Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges

Qiao Yan, F. Richard Yu, *Senior Member, IEEE*, Qingxiang Gong, and Jianqiang Li

***Abstract*—Distributed denial of service (DDoS) attacks in cloud computing environments are growing due to the essential characteristics of cloud computing. With recent advances in software-defined networking (SDN), SDN-based cloud brings us new chances to defeat DDoS attacks in cloud computing environments. Nevertheless, there is a contradictory relationship between SDN and DDoS attacks. On one hand, the capabilities of SDN, including software-based traffic analysis, centralized control, global view of the network, dynamic updating of forwarding rules, make it easier to detect and react to DDoS attacks. On the other hand, the security of SDN itself remains to be addressed, and potential DDoS vulnerabilities exist across SDN platforms. In this paper, we discuss the new trends and characteristics of DDoS attacks in cloud computing, and provide a comprehensive survey of defense mechanisms against DDoS attacks using SDN. In addition, we review the studies about launching DDoS attacks on SDN, as well as the methods against DDoS attacks in SDN. To the best of our knowledge, the contradictory relationship between SDN and DDoS attacks has not been well addressed in previous works. This work can help to understand how to make full use of SDN's advantages to defeat DDoS attacks in cloud computing environments and how to prevent SDN itself from becoming a victim of DDoS attacks, which are important for the smooth evolution of SDN-based cloud without the distraction of DDoS attacks.**

***Index Terms*—Software-defined networking (SDN), distributed denial of service attacks (DDoS), cloud computing.**

I. Introduction

CLOUD computing develops rapidly in both academia and industry due to its essential characteristics, including on-demand self-service, broadband network access, resource pooling, rapid elasticity, and measured service. It has significant advantages over traditional computing paradigms, such as reducing capital expenditure (CapEx) and operational expenditure (OpEx) [1]–[3].

Manuscript received September 9, 2014; revised March 16, 2015 and August 17, 2015; accepted September 29, 2015. Date of publication October 5, 2015; date of current version January 27, 2016. This work is jointly supported by the National Science Foundation of China (Grant Nos. 61572330 and 61170283), the Technology Planning Project of Guangdong Province, China (Grant No. 2014B010118005), and the Natural Sciences and Engineering Research Council of Canada. *(Corresponding author: Jianqiang Li.)*

Q. Yan, Q. Gong, and J. Li are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: yanq@szu.edu.cn; gongqingxiang@email.szu.edu.cn; lijq@szu.edu.cn).

F. R. Yu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: richard.yu@carleton.ca).

Digital Object Identifier 10.1109/COMST.2015.2487361

Cloud computing would not be possible without the underneath support of networking [4]–[7]. Recently, *software-defined networking* (SDN) has attracted great interests as a new paradigm in networking [8]–[10]. In SDN, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications [11]. Integration of these two promising technologies, cloud computing and SDN, can greatly improve cloud manageability, scalability, controllability and dynamism [4]. SDN-based cloud is a new type cloud in which SDN technology is used to acquire control on network infrastructure and to provide networking-as-a-service (NaaS) in cloud computing environments [12].

Security has been regarded as the dominate barrier of the development of cloud computing [13]. Good features of SDN offer new opportunities to defeat attacks in cloud computing environments [9], [10]. Among the security requirements of cloud computing, availability is crucial since the core function of cloud computing is to provide on-demand services of different levels [13]. *Denial of Service* (DoS) attacks and *Distributed Denial of Service*(DDoS) flooding attacks are the main methods to destroy availability of cloud computing [13]–[15]. DoS attacks or DDoS attacks are an attempt to make a machine or network resource unavailable to its intended users. DDoS attacks are sent by two or more persons, or bots [16], while DoS attacks are sent by one person or system. A bot is a compromised device created when a computer is penetrated by software from a malware code. Since DoS attacks can be seen as a special type of DDoS attacks, in the rest of this paper, we just use the term DDoS attacks to indicate both DoS and DDoS attacks.

Although the capabilities of SDN (e.g., software-based traffic analysis, logical centralized control, global view of the network, and dynamic updating of forwarding rules) make it easy to detect and to react DDoS attacks in cloud environments, the separation of the control plane from the data plane in SDN introduces new attack planes. SDN itself may be a target of some attacks, and potential DDoS vulnerabilities exist across SDN platforms. For example, an attacker can take advantages of the characteristics of SDN to launch DDoS attacks against the control layer, infrastructure layer plane and application layer of SDN [11].

In this paper, we discuss the new trends and characteristics of DDoS attacks in cloud computing, and provide a comprehensive survey of defense mechanisms against DDoS attacks using

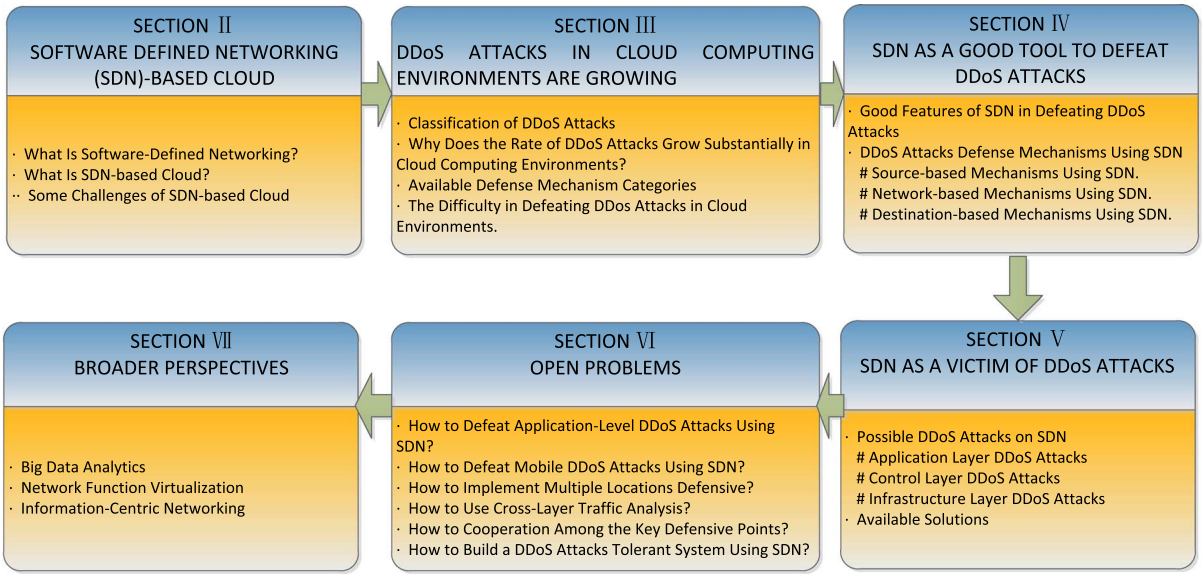

Fig. 1. The road map of this paper.

SDN. In addition, we review the studies about launching DDoS attacks on the control layer, infrastructure layer and application layer of SDN, as well as the methods against DDoS attacks in SDN.

To the best of our knowledge, the contradictory relationship between SDN and DDoS attacks has not been well addressed in previous works. Essentially, it is the unique dynamics tied with SDN and DDoS attacks that present unique challenges beyond the existing works. We believe that these initial steps we have taken here help understand how to make full use of SDN's advantages to defeat DDoS attacks in cloud environments and how to prevent SDN itself from becoming a victim of DDoS attacks, which are important for the smooth evolution of SDN-based cloud without the distraction of DDoS attacks.

A road map of our approach is given in Fig. 1. In Section II, we first introduce the concept of SDN, followed by SDN-based cloud. Then, we present some challenges of SDN-based cloud. In Section III, we give the classification of DDoS attacks. Then we discuss the new trends of DDoS in cloud computing environments based on the essential characteristics of cloud computing. We overview available defense mechanisms. Finally we discuss the difficulty in defeating DDoS attacks in cloud computing environments. In Section IV, we summarize the good features of SDN that bring a lot of benefits for defeating DDoS attacks, and provide a comprehensive survey on some of the works that have already been done to defend DDoS attacks using SDN. Section V presents the works about launching DDoS attacks on the planes of SDN and how to deal with this problem. Section VI discusses some open research issues. Some broader perspectives are presented in Section VII. Finally, we conclude this study in Section VIII.

II. Software-Defined Networking (SDN)-Based Cloud

Cloud computing offers an effective way to reduce capital expenditure (CapEx) and operational expenditure (OpEx) [13]. To reach this target, an agile and programmable network infrastructure is needed. SDN is the key technology that takes network control into the cloud [4].

In this section, we first introduce the concept of SDN, followed by SDN-based cloud. Then, we present some challenges of SDN-based cloud.

A. What is Software-Defined Networking

SDN is currently attracting significant attention from both academia and industry. The Open Networking Foundation (ONF) [17] is a nonprofit consortium dedicated to development, standardization, and commercialization of SDN. ONF has provided the most explicit and well received definition of SDN as follows: "In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications" [11].

ONF presents a high-level architecture for SDN that is vertically split into three main functional layers including infrastructure layer, control layer and application layer [9], [11], [18]–[20], as shown in Fig. 2.

1) Infrastructure Layer: Also known as the data plane, it consists mainly of Forwarding Elements (FEs) including physical switches, such as Juniper Junos MX-series, and virtual switches, such as Open vSwich. These switches are accessible via an open interface to switch and forward packets.
2) Control Layer: Also known as the control plane, it consists of a set of software-based SDN controllers providing a consolidated control functionality through open APIs to supervise the network forwarding behavior through an open interface. Three communication interfaces allow the controllers to interact: southbound, northbound and east/westbound interfaces [9].
3) Application Layer: It mainly consists of the end-user business applications that consume the SDN communications

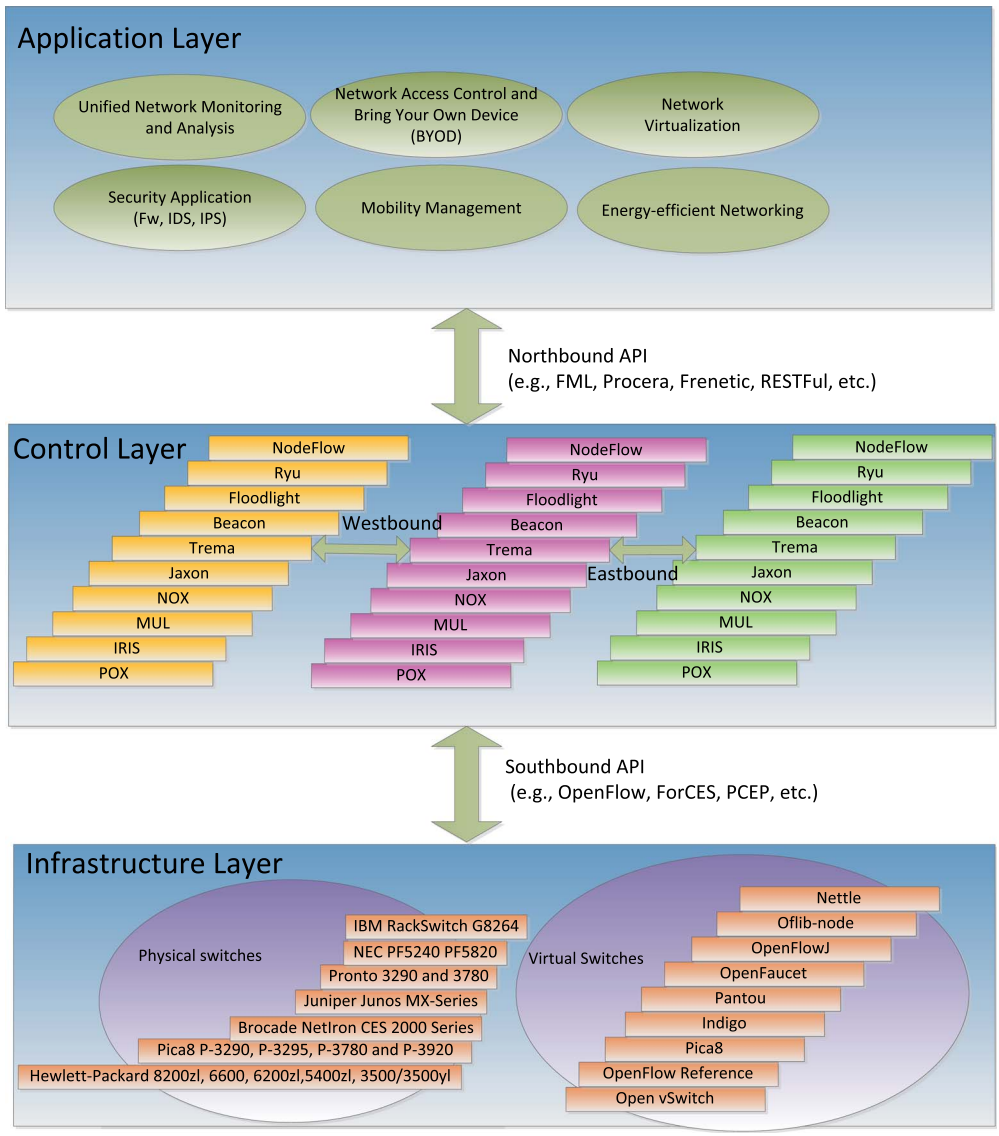

Fig. 2. High-level overview of the SDN architecture.

and network services [9]. Examples of such business applications include network virtualization, mobility management, security application and so on.

SDN is very often linked to the OpenFlow protocol. OpenFlow protocol is one element of the SDN architecture, which allows switches to perform flows-level control. It is an open protocol that was born in academia at Stanford University. It was proposed to standardize the communication between the switches and the software-based controller in an SDN architecture and to enable researchers to run experimental protocols in networks [9], [21]. Today cloud computing enables the networked computation and storage without using local resources. Such a decoupling of control and data plays a critical role in large-scale, high-speed computing systems [8].

