DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions

Article · February 2017

DOI: 10.1007/s13369-017-2414-5

3 authors:

Narmeen Bawany
Jinnah University for Women
19 PUBLICATIONS **93** CITATIONS

SEE PROFILE

Jawwad Shamsi
National University of Computer and Emerging Sciences, Karchi, Pakistan
64 PUBLICATIONS **313** CITATIONS

SEE PROFILE

Khaled Salah
Khalifa University
211 PUBLICATIONS **1,616** CITATIONS

SEE PROFILE

Arab J Sci Eng
DOI 10.1007/s13369-017-2414-5

REVIEW ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions

Narmeen Zakaria Bawany[1] · **Jawwad A. Shamsi**[1] · **Khaled Salah**[2]

Received: 20 August 2016 / Accepted: 9 January 2017

Abstract Distributed denial-of-service (DDoS) attacks have become a weapon of choice for hackers, cyber extortionists, and cyber terrorists. These attacks can swiftly incapacitate a victim, causing huge revenue losses. Despite the large number of traditional mitigation solutions that exists today, DDoS attacks continue to grow in frequency, volume, and severity. This calls for a new network paradigm to address the requirements of today's challenging security threats. Software-defined networking (SDN) is an emerging network paradigm which has gained significant traction by many researchers to address the requirement of today's data centers. Inspired by the capabilities of SDN, we present a comprehensive survey of existing SDN-based DDoS attack detection and mitigation solutions. We classify solutions based on DDoS attack detection techniques and identify requirements of an effective solution. Based on our findings, we propose a novel framework for detection and mitigation of DDoS attacks in a large-scale network which comprises a smart city built on SDN infrastructure. Our proposed framework is capable of meeting application-specific DDoS attack detection and mitigation requirements. The primary contribution of this paper is twofold. First, we provide an in-depth survey and discussion of SDN-based DDoS attack detection and mitigation mechanisms, and we classify them with respect to the detection techniques. Second, leveraging the characteristics of SDN for network security, we propose and present an SDN-based proactive DDoS Defense Framework (ProDefense). We show how this framework can be utilized to secure applications built for smart cities. Moreover, the paper highlights open research challenges, future research directions, and recommendations related to SDN-based DDoS detection and mitigation.

Keywords Software-defined networking · SDN · DDoS attacks · OpenFlow · DDoS mitigation

✉ Khaled Salah
khaled.salah@kustar.ac.ae

Narmeen Zakaria Bawany
nshawoo@gmail.com

Jawwad A. Shamsi
jawwad.shamsi@nu.edu.pk

[1] Systems Research Laboratory, FAST-National University of Computer and Emerging Sciences, Karachi, Pakistan

[2] Electrical and Computer Engineering Department, Khalifa University of Science, Technology and Research, PO Box 573, Sharjah, UAE

1 Introduction

Distributed denial-of-service (DDoS) attacks have been a real threat for network, digital, and cyber infrastructure [1].These attacks are capable to cause massive disruption in any information communication technology (ICT) infrastructure [2]. There could be numerous reasons for launching DDoS attacks. These include financial gains [3], political gains [2], and disruption [4,5]. DDoS attacks can paralyze networks and services by overwhelming servers, network links, and network devices (routers, switches, etc.) with illegitimate traffic. They can either cause degradation of service or a complete denial of service resulting in huge losses. Increasing reliance on Internet and data centers has aggravated this problem. The growing dependence of critical infrastructure of a country in ICT have given rise to the need of efficient solutions for protection against DDoS attacks [6,7]. For instance, data centers running critical services, such as smart grid, need to be protected in order to continue to provide highly reliable services.

Published online: 02 February 2017

Numerous proprietary and open-source solutions exist for DDoS attack detection and mitigation. However, these attacks continue to grow in frequency, sophistication, and severity [8,9]. Rapid detection and mitigation of DDoS attacks has become severely challenging as attackers continue to use novel techniques to launch DDoS attacks [10]. The rising number of DDoS attacks, coupled with growing diversity in their types, causing disastrous impact, has made DDoS attack detection, mitigation, and prevention the top most priority.

For instance, Arbor Networks Inc. [11], one of the leading DDoS threat protection solutions provider, reported a 334 Gbps attack targeting a network operator in Asia recently. Also, it reported many attacks larger than 100 Gbps globally in 2015 [12]. Many such incidents clearly shows that we need new approaches to address the DDoS attack problem. These new approaches must be designed to meet the performance and scalability requirements of modern data centers and provide maximum levels of protection against emerging, complex and elusive, attacks.

With recent advancements in software-defined networking (SDN) and its rapid and wide-scale acceptance in the network community [13], many researchers have been actively involved in developing SDN-based network security solutions. SDN-based solutions have attracted more attention since their adoption in large-scale wide area networks [13]. The technology enables developers to directly program, control, and manage network resources centrally through the SDN controller.

SDN offers novel ways to solve long standing networking problems, such as routing [14], policy-based network configurations, and security as discussed in Sect. 3 While security of SDN-based networks has been a point of debate and much literature is available discussing the security of SDN infrastructure itself [15–17]. This paper, however, takes the positive viewpoint on SDN-based security and presents a survey of SDN-based DDoS attack detection and mitigation mechanisms.

