallel computing, grid and cloud computing, scientific computing, and computer networks. He is a Fellow of the Institution of Engineering and Technology (FIET) and a Senior Member of the ACM.

VALERIO PERSICO received the Ph.D. degree in computer and automation engineering from the University of Napoli Federico II, in 2016, where he is currently an Assistant Professor with DIETI. His work focuses on Internet measurements, cloud network performance, and traffic analysis and classification. He has coauthored more than 30 conference and journal papers. He was a recipient of the Best Journal Paper Award of the IEEE ComSoc Technical Committee on Communications Systems Integration and Modeling (CSIM) 2018 and the Best Student Paper Award at the ACM CoNext 2013.

ANTONIO PESCAPÈ is currently a Full Professor of computer engineering with the University of Napoli Federico II. He has coauthored more than 200 conference and journal papers. His works focus on Internet technologies and specifically on measurement, monitoring, and analysis of the Internet. He was a recipient of a number of research awards.