B. What is SDN-Based Cloud?

Combining cloud technique and SDN paradigm provides a new opportunity to closely integrate application provisioning in the cloud with the network through programmable interfaces and automation [22]. With cloud development, the existing cloud networks face some significant challenges, including guaranteed performance of applications when applications are moved from on-premises to the cloud facility, flexible deployment of appliances (e.g., intrusion detection systems or firewalls), associated complexities to the policy enforcement and topology dependence, and the security and privacy protection [12]. More programmable, more flexible and more secure cloud infrastructure is needed. As a new networking paradigm, SDN is the key technology that can improve cloud manageability, scalability, controllability and dynamism. SDN can provide a new, dynamic network architecture that transforms traditional cloud network backbones into rich service-delivery platforms [12].

It is because of these reasons SDN-based cloud has been introduced recently [4], [12], [22]–[29]. SDN-based cloud is a new type cloud, in which SDN technology is used to acquire control on network infrastructure and to provide networking-as-a-service (NaaS). In SDN-based cloud, cloud computing extends from server centralization and virtualization as well as

storage centralization and virtualization to network centralization and virtualization.

SDN-based cloud has attracted great attentions recently. The authors of [22] argue that service-level network model that provides higher-level connectivity and policy abstractions are integral parts of cloud applications. So they introduce an SDN-based framework named Meridian, which supports a service-level model for application networking and can exploit multiple options for implementing virtual networks on the underlying physical network. Yen *et al.* [23] establish an SDN-based cloud computing environment via open source OpenFLow switch and controller packages. Then they extend the functionality of OpenFLow controller to provide load balancing, power-saving, and monitoring mechanisms.

The authors of [24] propose several designs for SDN-based mobile cloud architectures, focusing on ad hoc networks. They also introduce several instances of the proposed architectures based on frequency selection of wireless transmission that are designed around different use cases of SDN-based mobile cloud. Gharakheili *et al.* [25] develop an architecture, comprising a cloud-based front-end user interface and SDN-based APIs in the back-end, by which an ISP can allow users to customize their home network experience.

SDN-based cloud shows many advantages comparing with traditional cloud in terms of quality of service (QoS), VM orchestration, security, and so on. A QoS-guaranteed approach is proposed in [26] for bandwidth allocation that satisfies QoS requirements for all priority cloud users by using Open vSwitch based on SDN. The authors of [27] present an SDN-based orchestration framework for live VM management where server hypervisors exploit temporal network information to migrate VMs and minimize the network-wide communication cost of the resulting traffic dynamics. An unified solution is presented in [28] to combine two strategies, flow migration and VM migration, to maximize throughput and minimize energy simultaneously. The authors of [29] believe SDN offers new opportunities for network security in cloud scenarios, because SDN-based cloud provides more flexibility and faster reaction when the conditions are changing.

C. Some Challenges of SDN-Based Cloud

Although SDN-based cloud shows many good features, it faces several challenges that must be taken into consideration, including performance, availability, scalability **and** security.

- Performance:

  Performance refers to the processing speed of the network node considering both throughput and latency [11]. The method of SDN to handle new packets brings the programmability. But at the same time it produces performance problems. The authors of [30] show that current controllers cannot handle a big number of flows in 10 Gbps links. So how to improve performance and keep programmability need further research.
- Availability:

  Availability refers to the proportion of time an SDN system is in a functioning condition. The dependence on the controller brings a challenge regarding availability. One advantage of a traditional, distributed network architecture is that if a switch fails, the availability of the network can be maintained [31]. But in a pure SDN environment, if a controller fails, the availability of the network may be complete loss.
- Scalability:

  Scalability is the ability to be enlarged to accommodate network growth. The controller can become a bottleneck of scalability. By introducing distributed or peer-to-peer controller infrastructure may share the communication burden of the controller [11]. But an overall network view is required to direct the communications between the controllers using the east and westbound APIs [11]. Besides controller scalability, there are some other scalability concerns including the flow setup overhead and resilience to failures [32].
- Security:

  By decoupling the control plane from the data plane, the attack surface for SDN is augmented, when compared to traditional networks [9]. Security analysis has showed that the SDN framework suffers many security threats, including [33]:

  1) unauthorized access, e.g., unauthorized controller access or unauthenticated application access,
  2) data leakage, e.g., flow rule discovery (side channel attack on input buffer) forwarding policy discovery (packet processing timing analysis),
  3) data modification, e.g., flow rule modification to modify packets,
  4) malicious applications, e.g., fraudulent rule insertion controller hijacking,
  5) configuration issues, e.g., lack of TLS (or other authentication techniques) adoption policy enforcement, and
  6) denial of service, e.g., controller-switch communication flood switch flow table flooding.

  Seven main potential threat vectors in SDN are as follows [34]:
  1) forged or faked traffic flows,
  2) attacks on vulnerabilities in switches,
  3) attacks on control plane communications,
  4) attacks on and vulnerabilities in controllers,
  5) lack of mechanisms to ensure trust between the controller and management applications,
  6) attacks on and vulnerabilities in administrative stations, and
  7) lack of trusted resources for forensics and remediation.

  Among the well-known vulnerabilities of the SDN platform, DDoS attacks can have a devastating impact on the whole network. We will discuss this issue in details in Section V.

III. DDoS Attacks in Cloud Computing Environments are Growing

In this section, we briefly describe the classification of DDoS attacks. After that, we discuss the reasons why DDoS attacks

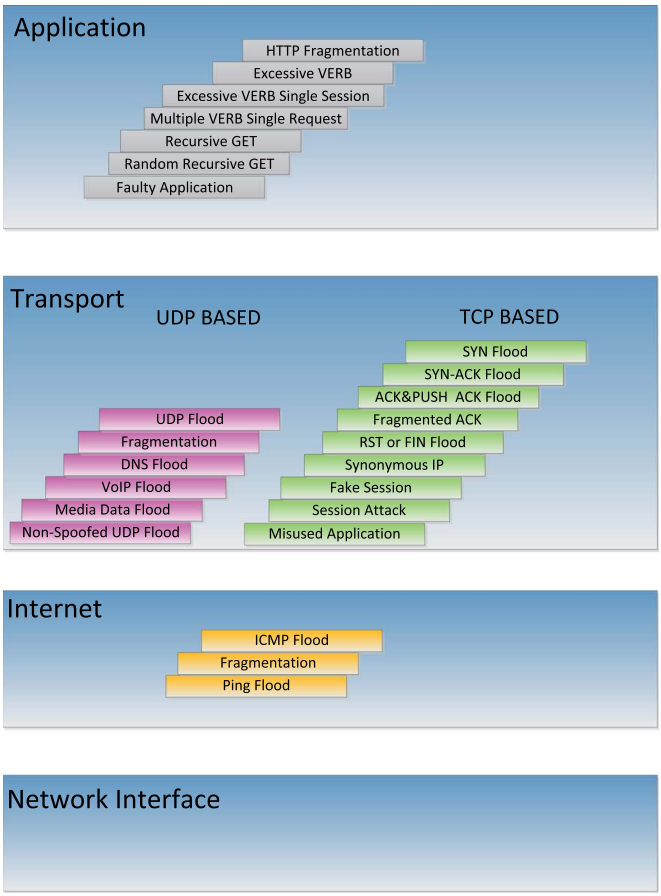

Fig. 3. Taxonomy of DDoS attacks.

are growing in cloud computing environments. Then we review various available defense mechanisms against DDoS attacks.

A. Classification of DDoS Attacks

DDoS attacks are easy to launch, but difficult to guard against. In order to launch an effective DDoS attack, cyber attackers often establish a network of computers, which is known as a botnet.

DDoS attacks can be classified into two categories based on the targeted protocol level [35]:

1) Network/transport-level DDoS flooding attacks: These attacks have been mostly launched using TCP, UDP, ICMP and DNS protocol packets and focus on disrupting legitimate user's connectivity by exhausting victim network's bandwidth [35].
2) Application-level DDoS flooding attacks: These attacks focus on disrupting legitimate users' services by exhausting the server resources (e.g., Sockets, CPU, memory, disk/database bandwidth, and I/O bandwidth) [36].

More detailed taxonomy of DDoS attacks is shown in Fig. 3 based on TCP/IP protocols [37].

B. Why Does the Rate of DDoS Attacks Grow Substantially in Cloud Computing Environments

With the network migrating to cloud computing environments, the rate of DDoS attacks is growing substantially.

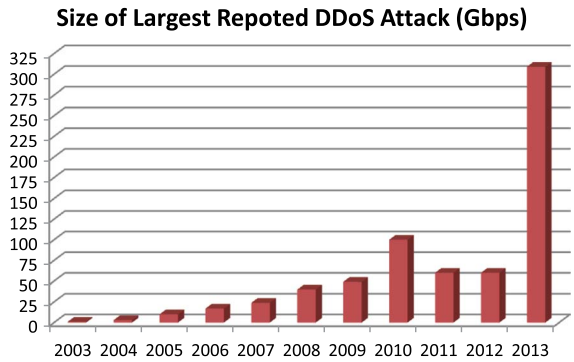

Fig. 4. The size of largest reported DDoS attack from 2003 to 2013 [42].

Traditional DDoS attacks defense mechanisms face many challenges in cloud computing environments. A recent Cloud Security Alliance (CSA) survey shows DDoS attacks are critical threats to cloud security [14], [38]. According to the quarterly State of the Internet Report (SOTI) from Akamai Technologies [39], DDoS attacks in the fourth quarter of 2012 were up by 200 percent over 2011.

In this subsection we discuss the reasons why the rate of DDoS attacks grow substantially in cloud computing environments, by reviewing the essential characteristics of cloud computing, including on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service.

1) On-Demand Self-Service Leading to Botnets Outbreak: One major reason is the emergence and development of botnets. Botnets are networks that are formed by bots or machines compromised by malware. Large-scale botnets (e.g., Srizbi, Kraken/Bobax, and Rustock) gained notoriety for their sizes and malicious activities (e.g., performing DDoS attacks [16]).

It remains fairly complex infecting a sufficient number of machines in a short time frame in traditional networks. But on-demand self-service capabilities of cloud that let legitimate businesses quickly add or subtract computing power could be used to instantly create a powerful botnet [40].

With cloud computing development, malware-as-a-service operations have started to take off since 2006. Malware-as-a-service is used for spamming and launching denial-of-service attacks. Because of competition among suppliers, prices of malware-as-a-service have been falling rapidly. Today, one can buy a 10,000-computer botnet for $1,000 [41].

2) Broad Network Access and Rapid Elasticity Leading to More Immense, Flexible, and Sophisticated DDoS Attacks: With cloud computing's capabilities of broad network access and rapid elasticity, attackers can not only launch immense DDoS attacks, but also produce more flexible and more sophisticated DDoS attacks using heterogeneous thin or thick client platforms, which are discussed in the following.

- More immense DDoS attacks in cloud computing:
  The size and frequency of DDoS attacks have grown dramatically as attackers take advantage of botnets and other high-speed Internet access technologies to overwhelm their victim's network infrastructure. Fig. 4 shows the size of largest reported DDoS attack from 2002 to 2013. We can see obviously that the size of DDoS attack is increasing year by year. In March 2013, Spamhaus,

an organization that maintains lists of spammers, came under a massive DNS reflection DDoS attack. The attack volume was reportedly as high as 300 Gbps [42].

- More flexible DDoS attacks in cloud computing:

  At the same time, the prevalence of mobile devices, such as smartphones and tablets, are expected to become a significant launching platform for DDoS attacks against cloud computing. The lack of security on the majority of mobile devices, coupled with the rising bandwidth and processing power, makes them a platform ripe for hackers to compromise. Researchers reported that Android malware could be used to launch DDoS attacks in 2013 [43]. Malicious attackers now carry a powerful attack tool in the palm of their hands, which requires minimal skill to use. Because it is so easy for mobile device users to opt-in to DDoS attack campaigns, a considerable increase in the use of these attack tools will be expected in the following years.
- More sophisticated DDoS attacks in cloud computing:

  Not only are DDoS attacks getting larger and more frequent, but they are also becoming more sophisticated as they pinpoint specific applications (e.g., DNS, HTTP or VoIP) with smaller, more stealthy attacks [44]. Sophisticated low-bandwidth DDoS attacks use less traffic and increase their effectiveness by aiming at a weak point in the victim's system design. While it requires more sophistication and understanding of the attacked system, a low-bandwidth DDoS attack has three major advantages in comparison to a high-bandwidth attack: 1) Lower cost—since it uses less traffic; 2) Smaller footprint—hence, it is harder to detect; 3) Ability to hurt systems that are protected by flow control mechanisms [45].

3) Resource Pooling Leading to the Victims More Vulnerable to DDoS Attacks: In cloud computing, the service provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand [46].

Virtualization technology and multi-tenant infrastructure on the one hand make attackers launch DDoS attacks more easily, on the other hand cause the victims more vulnerable to DDoS attacks.

- Virtualization technology and multi-tenant infrastructure make attacker launch DDoS attacks more easily:

  Virtualization technology help attacks preset for attacks. In each cloud, attackers can deploy an ISO image (or a virtual machine) set up to connect immediately upon startup to one or more meeting points to receive marching orders [40]. virtual machines (VMs) built for this purpose will of course be optimized so as to use very little live memory or disk space: this way the attacker can streamline costs and launch even more VMs with the extra cash [40].
- Virtualization technology and multi-tenant infrastructure cause the victims more vulnerable to DDoS attacks:

  Researchers show that, on a DoS attack, the performance of a web server hosted in a VM can degrade by up to 23%, while that of a non-virtualized server hosted on the same hardware degrades by only 8% [47]. Since the cloud computing environment is inherently a multi-tenant infrastructure, an attack against a single customer is actually an attack against all customers in that given cloud, or at least a significant proportion of those customers, as they are sharing not only a common network infrastructure but a common compute infrastructure, a common memory infrastructure, a common storage infrastructure, etc [48]. A DDoS attack in virtualization occurs when one VM occupies all the available physical resources such that the hypervisor cannot support more VMs, and availability is imperiled [49]. Although the hypervisors limit the amount of physical resources allocated to each VM, Side channels have been used in recent works to bypass virtual machine isolation in the cloud [50]. A side-channel in a software program is a means of communication via a medium not intended for information transfer.