During the study of existing SDN-based solutions, we observed that there are many approaches for SDN-based DDoS attack detection. Based on this study, we categorized the existing approaches according to their methods of anomaly detection. Further, we noted that there is a need for a proficient DDoS protection framework that can be customized with respect to requirements of various applications. This is specifically true for network infrastructures consisting of heterogeneous applications. For instance, in a smart city, there are multiple applications which require protection from DDoS attacks. Each of these applications has a different level of tolerance of network traffic. In addition, each application may have different responsiveness requirements as well. Modern data centers also have needs for scalable and distributed controllers in order to increase reliability and balance load in data centers. Motivated by these needs, we propose an SDN-based proactive DDoS Defense Framework (ProDefense) for smart city data center.

The main contributions of this paper are summarized as follows:

- We present an extensive survey of SDN-based DDoS attack detection techniques. We classify these techniques according to detection mechanisms. Classification allows better understanding and improved comprehension of the existing approaches.
- We identify pros and cons of each technique and elaborate key requirements of an effective DDoS attack prevention mechanism.
- We propose a novel SDN-based proactive DDoS Defense Framework (ProDefense) for smart city data center. ProDefense allows implementation of application-specific requirements for DDoS attack detection and mitigation. ProDefense also has distributed controllers, thereby allowing effective mechanisms for distributing load and improving reliability.
- We present a case study showing how ProDefense capabilities can be utilized to secure applications built for smart cities.

Our work is significant with multiple benefits. For researchers, it provides a comprehensive analysis of the existing work and identifies challenges, whereas for academicians, it offers a thorough study of the subject. Our work is also useful for the developer community in understanding strengths and weaknesses of different solutions. The industrial community could also find our work useful in understanding the requirements and assessing capabilities of these solutions.

The rest of the paper is organized as follows. In Sect. 2, we present a brief overview of DDoS attacks and SDN. We categorize the existing SDN-based DDoS attack detection techniques and present a survey of these techniques in Sect. 3. This is followed by description of DDoS attack mitigation strategies in Sect. 4. Further, in Sect. 5, we describe ProDefense which is our proposed framework for detection and mitigation of DDoS attack in a smart city data center. In Sect. 6, we discuss SDN challenges and open research areas. Section 7 concludes the paper and presents the future research directions.

2 Background

The growing prevalence of DDoS attacks show that legacy defense mechanisms are only partially effective. List of recent DDoS attacks on various recognized organizations is provided in Table 1. These attacks are targeting almost every organization. Even leading financial institutions and

Table 1 Today's popular DDoS attacks

Target	Description
Client of US-based security vendor Sucuri [84]. June 2016	US-based security vendor Sucuri [85] discovered a botnet comprising of CCTV-based botnet used for DDoS attacks. The botnet was generating about 50,000 HTTP requests per second to the server and occupying its memory with illegitimate traffic
Bank of Greece Web site [86]. May 2016	Series of DDoS attack on Bank of Greece website forced the servers to remain offline for more than 6 h
HSBC internet banking [87]. January 2016	DDoS attack on HSBC [88], one of the largest banking and financial services organizations in the world, forced its personal banking Web sites in the UK to shut down for many hours
Irish government websites [89]. January 2016	Several Irish government websites were temporarily forced offline by an apparent DDoS attack. The affected Web sites included Central Statistics Office, the Courts Service, the Health Service Executive, and the Houses of the Oireachtas (parliament)
BBC websites [90,91]. December 2015	The attack on the BBC targeted the main Web site as well as associated services including the main iPlayer catch-up service and iPlayer Radio app
Thai government websites [92]. October 2015	Several Thai government Web sites have been hit by a suspected distributed denial-of-service (DDoS) attack, making them impossible to access. Web sites that were targeted includes site of the ministry of information, communications and technology (ICT) and the main government Web site
Polish airline [93]. June 2015	Around 1400 passengers were stranded at Warsaw's Chopin airport when the flight plan system went down for around 5 hours. The problem was caused DDoS attack—that generated so many communication requests that it overloaded the server, and it could no longer carry out its normal functions
Canadian Government Web sites [94]. June 2015	DDoS attack on Canadian government's computer servers caused federal emails and several department websites to shutdown
GitHub [90]. March 2015	GitHub, a company that hosts programming repositories, was slammed by DDoS attack. GitHub's servers were struck with thousands of illegitimate requests causing intermittent outages
Client of CloudFlare [25,95]. February 2014	CloudFlare [96], reported an attack reaching to 400 Gbps against one of its customers. The attackers leveraged a flaw in the network time protocol (NTP) to launch the DDoS attack
Spamhaus [97]. March 2013	Spam-fighting organization, Spamhaus has been hit by DDoS attack of 300 Gbps

government organizations, having huge IT infrastructure and resources, are unable to encounter such attacks It seems necessary to explore new paradigms that can successfully respond to DDoS attacks. SDN has emerged as a potential candidate for addressing the escalating problem of DDoS attacks. The purpose of this section is to enlighten background information. First, we briefly describe DDoS attacks. This is followed by a description of an effective DDoS attack defensive mechanism. The section also explains the architecture and protocols of SDN.

2.1 Distributed Denial-of-Service (DDoS) Attacks

The objective of a DDoS attack is to bring down the services of a target using multiple sources that are distributed. A typical example of such an attack is the flooding based attack in which a victim is overwhelmed with the massive amount of network traffic sent to it. The idea of DDoS attacks revolves around a fact that a large number of sources distributed across multiple locations are used to target a victim. Botnets are typically useful for launching DDoS attacks as a large collection of compromised hosts (also called zombies) are typically available. Table 2 classifies popular DDoS attacks into three categories.

Prevention against DDoS attacks has been a major focus of the research community [10,18]. Since DDoS attacks are quite frequent, the focus has been to develop a proficient solution, which is capable to effectively detect and mitigate DDoS attacks. In our research, we identify following requirements for an effective DDoS mechanism:

(a) The DDoS protection mechanism must not disturb or disrupt legitimate user's activities
(b) The DDoS detection mechanism should be able to prevent attacks from within the network as well as from outside the network.
(c) The mechanism should meet the performance and scalability requirements of modern data centers.
(d) The DDoS protection mechanism should be inexpensive with low deployment cost such as additional hardware. In addition, the deployment must not require wide-scale changes leading to high overhead.

Table 2 Classification of DDoS attacks

I. Reflection-based flooding attacks	
Smurf Attack [98]	In Smurf Attack, large number of Internet Control Message Protocol (ICMP) packets with the intended victim's forged source IP are broadcast to a computer network using an IP Broadcast address. This generates huge amount of illegitimate traffic on the network causing network congestion
Fraggle Attack [98]	Fraggle attack is similar to Smurf attack, but it uses illegitimate UDP traffic instead of ICMP traffic to achieve the same goal
II. Protocol exploitation flooding attacks	
SYN Flooding attack [98]	In TCP SYN flood attack, an attacker sends the packet with the SYN bit setoff TCP three-way handshake. The victim responds with the packet back to the source address with SYN-ACK bit set. The attacker never responds to the reply packet, either on purpose or because the source address of the packet is forged. Therefore, the victim's TCP receive queues would be filled up, denying new TCP connections to legitimate clients
UDP Fragmentation attack [99]	In a UDP Fragmentation attack, attackers send large UDP packets (1500+ bytes) to consume more bandwidth with fewer packets. The victim's resources are consumed in reassembling these forged packets which no ability to be re-assembled
III. Reflection and amplification-based flooding attacks	
DNS amplification Flooding attack [100]	In domain name system (DNS) amplification flooding attack, the zombies generate small DNS queries with forged source IP address that generates large volume of network traffic directed toward the victim
NTP amplification Flooding attack [101]	Network Time Protocol (NTP) amplification attack is similar to DNS amplification attack, but it uses NTP servers instead of DNS servers

(e) The mechanism should be robust, adaptive, and flexible.
(f) Mitigation support should be provided.
(g) Low false-positive and high detection rates are also desirable characteristic for an effective DDoS protection mechanism.

The requirements mentioned above can only be considered as guidelines to develop a good DDoS attack protection solution. Severity of each requirements may vary with respect to network environment in which it is implemented and the level of protection required. However, there is no silver bullet to eliminate DDoS attacks. Attaining impeccable DDoS attack prevention mechanism presents many challenges mainly considering the growing complexity of attacks. Few of the challenges that security researchers have to consider can be listed as follows:

(a) Differentiating between legitimate and illegitimate traffic is much difficult due to attacks that mimic the behavior of legitimate network traffic. For instance, sophisticated botnets are capable of bypassing the DDoS attack detectors by mimicking the traffic patterns of flash crowds [19].
(b) Internal network threats are also growing in an organization along with external threats. Internal system weaknesses include software vulnerabilities, use of portable devices, and human weaknesses. Network security devices that are placed at the edge of network can do least for these internal threats.
(c) Emerging trends in networking including cloud and big data, are urging for high bandwidth, high reliability, ubiquitous accessibility, and dynamic management in data centers [20]. Ever growing demand and high accessibility requirements of data makes performance and scalability a critical challenge.
(d) Maintaining the low false-positive rate may lead to low detection rate, such that slow and steady DDoS attacks [21] may easily bypass the DDoS attack prevention solutions.
(e) Developing an effective open-source DDoS attack protection solution that can compete with the proprietary solutions is challenging because of lack of commercial support [22] along with integration issues with proprietary hardware and software which makes it more complicated.