4) Rapid Elasticity and Measured Service Leading to a New Breed of DDoS Attacks: With rapid elasticity and measured service, adopters of the cloud service model are charged based on a pay-per-use basis of the cloud's server and network resources. With this model, a conventional DDoS attack on server and network resources is transformed in a cloud environment to a new breed of attack that targets the cloud adopter's economic resource, namely Economic Denial of Sustainability attack (EDoS) [51], [52].

A representative EDoS attack is Fraudulent Resource Consumption (FRC) attack. Unlike an application-layer DDoS attack that consumes resources with the goal of disrupting short-term availability, an FRC attack is a considerably more subtle attack that instead seeks to disrupt the long-term financial viability of operating in the cloud by exploiting the utility pricing model over an extended time period [53]. By fraudulently consuming web resources in sufficient volume (i.e., data transferred out of the cloud), an attacker (e.g., botnet) is able to incur significant fraudulent charges to the victim. In cloud computing, the goal of a FRC attack is to deprive the victim (i.e., regular cloud customers) of their long-term economic availability of hosting web contents that are publicly accessible. In other words, attackers, who act as legal cloud service clients, continuously send requests to website hosting in cloud servers to consume bandwidth, which bills to the cloud customer owning the website; seems to the web server, those traffic does not reach the level of service denial, and it is difficult to distinguish FRC traffic from other legitimate traffic [13]. A FRC attack succeeds when it causes financial burden on the victim [13].

C. Available Defense Mechanisms Categories

Since Yahoo, Amazon and other well-known web sites were subjected to DDoS attacks in 2000, researchers have presented many methods to mitigate DDoS attacks. Several taxonomies of DDoS attacks defense mechanisms have been presented in the literature [35], [37], [54]. The authors of [35] focus on DDoS flooding attacks and defense mechanisms in wired network

systems. Farahmandian *et al.* concentrate on the methods against DDoS attacks in cloud computing [54].

The defense mechanisms against network/transport-level DDoS flooding attacks can be classified into four categories based on the deployment location [35]:

1) Source-based mechanisms: Source-based mechanisms are deployed near the sources of the attack to prevent network customers from generating DDoS flooding attacks [35]. Some examples of source-based mechanisms include ingress/egress filtering, which filters packets with spoofed IP addresses at the source's edge routers based on the valid IP address range internal to the network [55], and Source Address Validity Enforcement (SAVE) Protocol [56]. SAVE protocol enables routers to update the information of expected source IP addresses on each link and block any IP packet with an unexpected source IP address [57].
2) Network-based mechanisms: These mechanisms are deployed inside networks and mainly on the routers of the Autonomous Systems (ASs). Generally there are two groups of DDoS attack detection techniques. The first group is called DDoS-attack-specific detection. The second group is called anomaly-based detection [57] (e.g., route-based packet filtering [58]). DDoS-attack-specific detection is an attack detection method based on the special features of different DDoS attacks. For example, using SYN cookie technique can resist SYN flood attacks. In a SYN flood, a victim server receives spoofed SYN requests at a high packet rate that contain fake source IP addresses. The SYN flood overwhelms the victim server by depleting its system resources (connection table memory) normally used to store and process these incoming packets, resulting in performance degradation or a complete server shutdown [37]. SYN cookies are particular choices of initial TCP sequence numbers by TCP servers. They allow a server to avoid dropping connections when the SYN queue fills up [37]. Anomaly-based detection models the behavior of normal traffic, and then reports any anomalies.
3) Destination-based mechanisms: In the destination-based defense mechanisms, detection and response is mostly done at the destination of the attack (i.e., victim) [35]. Some examples of destination-based mechanisms include input debugging [59], probabilistic packet marking [60], and hash-based IP traceback [61]. Input debugging is a link testing mechanism, in which the traceback process starts from the router closest to the victim and iteratively tests its upstream links until it can be determined which link is used to carry the attacker's traffic [59]. In probabilistic packet marking, routers in the path to the victim probabilistically mark packets (i.e., add routers' identification to each packet) so that the victim can identify the path of attack traffic and distinguish it from legitimate traffic after the detection [35]. In hash-based IP traceback, routers in the path to the victim keep a hash record of every packet passing through the router using Bloom Filter, which is a hash structure to reduce the memory requirement to store packet records [35].
4) Hybrid (Distributed) mechanisms: Hybrid defense mechanisms are deployed at (or their components are distributed over) multiple locations such as source, destination or intermediate networks and there is usually cooperation among the deployment points [35] (e.g., Active Internet Traffic Filtering (AITF), which enables a receiver to contact misbehaving sources and ask them to stop sending it traffic [62]).

The defense mechanisms against application-level DDoS flooding attacks can be classified into two categories based on their deployment location [35]:

1) Destination-based (server-side) mechanisms (e.g., DDoS-Shield uses statistical methods to detect characteristics of HTTP sessions and employs rate-limiting as the primary defense mechanism [35], [63]).
2) Hybrid (Distributed) mechanisms (e.g., Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) [64], [65]).

D. The Difficulty in Defeating DDoS Attacks in Cloud Environments

As mentioned earlier, when the networks migrate to cloud computing, DDoS attacks evolve new forms and characteristics. Some researchers have presented some new methods to defeat DDoS attacks in cloud computing. Farahmandian *et al.* give a review and comparison of the existing methods against DDoS attacks in cloud computing [54].

The authors of [66] use reactive/on-demand in-cloud DDoS mitigation service (scrubber service) for mitigating the application-layer and network-layer DDoS attacks with the help of an efficient client-puzzle approach. The generated crypto puzzle is being solved by the service consumer/user by the brute force method in order to prove its legitimacy for acquiring service.

A dynamic resource allocation strategy is presented in [67] to counter DDoS attacks against individual cloud customers. When a DDoS attack occurs, they employ the idle resources of the cloud to clone sufficient intrusion prevention servers for the victim in order to quickly filter out attack packets and guarantee the quality of the service for benign users simultaneously.

Lua *et al.* propose a novel approach to mitigate DDoS attacks using an intelligent fast-flux swarm network. Fast flux is a DNS technique. Fast-flux hosting allows a fully qualified domain name to have many IP addresses assigned to it. It uses a very short time-to-live (TTL) parameter for any particular name record. The hostnames will be reassigned at very high frequency. Since a fast-flux hosting technique in itself is not robust enough to cope with sophisticated DDoS attacks that exploit fast-flux service networks, swarm intelligence techniques are applied to imbue the network with autonomy to address the above fallacy [68]. An intelligent swarm network is required to ensure autonomous coordination and allocation of swarm nodes to perform its relaying operations. An intelligent water drop algorithm is applied for distributed and parallel optimization. The movement of bodies of water in nature inspires the

Intelligent Water Drop (IWD) algorithm. Water always finds the path of least resistance. This is well suited to designing a relay system for the swarm network [68]. The fast-flux technique is used to maintain connectivity between swarm nodes, clients, and servers. Fast-flux service networks also allow them to build a transparent service, which allows minimal modifications of existing cloud services (e.g., HTTP and SMTP) [68].

A practical solution is presented in [69] to collect data trace and analyze these data in parallel in a cloud computing platform named the Collaborative Network Security Management System (CNSMS). They use cloud storage to keep huge volume of traffic data and process it with a cloud computing platform to find the malicious attacks.

Although the above excellent works have been done to defeat DDoS attacks in cloud computing environments, DDoS attacks are still showing a rising trend. Some difficulties exist in defeating DDoS attacks in cloud computing environments.

1) Data collection: Most DDoS attacks mitigating mechanisms need to collect data to build normal profile or to detect abnormals. For example, many methods need to extract features of interest from network traffic to find statistic abnormal. Since DDoS attacks have increased in size in cloud environments, collecting tremendous and heterogeneous data with a low overhead is becoming more and more difficult. Moreover the facts that cloud traffic is distributed between network devices by load balancing and the multi-tenant nature of cloud environments make the task of data collection for marginating DDoS attacks harder to achieve.
2) Intelligent algorithm selection: Because DDoS attacks have increased in complexity in cloud environments, many intelligent algorithms have been used, including artificial neural network, chaotic analysis, Bayesian classification, game theory, hidden semi-Markov model (HSMM), fuzzy logic and so on. Due to the complexity of DDoS attacks, there is no single intelligent algorithm that can deal with all DDoS attacks. How to choose different intelligent algorithms according to different attacks is one of the difficult problems to solve.
3) React promptly: Prompt attacks response is particularly important in highly dynamic cloud environments. But the complexity of today's cloud makes promptly response to attacks challenging. For example, in a data center, there are a large number of heterogeneous security devices that need to cooperate, and there are a large number of protocols that need to be supported.

IV. SDN as a Good Tool to Defeat DDoS Attacks

As SDN provides a new and dynamic network architecture for cloud computing, the good features of SDN make it easier to detect and react DDoS attacks in cloud computing. In this section, we first summarize the good features of SDN that bring a lot of benefits for defeating DDoS attacks, and then provide an overview of the available methods using SDN to defeat DDoS attacks.

A. Good Features of SDN in Defeating DDoS Attacks

SDN brings us new chances to defeat DDoS attacks in cloud computing environments. We summarize the good features of SDN as follows [10], [11], [31], [70], [71].

1) Separation of the control plane from the data plane:
   SDN decouples the data plane and control plane and thus enables to establish easily large scale attack and defense experiments. High configurability of SDN offers clear separation among virtual networks permitting experimentation on a real environment [10]. Progressive deployment of new ideas can be performed through a seamless transition from an experimental phase to an operational phase. Moreover it enables innovation and evolution by providing a programmable network platform to implement, experiment, and deploy new ideas, new applications. The feature of SDN brings great convenience in putting forward new thoughts and methods of DDoS attacks mitigation.
2) A logical centralized controller and view of the network:
   The controller has network-wide knowledge of the system and global views to build consistent security police and to monitor or analysize traffic patterns for potential security threats. Centralized control of SDN permits dynamically quarantine of compromised hosts and authentication of legitimate hosts based on information obtained through requesting end hosts, requesting a Remote Authentication Dial In User Service (RADIUS) server for users' authentication information and system scanning during registration [10].
3) Programmability of the network by external applications:
   The programmability of SDN supports a process of harvesting intelligence from existing Intrusion Detection Systems (IDSs) [72] and Intrusion Prevention Systems (IPSs) [33]. More intelligent algorithms can be flexibly used based on different DDoS attacks.
4) Software-based traffic analysis:
   Software-based traffic analysis greatly enables innovation, as it is possible to improve the capabilities of a switch using any software-based technique [31]. Traffic analysis can be performed in real time using machine learning algorithms, databases and any other software tool. Traffic of interest can be explicitly directed to IPSs for Deep Packet Inspection (DPI) [10].
5) Dynamic updating of forwarding rules and flow abstraction:
   Dynamic updating of forwarding rules helps promptly respond to DDoS attacks. Based on the analysis, new or updated security policy can be propagated across the network in the form of flow rules [33]. If attacks are detected, SDN can install packet forwarding rules to switching devices to block the attack traffic from entering and propagating in a network [10].

B. DDoS Attacks Defense Mechanisms Using SDN

We classify the DDoS attacks defense mechanisms using SDN into three categories based on the deployment location:

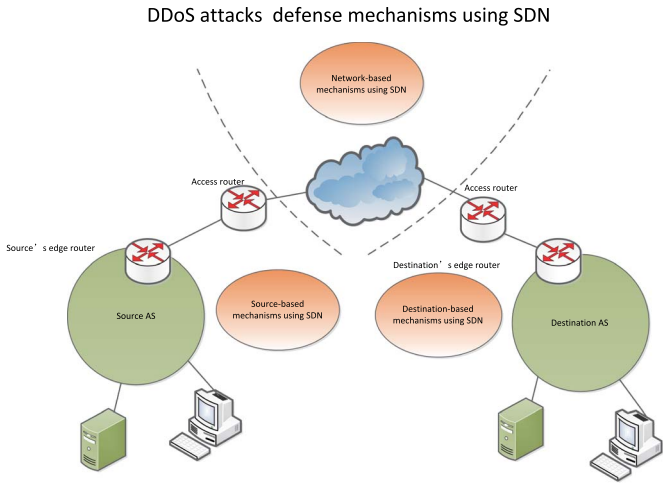

Fig. 5. A classification of the defense mechanisms against DDoS attacks using SDN.

source-based mechanisms using SDN, network-based mechanisms using SDN and destination-based mechanisms using SDN, as shown in Fig. 5.

1) Source-Based Mechanisms Using SDN: Most source-based mechanisms using SDN let SDN controllers detect anomaly traffic, filter the malicious packet, or validate the source IP address near the ingress of network.

By using the programmability of SDN, the authors of [73] show that a programmable home network router can provide the ideal platform and location in the network for detecting security problems in SOHO (Small Office/Home Office) networks. Four prominent traffic anomaly detection algorithms, threshold random walk with credit based rate limiting (TRW-CB algorithm), rate-limiting, maximum entropy detector and NETAD are implemented in an SDN context using OpenFlow compliant switches and NOX as a controller. Threshold random walk with credit based rate limiting (TRW-CB) is a classification method using sequential hypothesis testing (i.e., likelihood ratio test) to classify whether or not the internal host has a scanning infection. It is based on the observation that the probability of a connection attempt being a success should be much higher for a benign host than a malicious one. Rate Limiting uses the observation that an infected machine has different connection characteristic to limit new connection rate. The Maximum Entropy detector estimates the benign traffic distribution using maximum entropy estimation. Unlike TRW-CB and Rate Limiting, Maximum Entropy relies on examining every packet in order to build packet class distributions every t seconds. NETAD operates on rule-based filtered traffic in a modeled subset of common protocols. The filter removes uninteresting traffic based on the premise that the first few packets of a connection request are sufficient for traffic anomaly detection [73]. Experiments indicate that these algorithms are significantly more accurate in identifying malicious activities in the home networks as compared to the ISP and that the anomaly detectors can operate at line rates without introducing any performance penalties for the home network traffic.

Motivated by the flexibility of the SDN architecture and the observation that most mobile malware requires Internet connections, the authors of [21] design a system that detects mobile malware through real-time traffic analysis using the SDN architecture. Like Ingress filtering, the system can only allow traffic to enter network if its source addresses are within the expected IP address range. The access point, which is essentially an OpenFlow-enabled switch, is controlled by an OpenFlow controller. The access point forwards mobile traffic to the OpenFlow controller, and receives and installs flow entries from the controller through a secured channel. The malware detection system is a module inside the OpenFlow controller, which can extract the traffic information. Detection algorithms include: IP Blacklist, Connection Success Ratio, Throttling Connection, and Aggregation Analysis. IP Blacklist: A straightforward way to protect a network is maintaining a blacklist of malicious IP addresses, which can either be obtained from public available sources or from historic data, and denying immediately any network flow that involves an IP address in that blacklist. Connection Success Ratio: Connection Success Ratio is a malware detection algorithm based on connection success ratio, which leverages the observation that the successful connection probability for a benign host should be much higher than a malicious host. Throttling Connection: Throttling Connection can limit the rate of connections to new hosts and identify the infected clients based on the observation that during virus propagation, an infected machine will try to connect to as many different machines as fast as possible, while an uninfected machine behaves differently: connections are made at a lower rate, and are locally correlated (since repeated connections to recently accessed machines are likely). Aggregation Analysis: Aggregation Analysis is an algorithm that detects the infected hosts by identifying aggregates of "similar" communications based on the observation that when one host is infected by malware, multiple other hosts may be infected as well, especially in a large scale network and the infected client share behavioral characteristics in their network activities that are distinct from those of benign clients [21].

Source Address Validity Enforcement (SAVE) protocol enables routers to update the information of expected source IP addresses on each link and block any IP packet with an unexpected source IP address [57]. In cloud computing environments, routing locator spoofing problem is getting more serious since botnets are widely used. The current in-progress source address validation standard, i.e., SAVI, is not of enough protection due to the solution space constraints [74]. A mechanism named Virtual source Address Validation Edge (VAVE) is proposed to improve the SAVI solution. VAVE employs OpenFlow protocol to solve source address validation problem with a global view. OpenFlow devices are used to form a protective perimeter. Whenever a packet originated from outside of the perimeter reaches the perimeter, if it is not matched by any entry in the flow table of the device, the first packet will be redirected to the NOX controller [74]. A VAVE application of the NOX controller checks whether or not the source of the packet is valid based on generated rules.

We summarize source-based mechanisms using SDN as shown in Fig. 6.

2) Network-Based Mechanisms Using SDN: A lightweight method for DDoS attacks detection based on traffic flow

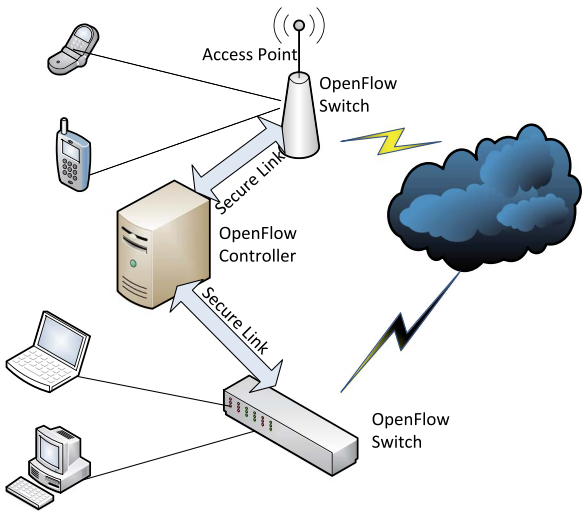

Fig. 6. Source-based mechanisms using SDN.

features is presented in [75], in which the extraction of such information is made with a very low overhead compared to traditional approaches. The method is divided into three modules [75]:

1) The Flow Collector module is responsible for periodically requesting flow entries from all Flow Tables of OF switches.
2) The Feature Extractor module receives the collected flows, extracts features that are important to DDoS flooding attack detection. These important features include: Average of Packets per flow (APf), Average of Bytes per flow (ABf), Average of Duration per flow (ADf), Percentage of Pair-flows (PPf), Growth of Single-flows (GSf), and Growth of Different Ports (GDP). The Feature Extractor module gathers them in 6-tuples to be passed to the classifier.
3) The Classifier module analyzes whether or not a given 6-tuple corresponds to a DDoS flooding attack or to legitimate traffic. Self Organizing Maps (SOMs) are used as the classification method.

Based on the capability of software-based traffic analysis of SDN, the method in [75] extracts features of interest with a low overhead when compared to traditional approaches. Based on the capability of a logical centralized controller and view of the network, the method is able to monitor more than one observation point. Based on the capability of programmability of the network by external applications, the method can use SOM to classify network traffic flows as either normal or abnormal.

Suh *et al.* [76] propose a novel content-oriented networking architecture (CONA) in which hosts request contents and their agents deliver the requested contents. Due to the accountability and content-aware supervision, CONA can react to resource exhaustive attacks like DDoS effectively. A DDoS attack is detected when the server that provides a given content type rece ives more requests than expected, based on a pre-defined range.

Chu *et al.* [77] propose a novel design of DDoS defender that is implemented on OpenFlow controller. DDoS defender can monitor the flows of OpenFlow switch and detect the DDoS attack via volume counting [31], [77].

In order to improve the scalability of proposed native OF approaches in [75], a new method combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments is presented by [78]. It designs a modular mechanism that permits anomaly detection and mitigation on SDN environments including collector, anomaly detection and anomaly mitigation. And it leverages the packet sampling capability of sFlow to acquire scalability improvements and to reduce the required communication between switches and OF controllers.

An OpenFlow-based intrusion prevention system, called SnortFlow in Xen-based cloud environment, is proposed in [79]. It inherits the intrusion detection capability from Snort and flexible network reconfiguration from OpenFlow. The evaluation results generated in the real cloud environment show the feasibility of SnortFlow.

Despite the success of OpenFlow, developing and deploying complex OF security services remains a significant challenge. FRESCO, an OpenFlow security application development framework designed to facilitate the rapid design, and modular composition of OF-enabled detection and mitigation modules is presented in [80]. FRESCO, which is itself an OpenFlow application, offers a Click-inspired programming framework that enables security researchers to implement, share, and compose together, many different security detection and mitigation modules. The FRESCO framework consists of an application layer (which provides an interpreter and APIs to support composable application development) and a security enforcement kernel (SEK, which enforces the policy actions from developed security applications). Both components are integrated into NOX, an open-source openflow controller.

Since SDN defines abstractions to represent network entities and logically centralize them in a network controller, the authors of [81] argue that SDN's abstraction is the most promising way to successfully create agent-based architectures to control and manage large-scale parts of the Internet. An agent-based framework, AgNOS, for the building of cooperative SDNs that extend their domains beyond enterprise networks is presented. This framework is built on top of the abstractions provided by SDN. A case study on mitigation of DDoS attacks is studied when thousands of attackers perform malicious packet flooding and SDN domains must cooperate to cope with packet filtering at the source [81].

Yu *et al.* propose a memory-efficient system for Distributed and Collaborative per-flow Monitoring (DCM) in [82]. DCM uses Bloom filters to represent monitoring rules using a small size of memory. It utilizes the tremendous convenience brought by SDN paradigm to install a customized and dynamic monitoring tool into the switch data plane. The novel monitoring tool used by DCM is called two-stage Bloom filters, including an admission Bloom filter to accept all flows assigned to the switch and a group of action Bloom filters to perform different measurement actions. SDN also allows DCM to perform updates or reconstruction of the two-stage Bloom filters in the switch data plane.

In a multi-tenant model like cloud computing, several stakeholders are involved. Distinguishing tenants' activities and provisioned resources, in the time domain, is the key factor for

accountability and anomaly detection [83]. Two types of solutions are introduced to address the heterogeneous environment. The first type introduces methods for monitoring tenants' activities, when an OpenFlow controller is not available. In these methods, tenants' specifications are retrieved from the networking and identity service. Then, raw monitoring data are processed for building per-tenant traffic statistics. The second type benefits from an OpenFlow controller to build the per-tenant view. In this approach, the controller provides the unified view of the network, and is aware of the tenant logic. The monitoring node communicates with the controller to build per-tenant view of the network and generates monitoring information for each tenant [83].

A new United States patent [84] presents a method for mitigating of denial of service (DDoS) attacks using SDN. The method comprises receiving a DDoS attack indication performed against at least one destination server, programming each network element in the SDN to forward a packet based on a diversion value designated in a packet diversion field, upon reception of the DDoS attack indication, instructing at least one peer network element in the SDN to mark a diversion field in each packet in the incoming traffic addressed to the destination server to allow diversion of the packet to a security server, and instructing edge network elements in the SDN to unmark the diversion field of each packet output by the security server, wherein each network element in the SDN is programmed to forward the unmarked packets processed by the security server to at least one destination server [84].

Radware company proposes DDoS protection as an SDN network service. The solution relies on the following components: Cisco ONE SDN controller; Cisco switches and routers that are SDN enabled; DefenseFlow anti-DDoS SDN application; and DefensePro attack mitigation solution [85]. The result is complete abstraction of the anti-DDoS resource provisioning and alignment with network operations to provision, manage and monitor DDoS protection as a service within the OpenFlow ecosystem.

Most mechanisms discussed in this subsection can be designed as SDN applications. A such SDN application generally includes four function modules: flow collector, feature extractor, anomaly detection, and attack mitigation, as shown in Fig. 7. Different mechanisms use similar function modules but with different implementation methods.

3) Destination-Based Mechanisms Using SDN: IP traceback can be used to find the origins and paths of attacking traffic. However, so far, most approaches for IP traceback are hard to be deployed in the Internet because of deployment difficulties. An incrementally deployable approach is presented in [86] based on sampled flows for IP traceback (SampleTrace). In SampleTrace, it is not necessary to deploy any dedicated traceback software and hardware at routers, and an AS-level overlay network is built for incremental deployment. The authors of [86] theoretically analyze the quantitative relation among the probability that a flow is successfully traced back various ASlevel hop number, independently sampling probability, and the packet number that the attacking flow comprises. Although it needs to build an AS-level overlay network for incremental deployment, it can be extended to SDN networks.

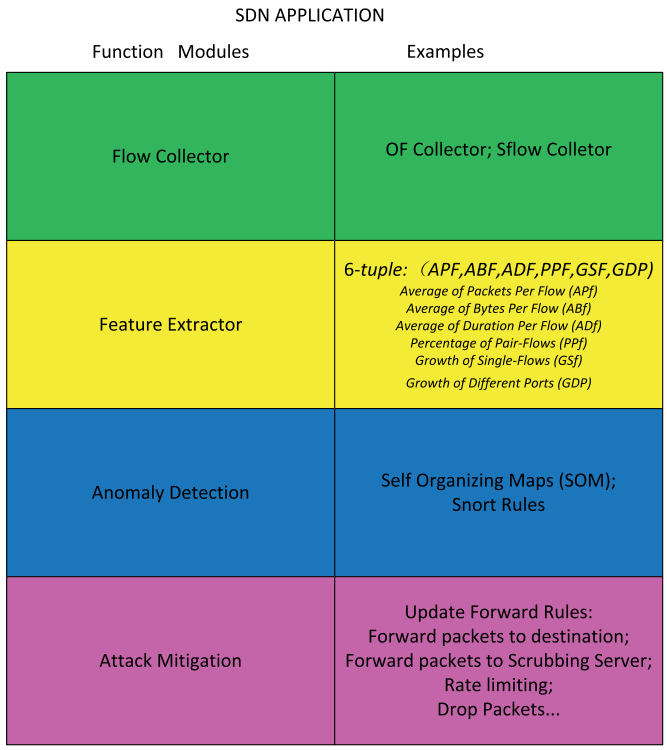

Fig. 7. Function modules in SDN applications for network-based mechanisms.

The authors of [87] show how packet histories (i.e., the full stories of every packet's journey through the network) can simplify network diagnosis. To demonstrate the usefulness of packet histories and the practical feasibility of constructing them, NetSight, an extensible platform that captures packet histories and enables applications to concisely and flexibly retrieve packet histories of interest, is built in [87]. Atop NetSight, four applications that illustrate its flexibility are presented: an interactive network debugger, a live invariant monitor, a path-aware history logger, and a hierarchical network profiler.

OFRewind is a tool for recording and playing SDN control plane traffic, and it enables scalable, temporally consistent, centrally controlled network recording and coordinated replay of traffic in an OpenFlow controller domain [88]. It takes advantage of the flexibility afforded by the programmable control plane, to dynamically select data plane traffic for recording [88].

Nikhil Handigol *et al.* introduce ndb, a prototype network debugger inspired by gdb [89]. A postcard-based approach is used in ndb to reconstruct the path taken by a packet, and it can record flow table state via a proxy and log packet traces.

Although these works pay more attention on troubleshooting than on locating source IP address, some techniques can be used in IP traceback like input debugging.

Table I summarizes these mechanisms.