Current DDoS trends are a clear indication of limitations of existing technologies. Large-scale volumetric attacks are further increasing in size and more sophisticated DDoS attacks are emerging [23–25]. Attackers have adopted sophisticated mechanisms to bypass the traditional protection shields. More recently, software-defined networks (SDN)

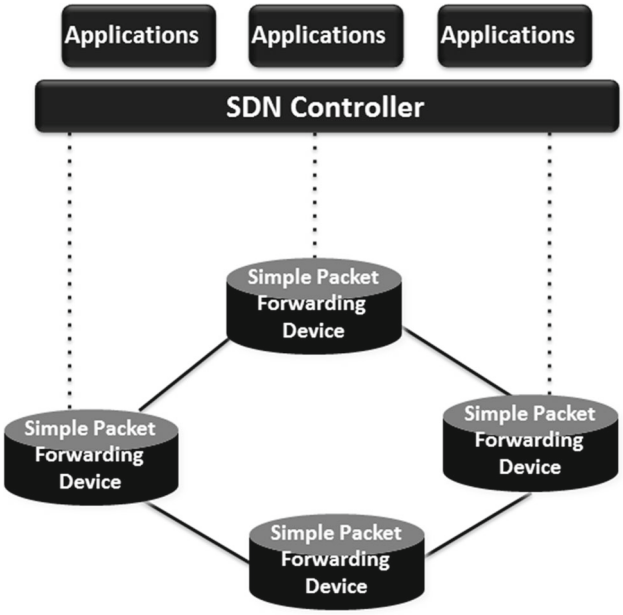

Fig. 1 SDN architecture

have emerged as new networking paradigm which have received wide-scale attraction. Among the most exciting features of SDN are the unprecedented control over network infrastructure and decoupling of control and data plane. These distinct characteristics of SDN have led to development of many SDN-based DDoS attack detection and mitigation mechanisms as discussed in the subsequent section of the paper. We briefly explain the mechanism of SDN.

2.2 Software-Defined Networking (SDN)

Software-Defined Networking simplifies network management by separating control logic (control plane) from the underlying hardware that forwards the traffic (data plane). With this decoupling of control plane and data plane, network switches become simple forwarding devices whereas control logic and functionality are implemented in logically centralized controller. Figure 1 illustrates the basic architecture of SDN. There are numerous open-source SDN controllers such as POX, NOX, Floodlight and OpenDaylight [26,27]. It is pertinent to mention that a logically centralized controller in SDN does not necessarily imply a physically centralized system [26]. In fact, many SDN frameworks, support distributed frameworks such as ONOS [28] and OpendayLight [29]. Efficacy of SDN has been established as it has been adopted by organizations, such as Google to manage Wide Area Network [13]. Distinguishing features of SDN makes it a strong candidate for next-generation Internet architecture [18].

OpenFlow [30] protocol is the first and most widely deployed protocol on SDN. The OpenFlow protocol defines the communication mechanism that enables the SDN controller to directly interact with the data plane. Controller pushes packet handling rules in flow tables of OpenFlow switches. The rule matches the traffic and performs certain actions, such as dropping, forwarding, modifying on traffic. Depending upon the rules installed by a controller application, an OpenFlow switch can take the role of a router, switch, firewall, load balancer, etc. [26,31].

SDN is anticipated to be an ideal platform that could contribute largely in developing pragmatic solutions for DDoS attack detection and mitigation. The notion of separation of control and data planes, and the concept of flow-based traffic make detection of attacks much easier. Further, the central point of control lead to rapid detection [32].

Research community has sought to utilize the distinctive features of SDN in order to enhance security against traditional cyber attacks, including DDoS attacks. Though, the SDN paradigm provides capabilities to develop effective measures to encounter DDoS attack, it also possess the risk of vulnerabilities and malfunctions due to attacks on the controller. These include code injection attack, man in the middle attack, and denial-of-service attack on the controller. Efficacy of SDN-based networks can be compromised due to these attacks [32]. However, our focus in this paper is to enlighten security enhancement features using SDN. In this context, we present a detail review on existing SDN-based DDoS attack detection and mitigation solutions. We also present the classification of DDoS attack detection and mitigation solutions based on detection techniques.

3 SDN-Based DDoS Attack Detection Techniques

SDN has many distinctive features which are key in detecting and mitigation DDoS attacks. These features include separation of the control plane from the data plane, logically centralized controller, programmability of the network by external applications, software-based traffic analysis, and capability to dynamically update forwarding rules [33]. We describe DDoS attack detection and mitigation techniques using SDN in this section. A summary of popular SDN-based DDoS attack detection techniques is given in Table 3.

3.1 Entropy

Entropy is used to measure the randomness of an attribute within a given period of time. Entropy-based techniques have been established as an effective approach to compute the randomness of a dataset [34]. High entropy values signify a more dispersed probability distribution, while low entropy values denote concentration of a distribution. Therefore, these techniques are used for anomaly detection extensively in traditional intrusion detection systems [35–37]. This approach