V. SDN as a Victim of DDoS Attacks

SDN holds great promise in terms of mitigating DDoS attacks in cloud computing environments by decoupling data plane from control plane. However, the security of SDN itself remains to be addressed. In this section, we first discuss SDN itself may be a target of DDoS attacks. Then we provide an overview of available solutions to this problem.

TABLE I
COMPARISON OF DDoS ATTACKS DEFENSE MECHANISMS USING SDN

Types	Publication	SDN capabilities exploited	Description of the solution
Source-based Mechanisms Using SDN	Mehdi *et al.* [73]	Programmability	A programmable home network router using OpenFlow compliant switches and NOX as a controller detects security problems in SOHO.
	Ruofan Jin *et al.* [21]	Traffic analysis	The access point, which is an OpenFlow-enabled switch and is controlled by an OpenFlow controller, detects mobile malware through real-time traffic analysis.
	Yao *et al.* [74]	Traffic analysis, dynamic rules updating and global views.	VAVE that employs OpenFlow protocol to solve source address validation problem with a global view is proposed to improve the SAVI solution.
Network-based Mechanisms Using SDN	Braga *et al.* [75]	Traffic analysis and centralized control	Statistic information in the flow table is used to classify traffic as normal or malicious by Self Organizing Maps.
	Suh *et al.* [76]	Traffic analysis and dynamic rules updating	A novel content-oriented networking architecture (CONA) can react to DDoS attacks by use of the accountability and content-aware supervision.
	Chu *et al.* [77]	Traffic analysis and dynamic rules updating	A novel DDoS defender, which is implemented on OpenFlow controller can monitor the flows of OpenFlow switch and detect the DDoS attack via volume counting.
	Giotis *et al.* [78]	Traffic analysis and dynamic rules updating	It leverages the packet sampling capability of sFlow to acquire scalability improvements and to reduce the required communication between switches and OF controllers.
	Xing *et al.* [79]	Programmability and dynamic rules updating	It inherits the intrusion detection capability from Snort and flexible network reconfiguration from OpenFlow to build an Intrusion Prevention System.
	Shin *et al.* [80]	Programmability	It presents FRESCO, which is itself an OpenFlow application. It can offer a Click-inspired programming framework.
	Passito *et al.* [81]	Abstraction ablility	An agent-based framework, AgNOS, for the building of cooperative SDNs that extend their domains beyond enterprise networks is presented, which is built on top of the abstractions provided by SDN.
	Yu *et al.* [82]	Programmability and dynamic rules updating	It proposes a memory-efficient system for Distributed and Collaborative per-flow Monitoring, called DCM. DCM uses Bloom filters to represent monitoring rules and installs a customized and dynamic monitoring tool into the switch data plane.
	TaheriMonfared *et al.* [83]	Global views and centralized control	The monitoring node communicates with the controller to build per-tenant view of the network and generates monitoring information for each tenant.
	An United States patent [84]	Programmability and dynamic rules updating	It comprises receiving a DDoS attack indication performed against at least one destination server; programming each network element in the SDN to forward a packet based on a diversion value designated in a packet diversion field, upon reception of the DDoS at indication.
	Radware company [85]	Global views and centralized control	It proposes complete abstraction of the anti-DDoS resource provisioning and alignment with network operations to provision, manage and monitor DDoS protection as a service within the OpenFlow ecosystem.
Destination-based Mechanisms Using SDN	Tian *et al.* [86]	Dynamic rules updating	It theoretically analyzes the quantitative relation among the probability that a flow is successfully traced back various ASlevel hop number, independently sampling probability, and the packet number that the attacking flow comprises.
	Handigol *et al.* [87]	Global views and centralized control	NetSight, an extensible platform that captures packet histories and enables applications to concisely and flexibly retrieve packet histories of interest, is built.

A. Possible DDoS Attacks on SDN

SDN itself may be a target of DDoS attacks. Since SDN is vertically split into three main functional layers, including infrastructure layer, control layer, and application layer, as shown in Fig. 2, potential malicious DDoS attacks can be launched on these three layers of SDN's architecture. Based on the possible targets, we can classify the DDoS attacks launching on SDN into three categories: application layer DDoS attacks, control layer DDoS attacks, and infrastructure layer DDoS attacks, as shown in Fig. 8.

1) Application layer DDoS attacks: There are two methods to launch application DDoS attacks. One is to attack some applications, the other is to attack northbound API. Since isolation of applications or resources of SDN is not well

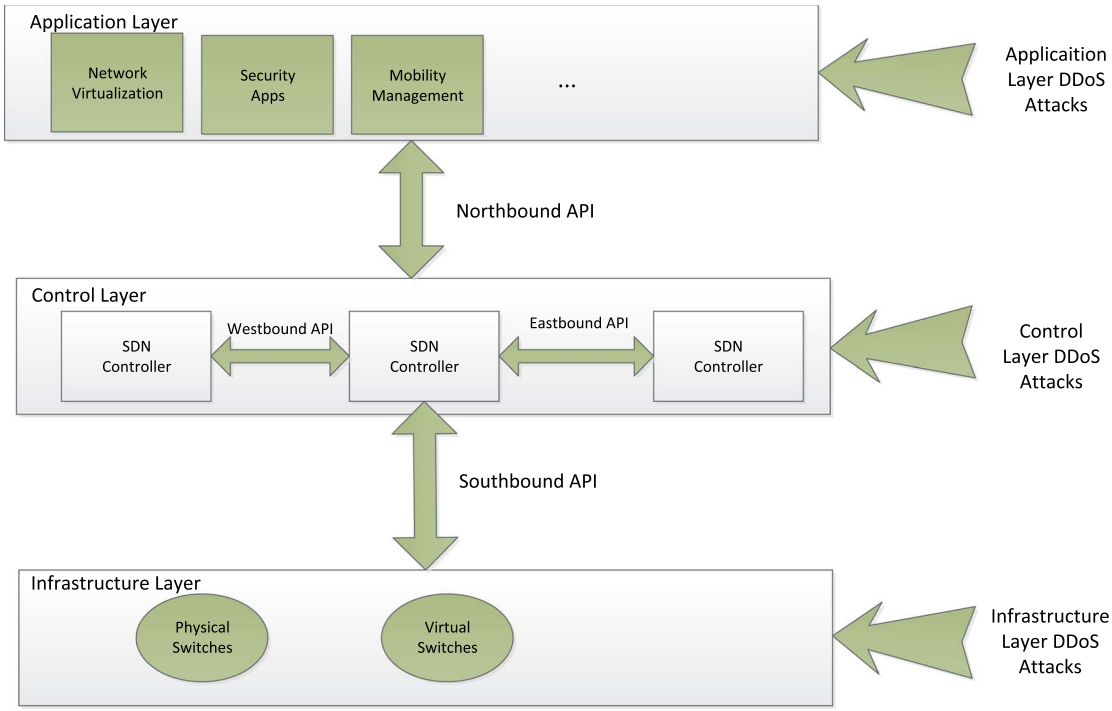

Fig. 8. Potential DDoS attacks can be launched on the three layers of the SDN's architecture.

solved [34], DDoS attacks on one application can affect other applications.

2) Control layer DDoS attacks: The controllers could potentially be seen as a single point of failure risk for the network, so they are a particularly attractive target for DDoS attack in the SDN architecture. The following methods can launch control plane DDoS attacks: attacking controller, northbound API, southbound API, westbound API or eastbound API. For example, many conflicting flow rules from different applications may cause DDoS attacks on the control layer. Within the operation of SDN, data plane will typically ask the control plane to obtain flow rules when the data plane sees new network packets that it does not know how to handle. There are two options for the handling of a new flow when no flow match exists in the flow table: either the complete packet or a portion of the packet header is transmitted to the controller to resolve the query [11]. With a large volume of network traffic, sending the complete packet to the controller would occupy high bandwidth [11].
3) Infrastructure layer DDoS attacks: There are two methods to launch data plane DDoS attacks. One is to attack some switches, the other is to attack southbound API. For example if only header information is transmitted to the controller, the packet itself must be stored in node memory until the flow table entry is returned. In this case, it would be easy for an attacker to execute a DDoS attack on the node by setting up a number of new and unknown flows. As the memory element of the node can be a bottleneck due to high cost, an attacker could potentially overload the switch memory (e.g., targeting to exhaust Ternary Content Addressable Memory (TCAM). [11]. The generated fake flow requests can produce many useless flow rules that need to be held by the data plane, thus making the data plane hard to store flow rules for normal network flows [11].

To demonstrate the feasibility of DDoS attacks, a new SDN network scanning prototype tool (named SDN scanner) is proposed in [90] to remotely fingerprint networks that deploy SDN. This method can be easily operated by modifying existing network scanning tools (e.g., ICMP scanning and TCP SYN scanning). The attack can be conducted to an SDN network by a remote attacker, and it can significantly degrade the performance of an SDN network without requiring high performance or high capacity devices.

It is shown that DDoS attack can overwhelm the controller in SDN architecture [91]. One serious scenario of DDoS that can directly affect the controller is swamping the controller with *Packet_In* events. Any new packets that do not have a match in the flow table will be sent to the controller for processing. Most DDoS attacks use spoofed source address, which translates into new incoming packet at the switch. This part is considered one of the advantages of SDN where the control plane is separated and manageable at the controller. It is also the main disadvantage when the number of new incoming packets is greater than the secure channel's bandwidth and the controller's processing power. In DDoS attacks, a large number of packets are sent to a host or a group of hosts in a network. If the source addresses of the incoming packets are spoofed, which they usually are, the switch will not find a match and has to forward the packet to the controller. The collection of legitimate and the DDoS spoofed packets can bind the resources of the controller into continuous processing that exhausts them. This will make the controller unreachable for the newly arrived legitimate packets

TABLE II
POSSIBLE DDoS ATTACKS ON SDN AND AVAILABLE SOLUTIONS

Possible DDoS attacks	Attack implementation methods	Available solutions
Application layer DDoS attacks	by attacking application	FortNOX [92]
	by attacking northbound API	
Control layer DDoS attacks	by attacking controller	Lightweight and fast DDoS detection [91]
	by attacking northbound API	Transport Layer SecurityTLS) [18]
	by attacking southbound API	SDMNs [95], FortNOX [92]
	by attacking westbound API	Onix [96], MCSSDN [97]
	by attacking eastbound API	AVANT-GUARD [98], Access Control [99]
Infrastructure layer DDoS attacks	by attacking switch	Transport Layer SecurityTLS) [18]
	by attacking southbound API	AVANT-GUARD [98], Rate Limiting [99]

and may bring the controller down causing the loss of the SDN architecture. Even if there is a backup controller, it has to face the same challenge [91].

The authors of [92] show that OF applications may compete/ contradict, override one another, incorporate vulnerabilities or possibly be written by adversaries. In the worst case an adversary can use the deterministic OF application to control the state of all OF switch in the network. A rule conflict is said to arise when the candidate OpenFlow rule enables or disables a network flow that is otherwise inversely prohibited (or allowed) by existing rules. Hackers may use rule conflict to launch DDoS attacks.

Because DDoS attacks use forged source IP address or faked traffic, simple authentication mechanism could mitigate forged or faked traffic flows. But if an attacker assumes the control of an application server that stores the details of many users, it can easily use the same authenticated ports and source MAC addresses to inject authorized, but forged, flows into the network [34].

OpenFlow provides optional support for encrypted Transport Layer Security (TLS) communication and a certificate exchange between the switches and the controller(s) [18]. As is well-known in the security community, using TLS/SSL does not per se guarantee secure communications, and it may compromise the controller device link [34]. The security of those communications is as strong as its weakest link, which could be a self-signed certificate, a compromised Certificate Authority, or vulnerable applications and libraries [34]. Moreover, the TLS/SSL model is not enough to establish and assure trust between controllers and switches. Once an attacker gains access to the control plane, it may be capable of aggregating enough power force (in terms of the number of switches under its control) to launch DDoS attacks. This lack of trust guarantees could even enable the creation of a virtual black hole network (e.g., by using OpenFlow-based slicing techniques allowing data leakage while the normal production traffic flows) [34].

For specific switches or controllers, they may have some special DDoS attacks weaknesses. For example, DDoS vulnerabilities that arise from an OpenFlow reactive rule design can be found in a layer 2 learning switch that is included in the POX controller as follows [93].

- The controller instructs the switches to flood multicast without installing a rule to match future multicast packets. Therefore, every multicast packet is sent to the controller, leaving an attacker a direct avenue to DDoS the controller with a traffic flood to a multicast address.
- Traffic to unknown MAC addresses is flooded without a rule insertion or a limit counter, creating another controller DDoS vulnerability.
- Since this application inserts rules into switches based on source MAC addresses, an attacker can generate an unlimited number of rules, quickly filling up a switch's flow-table by crafting packets with random source MAC addresses destined to a known network host. Additionally, since the mapping has no age-out or removal mechanism, an attacker could fill the memory of the controller by generating lots of traffic from random MAC addresses to other unknown MAC addresses on the network, which results in added mappings, but no new flows.

Dover *et al.* demonstrate a vulnerability in the Open Floodlight controller that allows an attacker with access to the OpenFlow control network to selectively deny communications between an individual switch and the controller, eventually disabling the switch [94].

B. Available Solutions

We summarize possible DDoS attacks on SDN and available solutions in Table II.

The authors of [33] point out that three issues of SDN include trust between all involved layers, SDN's control plane centralization and limited space in flow-tables. In order to overcome these problems, some efforts have already been taken.

FortNox is a new security policy enforcement kernel as an extension to the open source NOX OpenFlow controller, which mediates all Open-Flow rule insertion requests [92]. FortNOX implements role-based authentication for determining the security authorization of each OF application (rule producer), and enforces the principle of least privilege to ensure the integrity of the mediation process.

The use of oligarchic trust models with multiple trust-anchor certification authorities (e.g., one per sub-domain or per controller instance) is a possibility [34]. Moreover, securing communications with threshold cryptography across controller replicas (where the switch will need at least n shares to get a valid controller message) may be helpful. Additionally, the use of dynamic, automated and assured device association mechanisms may be considered, in order to guarantee trust between the control plane and data plane devices [34].

The authors of [95] propose a novel secure control channel architecture based on Host Identity Protocol (HIP). IPsec

tunneling and security gateways are widely used in today's mobile networks. The proposed architecture utilizes these technologies to protect the control channel of Software-Defined Mobile Networks (SDMNs). The proposed architecture is implemented in a test bed and the security features are analyzed. Moreover, the performance penalty of security of proposed architecture is measured and its ability to protect the control channel from various IP based attacks is analyzed.

The use of intrusion detection systems with support for runtime root-cause analysis could help identify abnormal flows [34]. This could be coupled with mechanisms for dynamic control of switch behavior (e.g., rate bounds for control plane requests).

The authors of [99] propose rate limiting, event filtering, packet dropping and timeout adjustment to defeat DDoS attacks. Rate limiting on the control channel and/or the data interface can allow the controller and/or the switch respectively to remain responsive during a DDoS attack, although it cannot protect other users from negative effects. Enforcement of access control lists in the form of flow rules on the table of the OpenFlow switch is also a feasible and low-cost approach.

A lightweight and fast DDoS detection mechanism is proposed in [91] based on entropy, to protect the controller by taking into account the abilities of the controller. Its broad view of the whole network is used for adding entropy statistics collection. And the author implements the proposed mechanism using Mininet and POX controller. Due to the limited resources of the controller, an early detection could be finished within the first few hundred packets of the attack.

A platform called Onix is presented in [96], in which a network control plane can be implemented as a distributed system. Onix provides a general API for control plane implementations, while allowing them to make their own trade-offs among consistency, durability, and scalability [96]. The authors of [97] present a a fault tolerant controller structure named MCSSDN to provide better security to the structure of SDN network, in which each device is managed by multiple controllers rather than a single one as in a traditional manner with Byzantine Fault-Tolerance (BFT) algorithm. It can resist Byzantine attacks on controllers and the communication links between controllers and SDN switches [97]. Although DDoS attacks can be mitigated by the use of multiple controllers, without careful rule design controllers still can be exposed to denial of service attacks.

AVANT-GUARD [98] is a new framework to advance the security and resilience of OpenFlow networks with greater involvement from the data-plane layer. The goal of AVANT-GUARD is to make SDN security applications more scalable and responsive to dynamic network threats. It address two security challenges for SDN-enabled networks. The first goal is to secure the interface between the control plane and the data plane and shield it from knowledgeable adversaries. To achieve this, AVANT-GUARD proposes a connection migration technique on the data plane to protect the control plane from the saturation attacks [100]. The second goal is to improve responsiveness so that security applications can efficiently access network statistics to response to threats. AVANT-GUARD addresses this by creating actuating triggers that can be inserted by the control plane to register asynchronous call back and by adding conditional flow rules that are activated when a predefined trigger condition is detected.

VI. Open Problems

There are many open research problems that are still not well investigated and need to be addressed by future research efforts. In this section, we discuss some of the most important open research issues to mitigate DDoS attacks in cloud computing environments by use of SDN.

A. How to Defeat Application-Level DDoS Attacks Using SDN

As we mentioned in Section II-A, application-level DDoS flooding attacks are another important type of DDoS attacks. They generally consume less bandwidth and are stealthier in nature compared to volumetric attacks, since they are very similar to benign traffic [35]. However, application-level DDoS flooding attacks usually have the same impact to the services since they target specific characteristics of applications, such as HTTP, DNS, or Session Initiation Protocol (SIP) [35].

According to the research by Gartner, there will be noticeable growth in the incidence of application-level DDoS attacks [101]. Access to payload information is crucial for application level DDoS attacks mitigation. Moreover, this information needs to be obtained at considerably reduced latencies in order to respond appropriately. But neither controller nor vSwitches has L4–7 application awareness. SDN architectures, by design, only provide the visibility and control required to implement security at the lower layers of the network stack [102]. For Example, in its current version, OpenFlow handles mostly layer 2/3 network traffic information, and the entire packet may be sent to the controller only in some special cases (because of no available buffers in the switch or the first packet of a given unknown flow) [9]. Thus, applications that need to have access and to manipulate data packet payload cannot benefit from the current OpenFlow implementation as both deep packet inspection and aggressive polling of the data plane can rapidly cause degradation of the latter's performance [9].

The challenge in applying SDN to Layer 4–7 networking is that this represents a diverse set of highly specialized applications that are difficult to consolidate and centralize. What's more, specialized hardware is often required to deliver high performance Layer 4–7 services [103].

SDN has the potential to significantly impact traditional Layer 4–7 appliances by offering more flexible, easy-to-manage and less expensive software-based functionality. The current leaders in Layer 4–7 will need to enhance their product offerings with SDN technologies to continue to be successful in this market [103]. L4–7 DPI and metadata engine can provide controller and its applications with App IDs and metadata to make smarter decisions [104].

Major efforts need to be spent in this area in order to propose solutions with good trade-offs between performance and security.

B. How to Defeat Mobile DDoS Attacks Using SDN

Prolexic Technologies reported that mobile applications are being used in DDoS attacks against enterprise customers [105]. The prevalence of mobile devices and the widespread availability of downloadable apps can be used for DDoS. So a considerable increase can be predicted in the use of these attack tools. Because mobile networks use super proxies, the method that simply uses a hardware appliance to block source IP addresses may not be effective, since it will also block legitimate traffic. Effective attack mitigation requires an additional level of fingerprinting and human expertise [106]–[109], so specific blocking signatures should be developed on-the-fly and applied in real-time.

Although some efforts have been done to extend SDN capability to mobile devices to provide true end-to-end SDN solutions for many network problems (e.g., QoS, virtualization, and fault diagnosis), more research needs to be done to defeat mobile DDoS attacks using SDN [110].

C. How to Implement Multiple Locations Defensive

Multiple locations defensive is comprised of multiple defense nodes deployed at various locations such as source, destination or networks [35]. For instance, detection can be done at the victim side and the response can be initiated and distributed to other nodes by the victim. Many multiple locations defensive methods have been presented in traditional networks. So we believe that, with widely deployment of SDN, there are a lot of research opportunities in designing multiple locations defensive methods using SDN to defeat DDoS attacks.

D. How to Use Cross-Layer Traffic Analysis

Cross-Layer traffic analysis is looking at the information at multiple protocol layers simultaneously to detect and respond to the DDoS attacks. Current SDN architectures focus on L2–L4. There is a need to extend traffic intelligence to L4–L7. Recently, the market and the ONF have begun to expand the dialogue around SDN, from defining and implementing the building blocks for L2–L3 networking and overlay networking to include the additional capabilities that exist beyond layer 4 [111].

E. How to Cooperate Among the Key Defensive Points

Since attackers cooperate to perform successful attacks, defenders must also form alliances and collaborate with each other to defeat DDoS attacks [35]. Cooperation among the key defensive points can be greatly beneficial to attacks prevention, detection, and response. The feature of global view and dynamic updating of forwarding rules of SDN will greatly reduce the cost of cooperation. A cooperative defense mechanism is an effective way to combat DDoS attacks. Although many cooperative defense mechanisms have been proposed in traditional networks, this topic has not been well researched in SDN.

F. How to Build a DDoS Attacks Tolerant System Using SDN

The experience in DDoS attack mitigation indicates that it is difficult to completely prevent DDoS attacks, and it is often impossible to accurately detect the act of DDoS attacks and stop them early enough [112]. Therefore, it is desirable to build a DDoS attacks tolerant system, which is designed by fault-tolerant design approach and can operate correctly despite attacks existence. For instance, a DDoS attacks tolerant system may provide services meeting service-level agreement (SLA) even under an attack by triggering automatic mechanisms to regain and recover the compromised services and resources. Other descriptions used for similar themed research include Survivability, Resilience, Trustworthy Systems, Byzantine Fault Tolerance, and Autonomic Self-Healing Systems.

A DDoS attacks tolerant system often has some essential properties such as redundancy, diversity and independence [113].

- Redundancy: Alternative systems and components are included, so that any one can perform the required function if the others fail.
- Diversity: Different components can be used based on different designs and principles, from different vendors.
- Independence: Independence is achieved by electrical isolation, physical separation and independence of communications between systems.

Although some efforts on building a DDoS attacks tolerant system have been done [114], [115], how to use SDN characteristics to realize the tolerant system is a new direction that needs to be addressed by future research efforts.

VII. Broader Perspectives

Since SDN is just one of the promising technologies in next-generation networks, many other technologies may affect the development of SDN. Meanwhile, SDN may have impacts on them as well. With a broader horizon, we also identify some research opportunities in other related areas.

A. Big Data Analytics

Big data is information assets whose complexity hinders them from being managed, queried and analyzed through traditional data storage architectures, algorithms, and query mechanisms [116]. The complexity of big data is defined through 3 V's:

- *Volume* referring to terabytes, petabytes, or even exabytes (1000^6 bytes) of stored information.
- *Variety* referring to the co-existence of unstructured, semi-structured and structured data.
- *Velocity* referring to the rapid pace at which big data is being generated.

As DDos attackers take advantage of botnets and other high-speed Internet access technologies, the size of DDoS attacks has grown dramatically. For example, the size is as high as 300 Gbps in 2013 DDoS attack to Spamhaus. Therefore,

traditional data analysis methods have many difficulties in defeating DDos attacks.

The application of big data analytics to mitigate DDoS attacks problems becomes more and more attractive because its ability to comprehensively analyze large volumes of disparate and complex data, such as threats, risks and incidents [116]–[118]. Kamaldeep Singh *et al.* build open source tools, such as Hadoop, Hive and Mahout, to detect Peer-to-Peer Botnet attacks using machine learning approach [118].

But the current IT infrastructures have innate shortcomings when it comes to big data [119]. SDN gives good chances to satisfy the requirements of big data anylytics such as automation and scalability.

B. Network Function Virtualization

Network function virtualization has become very popular in both wired networks and wireless networks. With network virtualization, multiple Virtual Networks (VNs) operated by different Service Providers (SPs) can dynamically share the physical substrate networks operated by Infrastructure Providers (InPs) [120], [121]. So network virtualization gives each 'tenant' in a data center its own network topology and control over its traffic flow [122]. By allowing multiple heterogeneous network architectures to cohabit on a shared physical substrate, network virtualization provides flexibility, promotes diversity, and promises security and increased manageability [120], [123].

SDN is an appealing platform for network virtualization because each tenant's control logic can run on a controller rather than on physical switches [122]. In a virtualized network, DDoS attacks can be launched by one virtual network to attack other virtual networks or the substrate that controls the different virtual networks [124].

In network virtualization, a widely used assumption is that different parties are always trusted [121]. However, this assumption may not be valid, since there are a large number of intelligent devices/nodes with self adaptation/context awareness capabilities in network virtualization [121]. A compromised party can take advantage of the virtualization mechanisms to launch DDoS attacks.

Research on DDoS attacks in virtualized networks that use SDN can be a promising research direction.

C. Information-Centric Networking

Information-Centric Networking (ICN) is a novel architecture that has been proposed as a solution for increasing the efficiency of content delivery and content availability [125]–[128]. Current Internet is information-driven, yet networking technology is still focused on the idea of location-based addressing and host-to-host communications [18]. ICN provides a location-independent network architecture in which the content is named. A number of research projects have studied ICN approaches, such as Named Data Networking (NDN) [129], PURSUIT [130].

The separation principle between information processing and forwarding in ICN is aligned with the decoupling of the data plane and control plane in SDN [18]. OpenFlow is expected to become the intermediary for migration from the current Internet to ICN. An architecture and implementation of an OpenFlow-based ICN are presented in [125].

Interest flooding and content/cache poisoning are two new types of Named Data Networking (NDN)-specific DDoS attacks. NDN is a new network architecture based on named content. By naming data instead of its locations, NDN transforms data into a first-class entity and makes itself an attractive and viable approach to meet the needs for many current and emerging applications [131]. NDN routers include the following components [131]:

- Content Store (CS), used for content caching and retrieval;
- Forwarding Interest Base (FIB), which contains a table of name prefixes and corresponding outgoing interfaces;
- Pending Interest Table (PIT), a table containing currently unsatisfied interests and corresponding incoming interfaces;

Interest flooding is a kind of DDoS attack aiming at the PIT state in NDN routers. In this attack, the adversary uses a large set of zombies to generate a large number of closely spaced interest packets, aiming to overflow PIT's in routers, preventing them from handling legitimate interests, and/or to swamp the specific content producer(s). Content/cache poisoning's goal is to cause routers to forward and cache corrupted or fake content, consequently preventing consumers from retrieving legitimate content [131].

Research on DDoS attacks in SDN-based ICN can be an interesting research direction.

VIII. Conclusion

In this paper, we first discussed the reasons why DDoS attacks are growing in cloud computing environments. Then we summarized the difficulty in defeating DDoS attacks in cloud computing environments. In addition, we presented some good features of SDN-based cloud in defeating DDoS attacks and discussed some challenges of SDN-based cloud. Since SDN-based cloud is still in its concept phase, we provided a comprehensive survey on some of the works that have already been done to defend DDoS attacks using SDN. We categorized the existing methods in three different class and presented a thorough comparison. Since SDN may be a victim of DDoS attacks, we reviewed the studies about how to launch DoS attacks on SDN and how to deal with this problem. We also discussed some significant open problems, including how to defeat application-level DDoS attacks using SDN, how to defeat mobile DDoS attacks using SDN, how to implement multiple locations defensive, how to use cross-layer traffic analysis, how to cooperate among the key defensive points, and how to build a DDoS attacks tolerant system using SDN. Finally, we explored some broader perspectives, such as big data analytics,

network virtualization and ICN to identify more research opportunities.