Table 3 Summary of popular SDN-based DDoS attack detection techniques

Techniques	Description
Entropy [34,41,42]	Entropy-based methods depend on network feature distributions to detect anomalous network activities. Probability distributions of various network features such as source IP address, destination IP address, and port numbers are used to calculate the entropy. Predefined thresholds on changes in the entropy values are used to identify the presence of anomolies
Machine learning [55,56,58,59]	Machine learning-based methods employ techniques such as Bayesian networks, SOM, and fuzzy logic to identify the presence of anomolies. These algorithms takes into account various network features and traffic characteristics to detect the presence of anoamlies
Traffic pattern analysis [61,63]	These techniques work on the assumptions that the infected hosts exhibit similar behavioral patterns which are different from benign hosts. Typically, in case of a botnet attack, infected machines (bots) are usually controlled by single bot master. Similar traffic patterns are observed as a result of command that is sent to many members of same botnet causing the similar behavior (e.g., sending illegitimate packets, starting to scan)
Connection rate [42,56,57,61,63,66]	These techniques are classified into two types: 1) connection success ratio and 2) connection rate, where connection rate refers to the number of connections instantiated within a certain window of time. See Table 4
SNORT and OpenFlow integrated [71,72]	These technique use combination of intrusion detection system (such as SNORT) and OpenFlow to detect attacks and reconfigure the network dynamically. An intrusion detection system monitor the traffic to identify malicious activities. OpenFlow switches are then dynamically reconfigured based on the detected attacks in real time

is more successful in getting the fine grained patterns that typical volume-based traffic analysis cannot capture [38–40]. Entropy can be computed on several features such as network flows, IP addresses, and number of packets. These techniques have the advantage of having a low calculation overhead [41]. Due to success of entropy-based algorithms in traditional network in detecting DDoS attacks, they have effectively been employed in SDN-based networks as well.

Mehdi et al. [42] uses maximum entropy estimation [36] to estimate the benign traffic distribution to detect network security problems in home and office networks using SDN. Traffic is divided into packet classes and maximum entropy estimation is then used to develop a baseline benign distribution for each class. Packet classes are based on protocols and destination port numbers. Experiments were conducted using OpenFlow switches and a NOX controller. However, the authors only used the low rate network traffic to do the experiments as they are focused on a home environment.

Giotis et al. [34] implemented a widely used entropy-based approach [43] to effectively detect DDoS, worm propagation, and portscan attacks. The flow-related traffic features used to detect anomolies are source and destination IP addresses and ports. Predefined thresholds on changes in the entropy values are used to identify the presence of anomolies. Moreover, the authors have used sflow [44] protocol that reduces communication between OpenFlow switches and the controller. However, flow rate sampling may affect the accuracy of anomaly detection schemes [41].

A distributed algorithm for entropy-based anomaly detection scheme was proposed by Wang et al. [41]. In that, DDoS flooding attack detection module runs in every OpenFlow edge switch thereby reducing the flow collection overload on the controller. Probability distribution of destination IP address at each switch is used to calculate the entropy. Once a DDoS flooding attack is detected, an alert information is sent to the controller.

While entropy-based techniques can be used to detect DDoS attacks, they suffer with a few limitations. Typically, entropy-based detection schemes detect unexpected changes in the time series of the entropy of certain traffic features. The probability distribution of a feature is represented by a single value when entropy is calculated. This is very effective for analysis; however, relevant information about the distribution of the analyzed feature is lost. This leads to masking of anomaly effects in some cases [45]. Similarly, the different distributions with the same amount of uncertainty cannot be distinguished by entropy values. Hence, the anomalies which do not disturb randomness remain undetected [46].

3.2 Machine Learning

Machine learning-based techniques are widely applied in tradational intrusion detection systems (IDS) [47,48]. Artificial neural networks [49], Bayesian networks [50], self-organizing map (SOM) [51], and fuzzy logic [52] principles and concepts are extensively used for detecting anomalies in IDS. They have been considerably successful and are

implemented widely both in wired and wireless networks [53]. Likewise, these techniques are effectively applied in SDN-based DDoS attack detection. In general, a machine learning-based technique differentiates the network flows based on certain features related to traffic characteristics and categorizes them as malicious and benign [54].

Self-organizing map, one of the most popular neural network models, is used by Braga et al. [55] for detection of DDoS attacks. This work presents SDN-centered DDoS attack detection based on six traffic flow features. These 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 features are collected by a flow collector module in a NOX based network and are passed to the classifier module for detection of illegitimate flows. Self-organizing map is used for flow analysis. This work shows successful DDoS attack detection with high rate of detection and very low rate of false alarms. Exploiting the capability of SDN for software-based traffic analysis, the method incurs low overhead when compared to traditional counterparts. Major reason for this low overhead is that traditional approaches collect and process every packet. However, in flow-based approach, samples are collected after every three seconds without hammering the rate of attack detection. Though, this work proposes an efficient DDoS attack detection method it does not discuss any mitigation strategy. Further, experiments are conducted on limited scale with small topology using single controller

Dotcenko et al. [56] proposed a security mechanism that uses the combination of rate limiting and TRW algorithms [57] with fuzzy logic inference. Mamdani algorithm was used to carry out fuzzy inference. This is considered as one of the most pragmatic approach in solving fuzzy modeling problems as it is based on fuzzy logic, and it avoids excessive computation. Results show that decision making based on fuzzy rules is better than using the security algorithms, separately. However, experiments were carried with low traffic and no mitigation strategy has been presented this work.

NICE [58] is an intrusion detection framework to prevent vulnerable virtual machines from being compromised in the cloud. NICE uses a novel attack graph approach for attack detection and prevention by correlating attack behavior. Attack graph in NICE is constructed based on multiple attributes including cloud system information, virtual network topology, configuration information, and vulnerability information. When vulnerabilities are discovered and attack is identified, several countermeasures including, traffic redirection, traffic isolation, port blocking, and network reconfiguration can be taken.