In summary, SDN brings a fascinating dilemma: a promising tool to defeat DDoS attacks in cloud computing environments, versus a vulnerable target to DDoS attacks. It is in favor of the community to study how to make full use of SDN's advantages to defeat DDoS attacks and how to prevent SDN itself becoming a victim of DDoS attacks in cloud computing environments. This paper attempts to briefly explore the current technologies related to SDN and DDoS attacks, and we discuss future research that may be beneficial in these issues.

ACKNOWLEDGMENT

The authors would like to thank the editors and reviewers for their careful examination of the manuscript and valuable comments, which have greatly helped to improve the quality of the paper.

REFERENCES

[1] G. Pallis, "Cloud computing: The new frontier of Internet computing," *IEEE Internet Comput.*, vol. 14, no. 5, pp. 70–73, Sep. 2010.

[2] T. Taleb, "Toward carrier cloud: Potential, challenges, and solutions," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 80–91, Jun. 2014.

[3] F. R. Yu and V. C. M. Leung, *Advances in Mobile Cloud Computing Systems*. New York, NY, USA: CRC Press, 2015.

[4] Y.-D. Lin, D. Pitt, D. Hausheer, E. Johnson, and Y.-B. Lin, "Software-defined networking: Standardization for cloud computing's second wave," *Computer*, vol. 47, no. 11, pp. 19–21, Nov. 2014.

[5] R.-I. Chang and C.-C. Chuang, "A service-oriented cloud computing network management architecture for wireless sensor networks," *Ad Hoc Sens. Wireless Netw.*, vol. 22, no. 1/2, pp. 65–90, 2014.

[6] Z. Yin, F. R. Yu, S. Bu, and Z. Han, "Joint cloud and wireless networks operations in mobile cloud computing environments with telecom operator cloud," *IEEE Trans. Wireless Commun.*, vol. 14, no. 7, pp. 4020–4033, Jul. 2015.

[7] Y. Cai, F. R. Yu, and S. Bu, "Dynamic operations of cloud radio access networks (C-RAN) for mobile cloud computing systems," *IEEE Trans. Veh. Tech.*, accepted for publication, DOI: 10.1109/TVT.2015.2411739.

[8] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network (SDN) and openflow: From concept to implementation," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2181–2206, 4th Quart. 2014.

[9] M. D. Yosr Jarraya and T. Madi, "A survey and a layered taxonomy of software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1955–1980, 4th Quart. 2014.

[10] W. Xia, Y. Wen, C. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart. 2015.

[11] S. Sezer *et al.*, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[12] S. Azodolmolky, P. Wieder, and R. Yahyapour, "SDN-based cloud computing networking," in *Proc. IEEE ICTON*, Jun. 2013, pp. 1–4.

[13] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 843–859, 2nd Quart. 2013.

[14] "The notorious nine cloud computing top threats in 2013," Cloud Security Alliance, Seattle, WA, USA, Tech. Rep., Feb. 2013. [Online]. Available: http://www.cloudsecurityalliance.org/topthreats

[15] D. Linthicum, "As cloud use grows, so will rate of DDoS attacks," InfoWorld, Framingham, MA, USA, Tech. Rep., Feb. 2013. [Online]. Available: http://www.infoworld.com/d/cloud-computing/cloud-use-grows-so-will-rate-of-ddos-attacks-211876

[16] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Comput. Netw.*, vol. 57, no. 2, pp. 378–403, Feb. 2013.

[17] Open Networking Foundation, Jun. 2014. [Online]. Available: https://www.opennetworking.org/

[18] M. Mendonca, B. A. A. Nunes, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart. 2014.

[19] Y. Cai, F. R. Yu, C. Liang, B. Sun, and Q. Yan, "Software defined device-to-device (D2D) communications in virtual wireless networks with imperfect network state information (NSI)," *IEEE Trans. Veh. Tech.*, accepted for publication, DOI: 10.1109/TVT.2015.2483558.

[20] List of Openflow Software Projects, Apr. 2013. [Online]. Available: http://yuba.stanford.edu/casado/of-sw.html

[21] R. Jin and B. Wang, "Malware detection for mobile devices using software-defined networking," in *Proc. IEEE 2nd GREE Workshop*, 2013, pp. 81–88.

[22] 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.

[23] T.-C. Yen and C.-S. Su, "An SDN-based cloud computing architecture and its mathematical model," in *Proc. IEEE Int. Conf. ISEEE*, Apr. 2014, vol. 3, pp. 1728–1731.

[24] I. Ku, Y. Lu, and M. Gerla, "Software-defined mobile cloud: Architecture, services and use cases," in *Proc. IEEE IWCMC*, Aug. 2014, pp. 1–6.

[25] H. Gharakheili, J. Bass, L. Exton, and V. Sivaraman, "Personalizing the home network experience using cloud-based SDN," in *Proc. IEEE Int. Symp. WoWMoM*, Jun. 2014, pp. 1–6.

[26] A. Akella and K. Xiong, "Quality of service (QoS)-guaranteed network resource allocation via software defined networking (SDN)," in *Proc. Int. Conf. DASC*, Aug. 2014, pp. 7–13.

[27] R. Cziva, D. Stapleton, F. P. Tso, and D. Pezaros, "SDN-based virtual machine management for cloud data centers," in *Proc. IEEE Int. Conf. CloudNet*, Oct. 2014, pp. 388–394.

[28] W.-C. Lin, C.-H. Liao, K.-T. Kuo, and C.-P. Wen, "Flow-and-VM migration for optimizing throughput and energy in SDN-based cloud datacenter," in *Proc. Int. Conf. CloudCom*, Dec. 2013, vol. 1, pp. 206–211.

[29] S. Seeber and G. Rodosek, "Improving network security through SDN in cloud scenarios," in *Proc. Int. CNSM*, Nov. 2014, pp. 376–381.

[30] M. Jarschel *et al.*, "Modeling and performance evaluation of an openflow architecture," in *Proc. 23rd ITC*, Sep. 2011, pp. 1–7.

[31] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2046–2069, 4th Quart. 2013.

[32] S. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 136–141, Feb. 2013.

[33] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in *Proc. IEEE SDN4FNS*, 2013, pp. 1–7.

[34] D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, 2013, pp. 55–60.

[35] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2046–2069, 4th Quart. 2013.

[36] S. Ranjan, R. Swaminathan, M. Uysal, and E. W. Knightly, "DDoS-resilient scheduling to counter application layer attacks under imperfect detection," in *Prof. IEEE INFOCOM*, Apr. 2006, pp. 1–13.

[37] "Riorey taxonomy of DDoS attacks," RioRey Inc., Bethesda, MD, USA, Tech. Rep. Rioreytaxonomy rev 2.3 2012, 2012. [Online]. Available: http://www.riorey.com/x-resources/2012/RioReyTaxonomyDDoSAttacks2012.eps

[38] W. Ren, "uleepp: An ultra-lightweight energy-efficient and privacy-protected scheme for pervasive and mobile wbsn-cloud communications," *Ad Hoc Sens. Wireless Netw.*, vol. 27, no. 3/4, pp. 173–195, 2015.

[39] "State of the internet report 2012 Q4," Akamai Technologies, Cambridge, MA, USA, Tech. Rep., Dec. 2012. [Online]. Available: http://www.slideshare.net/AkamaiTechnologies/q4-2012-sotiweb

[40] Jean-Francois, Cloud Computing: Weapon of Choice for DDoS?, Dec. 2012. [Online]. Available: http://www.orange-business.com/en/blogs/connecting-technology/security/cloud-computing-weapon-of-choice-for-ddos

[41] A. Gonsalves, Prices Fall, Services Rise in Malware-as-a-Service Market, Mar. 2013. [Online]. Available: http://www.csoonline.com/article/2133045/malware-cybercrime/prices-fall-services-rise-in-malware-as-a-service-market.html

[42] "Arbor special report: Worldwide infrastructure security report volume IX," Arbor Netw., Inc., Burlington, MA, USA, Tech. Rep.,

Dec. 2012. [Online]. Available: http://pages.arbornetworks.com/rs/arbor/images/WISR2012EN.pdf
[43] A. Gonsalves, Mobile Devices Set to Become Next DDoS Attack Tool, Jan. 2013. [Online]. Available: http://www.csoonline.com/article/2132699/mobile-security/mobile-devices-set-to-become-next-ddos-attack-tool.html
[44] "Arbor application brief: The growing threat of application-layer DDoS attacks," Arbor Netw., Inc., Burlington, MA, USA, Tech. Rep., Oct. 2010. [Online]. Available: http://www.arbornetworks.com/component/docman
[45] U. Ben-Porat, A. Bremler-Barr, and H. Levy, "Vulnerability of network mechanisms to sophisticated DDoS attacks," *IEEE Trans. Comput.*, vol. 62, no. 5, pp. 1031–1043, May 2013.
[46] A. Girma, M. Garuba, and R. Goel, "Cloud computing vulnerability: DDoS as its main security threat, and analysis of IDS as a solution model," in *Proc. 11th Int. Conf. ITNG*, 2014, pp. 307–312.
[47] R. Shea and J. Liu, "Performance of virtual machines under networked denial of service attacks: Experiments and analysis," *IEEE Syst. J.*, vol. 7, no. 2, pp. 335–345, Jun. 2013.
[48] T. Lohman, DDoS is Cloud's Security Achilles Heel, Sep. 2011. [Online]. Available: http://www.computerworld.com.au/article/401127
[49] H.-Y. Tsai, M. Siebenhaar, A. Miede, Y. Huang, and R. Steinmetz, "Threat as a service?: Virtualization's impact on cloud security," *IT Professional*, vol. 14, no. 1, pp. 32–37, Jan. 2012.
[50] M. Godfrey and M. Zulkernine, "Preventing cache-based side-channel attacks in a cloud environment," *IEEE Trans. Cloud Comput.*, vol. 2, no. 4, pp. 395–408, Oct. 2014.
[51] M. H. Sqalli, F. Al-Haidari, and K. Salah, "EDoS-shield-a two-steps mitigation technique against EDoS attacks in cloud computing," in *Proc. 4th IEEE Int. Conf. UCC*, 2011, pp. 49–56.
[52] S. VivinSandar and S. Shenai, "Economic denial of sustainability (EDoS) in cloud services using http and xml based DDoS attacks," *Int. J. Comput. Appl.*, vol. 41, no. 20, pp. 11–16, Mar. 2012.
[53] J. Idziorek, M. Tannian, and D. Jacobson, "Attribution of fraudulent resource consumption in the cloud," in *Proc. IEEE 5th Int. CLOUD*, 2012, pp. 99–106.
[54] S. Farahmandian *et al.*, "A survey on methods to defend against DDoS attack in cloud computing," in *Proc. Recent Adv. Knowl. Eng. Syst. Sci.*, Feb. 2013, pp. 185–190.
[55] P. Ferguson, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," Internet Eng. Task Force (IETF), Fremont, CA, USA, Internet RFC 3704, 2000.
[56] J. Li *et al.*, "SAVE: Source address validity enforcement protocol," in *Proc. IEEE INFOCOM*, 2002, vol. 3, pp. 1557–1566.
[57] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surveys*, vol. 39, no. 1, pp. 1–42, Apr. 2007.
[58] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 15–26, Aug. 2001.
[59] R. Stone *et al.*, "Centertrack: An IP overlay network for tracking DoS floods," in *Proc. USENIX Security Symp.*, 2000, vol. 21, pp. 114–128.
[60] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," *SIGCOMM Comput. Commun. Rev*, vol. 30, no. 4, pp. 295–306, Aug. 2000.
[61] A. C. Snoeren *et al.*, "Hash-based IP traceback," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 3–14, 2001.
[62] K. Argyraki and D. R. Cheriton, "Scalable network-layer defense against Internet bandwidth-flooding attacks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1284–1297, Apr. 2009.
[63] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, and E. Knightly, "DDoS-shield: DDoS-resilient scheduling to counter application layer attacks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 26–39, Feb. 2009.
[64] A. Kolupaev and J. Ogijenko, "Captchas: Humans vs. bots," *IEEE Security Privacy*, vol. 6, no. 1, pp. 68–70, Jan. 2008.
[65] R. Datta, J. Li, and J. Wang, "Exploiting the human-machine gap in image recognition for designing captchas," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 3, pp. 504–518, Sep. 2009.
[66] M. Naresh Kumar *et al.*, "Mitigating economic denial of sustainability (EDoS) in cloud computing using in-cloud scrubber service," in *Proc. IEEE 4th Int. Conf. CICN*, 2012, pp. 535–539.
[67] S. Yu, Y. Tian, S. Guo, and D. Wu, "Can we beat DDoS attacks in clouds?" *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2245–2254, Sep. 2014.
[68] R. Lua and K. C. Yow, "Mitigating DDoS attacks with transparent and intelligent fast-flux swarm network," *IEEE Netw.*, vol. 25, no. 4, pp. 28–33, Apr. 2011.
[69] Z. Chen, F. Han, J. Cao, X. Jiang, and S. Chen, "Cloud computing-based forensic analysis for collaborative network security management system," *Tsinghua Sci. Technol.*, vol. 18, no. 1, pp. 40–50, Jan. 2013.
[70] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 52–59, Apr. 2015.
[71] S. Jajodia *et al.*, *Secure Cloud Computing*. New York, NY, USA: Springer-Verlag, 2014.
[72] S. Bu, F. R. Yu, X. P. Liu, and H. Tang, "Structural results for combined continuous user authentication and intrusion detection in high security mobile ad-hoc networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 9, pp. 3064–3073, Sep. 2011.
[73] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Proc. Recent Adv. Intrusion Detect.*, 2011, pp. 161–180.
[74] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with openflow/nox architecture," in *Proc. 19th IEEE ICNP*, 2011, pp. 7–12.
[75] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using nox/openflow," in *Proc. 35th IEEE Conf. LCN*, 2010, pp. 408–415.
[76] H. C. J. Suh, T. Y. W. Yoon, T. Kwon, and Y. Choi, "Implementation of a content-oriented networking architecture (CONA): A focus on DDoS countermeasure," in *Prof. Eur. NetFPGA Develop. Workshop*, 2010, pp. 1–5.
[77] C. YuHunag, T. MinChi, C. YaoTing, C. YuChieh, and C. YanRen, "A novel design for future on-demand service and security," in *Proc. 12th IEEE Int. Conf. Commun. Technol.*, 2010, pp. 385–388.
[78] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Comput. Netw.*, vol. 62, pp. 122–136, 2014.
[79] T. Xing, D. Huang, L. Xu, C.-J. Chung, and P. Khatkar, "Snortflow: A openflow-based intrusion prevention system in cloud environment," in *Proc. IEEE 2nd GREE Workshop*, 2013, pp. 89–92.
[80] S. Shin *et al.*, "Fresco: Modular composable security services for software-defined networks," in *Proc. ISOC Netw. Distrib. Syst. Security Symp.*, 2013, pp. 1–16.
[81] A. Passito, E. Mota, R. Bennesby, and P. Fonseca, "AgNOS: A framework for autonomous control of software-defined networks," in *Proc. 28th IEEE Int. Conf. AINA*, 2014, pp. 405–412.
[82] Y. Yu, Q. Chen, and X. Li, "Distributed collaborative monitoring in software defined networks," in *Proc. HotSDN*, 2014, pp. 85–90.
[83] A. TaheriMonfared and C. Rong, "Multi-tenant network monitoring based on software defined networking," in *Proc. OTM Conf. Move Meaningful Internet Syst.*, 2013, pp. 327–341.
[84] A. Chesla and E. Doron, "Techniques for traffic diversion in software defined networks for mitigating denial of service attacks," U.S. Patent App. 13/913 916, Jun. 10, 2013.
[85] DDoS Protection as an SDN Network Service, Jun. 2014. [Online]. Available: http://www.radware.com/
[86] H. Tian and J. Bi, "An incrementally deployable flow-based scheme for IP traceback," *IEEE Commun. Lett.*, vol. 16, no. 7, pp. 1140–1143, Jul. 2012.
[87] N. Handigol, B. Heller, V. Jeyakumar, D. Mazieres, and N. McKeown, "I know what your packet did last hop: Using packet histories to troubleshoot networks," in *Proc. Symp. NSDI*, 2014, pp. 71–85.
[88] A. Wundsam *et al.*, "OFRewind: Enabling record and replay troubleshooting for networks," in *Proc. USENIX Annu. Tech. Conf.*, 2011, pp. 1–39.
[89] N. Handigol, B. Heller, V. Jeyakumar, D. Mazières, and N. McKeown, "Where is the debugger for my software-defined network?" in *Proc. ACM 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 55–60.
[90] S. Shin and G. Gu, "Attacking software-defined networks: A first feasibility study," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, 2013, pp. 165–166.
[91] S. M. Mousavi, "Early detection of DDoS attacks in software defined networks controller," M.S. thesis, Dept. Syst. Comput. Eng., Carleton Univ., Ottawa, ON, USA, 2014.