Dillon et al. [59] proposed anomaly detection algorithm that computes standard deviation of packet/byte rate collected for certain intervals. These flow statistics are collected from OpenFlow switch by the controller. Anomalies are spotted by comparing the expected deviation and the real deviation in the dataset. The authors have presented a three phase solution for DDoS attack detection and mitigation. In the first phase, anomaly in network flow is detected. This is then followed by identification of sources through packet analysis of samples in the second phase. The third phase drops all flows coming from malicious sources. The solution was deployed on a controller developed using Ryu SDN framework.

Machine learning-based techniques are preferred to detect suspicious activity based on the abnormal behavior of the network. However, the performance of these techniques is typically dependent upon the dataset that has been used for training.

3.3 Traffic Pattern Analysis

This technique works on the assumptions that the infected hosts exhibit similar behavioral patterns which are different from benign hosts. Typically, in case of botnet attacks, infected machines (known as bots) are usually controlled by single bot master. Similar traffic patterns are observed as a result of command that is sent to many members of same botnet causing the similar behavior (e.g., sending illegitimate packets, starting to scan) [60]. Foreseeing the SDN platform's success in providing security solutions, Shin et al. [61] presented a comprehensive framework—FRESCO, for development of OpenFlow-based security applications. Several applications are implemented to demonstrate the pragmatism of the FRESCO framework. For instance, BotMiner is an application which is built upon the FRESCO framework [62].The application assumes that hosts infected with the same botnet exhibit similar patterns at the network level, and these patterns are different from benign hosts. Hosts with similar packets per second and bytes per second are marked as potential threats. Jin and Wang [63] proposed a system to detect mobile malware by identifying suspicious network activities through real-time traffic analysis. One of the algorithm presented in this work detects the infected hosts by identifying aggregates of similar communications. Common destination, common connection time, and common platform features were used to extract aggregates of malware communications.

3.4 Connection Rate

Connection rate-based anomaly detection techniques can be further classified into two types. Connection success ratio and number of connections established as summarized in Table 4. Such that the former denotes ratio of successful connections over the overall connections, whereas the latter represents the

Table 4 DDoS attack detection using connection rate

Connection rate-based techniques	
Connection success ratio [42,57,61,63]	These techniques are based on the probability of a connection attempt being successful should be much higher for a benign host than a malicious host. Whenever the likelihood ratio for a host exceeds a certain threshold, it is declared as infected
Number of connections established [42,56,63,66]	These techniques are based on the notion that an infected machine attempts to connect to many different machines in a short period of time. On the other hand, an uninfected machine makes connections at a lower rate and is more likely to repeat connection attempts to recently accessed machines. It uses a threshold to limit the number of new connection attempts within a certain time period

number of connections instantiated within a certain window of time.

Connection success ratio technique takes into account that the probability of successful connection is much higher for a benign host as compared to malicious host [57]. For each host, the algorithm maintains a list of new connection requests, like TCP SYNs, which have not received a response, that is, SYN-ACK. If any of these connection times out or receives a TCP RST, the algorithms increases the likelihood ratio of the host being infected. The threshold random walk with credit-based rate limiting (TRW-CB) algorithm is typically used for detecting anomalies on the basis of connection success ratio. TRW-CB algorithm [57] has also been incorporated by SDN-based DDoS attack detection schemes [42,56,63]. Experiments have been conducted using various controllers which include using NOX controller [42], Beacon Controller [56], and Floodlight controller [63]. All results show that attack detection has been successful.

Likewise, connection rate-based technique assumes that number of connection attempts from a compromised machine to a server or other machines is much higher than benign hosts. In general, benign hosts make connections at a lower rate and are expected to repeat connection attempts to recently accessed machines. It utilizes a threshold to limit the number of new connection attempts within a certain time period [64,65]. These algorithms have also been implemented on many SDN controllers such as NOX [42], Beacon controller [56], Floodlight controller [63], and POX controller [66].

3.5 Integration of SNORT and OpenFlow

SNORT [67] has been a popular open-source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) since long. Recently, Open Information Security Foundation (OISF) has developed Suricata [68] as an alternative to SNORT. Suricata, also an open-source network security solution, has been termed as a next-generation NIDS [69]. It supports multi-threaded operations that makes it more efficient as compared to SNORT that typically is a single threaded system. Though Suricata appears to be gaining acceptance in traditional networking environments, we could not find any SDN-based solution that utilizes Suricata. However, the wide acceptance of SNORT in network community and increasing recognition of SDN have led to development of systems that implements the combination of OpenFlow and SNORT. Snortflow [43] is one such system that integrates both open flow and SNORT. The system enables cloud system to detect intrusions and deploy countermeasures by reconfiguring the cloud networking system on-the-fly. Similarly, NICE [58] presents a multi-phase distributed vulnerability detection, measurement, and countermeasure selection mechanism. It utilizes the capabilities of both SNORT and OpenFlow to detect an intrusion in cloud-based system. In particular, the solution is aimed at preventing virtual machines from being compromised in cloud-based system.