[92] P. Porras *et al.*, "A security enforcement kernel for openflow networks," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 121–126.
[93] K. Benton, L. J. Camp, and C. Small, "Openflow vulnerability assessment," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, 2013, pp. 151–152.
[94] J. M. Dover, "A denial of service attack against the open floodlight SDN controller," Dover Netw., Edgewater, MD, USA.
[95] M. Liyanage, M. Ylianttila, and A. Gurtov, "Securing the control channel of software-defined mobile networks," in *Proc. 1st IEEE WoWMoM Workshop Softw. Defined Netw. Archit. Appl.*, Sydney, NSW, Australia, Jun. 2014, pp. 1–6.
[96] T. Koponen *et al.*, "Onix: A distributed control platform for large-scale production networks," in *Proc. OSDI*, 2010, vol. 10, pp. 1–6.
[97] H. Li, P. Li, S. Guo, and S. Yu, "Byzantine-resilient secure software-defined networks with multiple controllers," in *Proc. IEEE ICC*, Jun. 2014, pp. 695–700.
[98] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Avant-guard: Scalable and vigilant switch flow management in software-defined networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2013, pp. 413–424.
[99] R. Kloti, V. Kotronis, and P. Smith, "Openflow: A security analysis," in *Proc. 21st IEEE ICNP*, Oct. 2013, pp. 1–6.
[100] A. Y. Ding, J. Crowcroft, S. Tarkoma, and H. Flinck, "Software defined networking for security enhancement in wireless mobile networks," *Comput. Netw.*, vol. 66, pp. 94–101, May 2014.
[101] Gartner: Application Layer DDoS Attacks to Increase in 2013, Mar. 2013. [Online]. Available: http://www.securitybistro.com/?p=5493/
[102] SDN and Security: Network Versus Applications, Jan. 2014. [Online]. Available: https://devcentral.f5.com/articles/sdn-and-security-network-versus-applications
[103] How SDN Applications Will Change Layer 4–7 Network Services, Apr. 2013. [Online]. Available: http://searchsdn.techtarget.com/tip/How-SDN-applications-will-change-Layer-4-7-network-services
[104] G. Finnie, "The role of DPI in an SDN world," QOSMOS, Paris, France, Tech. Rep., Dec. 2012. [Online]. Available: www.qosmos.com
[105] Mobile applications being used for DDoS Attacks According to Prolexic's Latest Quarterly Report, Jan. 2014. [Online]. Available: http://www.prolexic.com/news-events-pr-mobile-apps-applications-being-used-for-ddos-attacks-q4-2013-ddos-attack-report.html
[106] B. Sun, F. Yu, K. Wu, Y. Xiao, and V. Leung, "Enhancing security using mobility-based anomaly detection in cellular mobile networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 4, pp. 1385–1396, Jul. 2006.
[107] F. R. Yu, H. Tang, P. Mason, and F. Wang, "A hierarchical identity based key management scheme in tactical mobile ad hoc networks," *IEEE Trans. Netw. Service Manage.*, vol. 7, no. 4, pp. 258–267, Dec. 2010.
[108] S. Bu, F. R. Yu, P. Liu, P. Manson, and H. Tang, "Distributed combined authentication and intrusion detection with data fusion in high-security mobile ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1025–1036, Mar. 2011.
[109] Z. Wei, H. Tang, F. R. Yu, M. Wang, and P. Mason, "Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4647–4658, Nov. 2014.
[110] J. Lee *et al.*, "meSDN: Mobile extension of SDN," in *Proc. 5th Int. Workshop Mobile Cloud Comput. Services*, 2014, pp. 7–14.
[111] J. Giacomoni, "Extending SDN architectures with F5's L4-7 application and gateway services," F5 Networks, Inc., Seattle, WA, USA, , Tech. Rep., Dec. 2013. [Online]. Available: https://www.f5.com/pdf/white-papers/extending-sdn-architectures-with-f5-l4-7-application-and-gateway-services-white-paper.pdf
[112] I. Gashi and O. P. Kreidl, "6th workshop on recent advances in intrusion tolerance and resilience (WRAITS 2012)," in *Proc. 42nd IEEE/IFIP Int. Conf. DSN-W*, Jun. 2012, pp. 1–2.
[113] C. Bernardeschi, Intrusion Tolerance, Nov. 2011. [Online]. Available: http://www.iet.unipi.it/g.dini/Teaching/ssi/materiale-didattico/Intrusion_Tolerance.pdf
[114] M. Garcia, A. Bessani, I. Gashi, N. Neves, and R. Obelheiro, "Os diversity for intrusion tolerance: Myth or reality?" in *Proc. 41st IEEE/IFIP Int. Conf. DSN*, Jun. 2011, pp. 383–394.
[115] A. Bessani, "From byzantine fault tolerance to intrusion tolerance (a position paper)," in *Proc. 41st IEEE/IFIP Int. Conf. DSN-W*, Jun. 2011, pp. 15–18.
[116] D. Tariq and T. Mahmood, "Security analytics: Big data analytics for cybersecurity," in *Proc. 2nd NCIA*, 2013, pp. 129–134.
[117] L. Lan and L. Jun, "Some special issues of network security monitoring on big data environments," in *Proc. 11th IEEE Int. Conf. Dependable, Auton. Secure Comput.*, 2013, pp. 10–15.
[118] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forests," *Inf. Sci.*, vol. 278, pp. 488–497, Sep. 2014.
[119] P. Raj, *Handbook of Research on Cloud Infrastructures for Big Data Analytics*. Hershey, PA, USA: IGI Global, 2014.
[120] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 47, no. 7, pp. 20–26, Jul. 2009.
[121] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart. 2015.
[122] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 20–27, Mar./Apr. 2013.
[123] C. Liang and F. R. Yu, "Wireless virtualization for next generation mobile cellular networks," *IEEE Wireless Comm.*, vol. 22, no. 1, pp. 61–69, Feb. 2015.
[124] A. Wang, M. Iyer, R. Dutta, G. N. Rouskas, and I. Baldine, "Network virtualization: Technologies, perspectives, and frontiers," *J. Lightw. Technol.*, vol. 31, no. 4, pp. 523–537, Feb. 2013.
[125] A. Ooka, S. Ata, T. Koide, H. Shimonishi, and M. Murata, "Openflow-based content-centric networking architecture and router implementation," in *Proc. FutureNetworkSummit*, 2013, pp. 1–10.
[126] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *IEEE Comm. Mag.*, vol. 52, no. 11, pp. 122–129, Nov. 2014.
[127] C. Liang, F. R. Yu, and X. Zhang, "Information-centric network function virtualization over 5G mobile wireless networks," *IEEE Netw.*, vol. 29, no. 3, pp. 68–74, May 2015.
[128] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart. 2015.
[129] L. Zhang *et al.*, "Named Data Networking (NDN) project," Palo Alto Research Center (PARC), Palo Alto, CA, USA, Tech. Rep. NDN-0001, PARC, 2010. [Online]. Available: http://www.named-data.net/techreport/TR001ndn-proj.pdf
[130] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos, "Developing information networking further: From PSIRP to PURSUIT," in *Broadband Communications, Networks, and Systems*, vol. 66, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. New York, NY, USA: Springer-Verlag, 2012, pp. 1–13.
[131] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named data networking," in *Proc. 22nd IEEE ICCCN*, 2013, pp. 1–7.

Qiao Yan received the Ph.D. degree in information and communication engineering from Xidian University, Xi'an, China, in 2003.

She is a Professor in the College of Computer Science and Software Engineering at Shenzhen University, Shenzhen, China. From 2004 to 2005 she was with Tsinghua University, Beijing, China, as a PostDoc Fellow. From 2013 to 2014, she was with Carleton University, Ottawa, ON, Canada, as a Visiting Scholar. Her research interests are in network security, cloud computing, and software-defined networking. Her current focus is research and development of security of software-defined networking.

F. Richard Yu (S'00–M'04–SM'08) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2003.

From 2002 to 2004, he was with Ericsson, Lund, Sweden, where he worked on the research and development of wireless mobile systems. From 2005 to 2006, he was with a start-up in California, USA, where he worked on the research and development in the areas of advanced wireless communication technologies and new standards. He joined Carleton School of Information Technology and the Department of Systems and Computer Engineering at Carleton University, Ottawa, ON, Canada, in 2007, where he is currently an Associate Professor. His research interests include cross-layer/cross-system design, security, green IT, and QoS provisioning in wireless-based systems.

Prof. Yu received the IEEE Outstanding Leadership Award in 2013, Carleton Research Achievement Award in 2012, the Ontario Early Researcher Award (formerly Premier's Research Excellence Award) in 2011, the Excellent Contribution Award at IEEE/IFIP TrustCom 2010, the Leadership Opportunity Fund Award from Canada Foundation of Innovation in 2009, and the Best Paper Awards at IEEE ICC 2014, Globecom 2012, IEEE/IFIP TrustCom 2009, and International Conference on Networking 2005. He serves on the editorial boards of several journals, including Co-Editor-in-Chief for Ad Hoc & Sensor Wireless Networks, Lead Series Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, *EURASIP Journal on Wireless Communications Networking*, *Journal on Security and Communication Networks*, and *International Journal of Wireless Communications and Networking*, a Guest Editor for IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING special issue on Advances in Mobile Cloud Computing, and a Guest Editor for IEEE SYSTEMS JOURNAL for the special issue on Smart Grid Communications Systems. He has served on the Technical Program Committee (TPC) of numerous conferences, as the TPC Co-Chair of IEEE GreenCom'15, INFOCOM-MCV'15, Globecom'14, WiVEC'14, INFOCOM-MCC'14, Globecom'13, GreenCom'13, CCNC'13, INFOCOM-CCSES'12, ICC-GCN'12, VTC'12S, Globecom'11, INFOCOM-GCN'11, INFOCOM-CWCN'10, IEEE IWCMC'09, VTC'08F, and WiN-ITS'07, as the Publication Chair of ICST QShine'10, and the Co-Chair of ICUMT-CWCN'09. He is a registered Professional Engineer in the province of Ontario, Canada.

Qingxiang Gong is pursuing the master's degree in the College of Computer Science and Software Engineering at Shenzhen University, Shenzhen, China.

His research focuses on software define network and DDoS.

Jianqiang Li received the B.S. and Ph.D. degrees from South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively. He is an Associate Professor in the College of Computer and Software Engineering at Shenzhen University, Shenzhen, China. He is leading two projects funded by the National Natural Science Foundation of China and two projects funded by the Natural Science Foundation of Guangdong province, China. His major research interests include Internet of thing, robotic, hybrid systems, and embedded systems.