Once the DDoS attack is detected, we need mitigation strategies to ensure continuity of services. Effective DDoS attack protection solution is incomplete without the mitigation capabilities. Leveraging the capabilities of SDN various DDoS attack mitigation functionalities are incorporated in DDoS protection solutions as discussed in next section.

4 SDN-Based DDoS Attack Mitigation Techniques

SDN has been focused to improve agility and flexibility of a network. It empowers networks to respond quickly to changing network requirements via a centralized controller. The SDN controller provides a global view of the network. Further, the notion of the centralized controller, leads to consistent configuration throughout the network. Since, all network policies are defined at a centralized controller, it not only simplifies anomaly detection, but also facilitates in prompt invocation of mitigation mechanism [70]. For example, when a DDoS attack is detected, a threat mitigation application may effectively reprogram switches to block malicious traffic.

Many SDN-based DDoS defense mechanisms incorporates mitigation strategies. Dropping packets, blocking ports, and redirecting traffic are commonly used mitigation strate-

Table 5 Summary of prevalent DDoS attack mitigation techniques

Attack Mitigation techniques	
Drop packets [34,59,71]	The network traffic conforming to defined rules is transmitted and remaining is dropped
Block port [58,72]	The network traffic from attacking port is completely blocked
Redirection [58,66,72]	The legitimate traffic is redirected to new IP address
Control bandwidth [74]	The controller limits the flow transmission rate by allocating average bandwidth to each interface
Network reconfiguration and topology change [58]	The network controller changes the flow table on each switch to change the network topology
Deep Packet Inspection [58,72]	Deep packet inspection is a process that may completely examine both header and data part of packet. Deep packet inspection enables security functions and makes it possible to detect several types of attacks including buffer overflow attacks, denial-of-service attacks, and worms and virus attacks
MAC address change and/or IP address change [58,72]	When the attack is detected MAC address or IP address of the victim is changed. Legitimate traffic is routed to new address and malicious traffic it blocked
Quarantine or Traffic isolation [58,72]	This mitigation technique prevents the network resources from being overwhelmed by the volume-based attack by isolating the malicious traffic

gies that are extensively deployed in SDN [34,58,59,66,71,72]. Nevertheless, other mitigation strategies such as deep packet inspection, changing MAC and IP addresses, and isolating traffic are also implemented in SDN-based DDoS defense mechanisms [58,72]. Brief description of prevalent DDoS mitigation techniques is summarized in Table 5.

Dropping packets and blocking ports are simple and fast mitigation techniques that completely block the potential attack sources. These mitigation techniques may result in dropping the legitimate traffic. For instance, if a legitimate node or port is compromised, it is completely blocked resulting in dropping of any legitimate traffic it may perhaps generate. Similarly, in case of false attack detection, legitimate users are barred from using services. The traffic redirection technique forwards the traffic to new IP address after the attack is detected on an existing server. Initially, all connections to the existing server are torn down in order to prevent bots from directly accessing the redirected address. The redirected address either imposes high computation barrier for bots, such as using captcha [73], or imply deep packet inspection [58,72] to analyze and forward only legitimate traffic. Nonetheless, this technique increases the processing time of the network traffic along with the overall complexity. Attack mitigation by controlling bandwidth [74] at each router interface prevents user's starvation. Flows exceeding the assigned average bandwidth are penalized, such as experiencing delays. This mitigation strategy benefits low bandwidth flows, which cause low impact on the network and have short duration. However, legitimate flows may suffer if they do not adhere to network usage policies. Network topology change strategy [58] is very effective in SDN-based DDoS attack mitigation. This counter measure ensures that the path of an attack is disconnected immediately after an attack is detected. However, alternate paths can be utilized for legitimate users.

All the above-mentioned DDoS defense mechanisms are based on computing a threshold value that serves as a baseline of for attack detection. This threshold value remains constant and consistent for all the applications. That is, it cannot be customized with respect to different applications utilizing the network. However, in modern data centers, there are diverse applications with different requirements and expectations of tolerance of network interruption and service.

For instance, there are certain critical applications with high requirements of uptime and availability. For such applications, rapid detection of an attack is critical. In contrast, there are certain applications with comparatively moderate requirements of uptime. These applications can tolerate some delay in network availability, and they can also afford high false positive which may lead to service denial to legitimate users. Consequently, in case of a large-scale data center running diverse applications, a customizable solution is needed with varying degree of sensitivity to possible attacks [7,25].

We believe that the above-mentioned requirement is a highlighting factor for DDoS detection. While current SDN-based solutions provide improved flexibility over conventional DDoS attack detection solutions, they do not provide application-specific detection and mitigation.

In addition, scalability has now become an intrinsic requirement for DDoS attack prevention solutions. Continuous increase in massive DDoS attacks rationalize the need of scalability. In an SDN solution, scalability implies the ability of the network to handle large-scale traffic. For this purpose, a distributed controller architecture with capabilities of managing large amount of traffic is required. However, existing SDN solutions are focused only on a single controller, which

limits the capabilities of the network. A distributed controller framework also improves the availability of the controller in case of controller failure [75].

Motivated by the above-mentioned needs, we propose ProDefense framework that addresses the aforementioned requirements. Also, we present a case study of smart city data center to show the diversity in security requirements of various applications.

5 SDN-Based Proactive DDoS Defense Framework (ProDefense)

In this section, we present a customizable DDoS defense framework, named as ProDefense, which generates DDoS attack alerts considering the application's security requirement. ProDefense has been motivated by our analysis that different applications have different security requirements. Therefore, DDoS attack detection solution must include a customizable reaction mechanism for generating DDoS attack alerts to match the application-specific needs as discussed in previous section.

ProDefense leverages the programming and dynamic nature of SDN and implements an adaptive DDoS protection mechanism. Figure 2 depicts the architecture of the proposed framework. The distributed controller layer increases the reliability and scalability of the solution to live up to the needs of a huge data center.

We use smart city as a case study to highlight the versatility and key features of the ProDefense framework. In such a city, operations and services such as traffic control, power distribution, transport, and other utilities are monitored, operated, and controlled through ICT based infrastructure, smartly. A smart city system has many distinct characteristics. It is a highly complex system due to its enormous structure and heterogeneity. Moreover, smart city comprises of extremely critical control systems. These systems are so vital that their destruction or service disruption can affect safety, security and economy of the city. Consequently, smart city can become a prime target for terrorism and cyber attacks because of the critical nature of services that it provides. Adversaries can target smart city networks and application servers to paralyze the city almost instantly. Therefore, cyber security is one of the major challenges in a smart city system. Securing smart city data centers from cyber attacks and ensuring continuity of services is utmost desirable.

We argue that smart city network requires a highly customizable security solution to accommodate the need of diverse smart city applications. ProDefense framework is well suited for smart city network as it caters the need of application-specific security requirements. Numerous smart city applications can be developed to facilitate masses. These includes e-government, smart transport system, smart grid, smart healthcare, and smart rescue services. These applications can be divided into three categories, namely, catastrophic, critical, and moderate, with respect to their cyber security requirements as shown in Table 6.

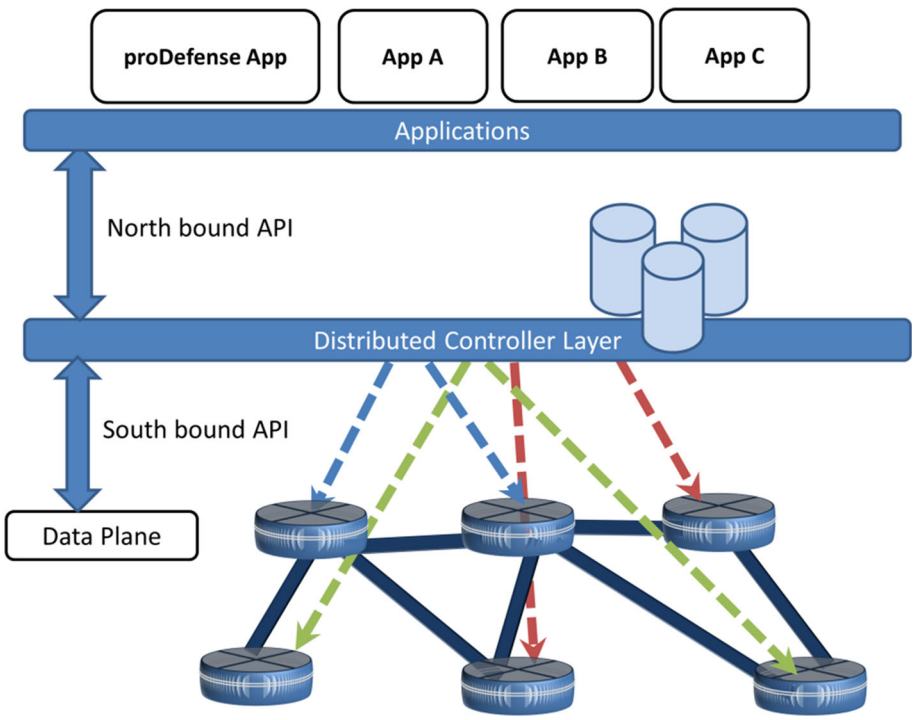

Fig. 2 Architectural design of ProDefense

Table 6 Categories of Smart city applications and related security aspects

Applications	Impact of DDoS attack	Application security requirements	Security solution requirement	ProDefense Filters
Smart Grid, Traffic Control System	Catastrophic	Such applications needs to trigger the mitigation system immediately. The security alert is generated foreseeing the malicious behavior before reaching the threshold. The security solution monitors the network traffic trends and predict the attack	**Extremely Agile** As the impact of attack is highly critical such applications requires very low false-negative rate of attack detection The security solution must take the proactive approach	HR filter
Healthcare, Location-based services	Critical	Such application trigger the mitigation system within a certain time period after the attack is detected	**Agile** As the impact of attack is critical such applications needs an adequate solution that neither reacts before time nor delays the alert	IR filter
Weather update, News, Parking	Moderate	Such applications delay the invoking of the mitigation system. Only when the attack is confirmed the mitigation system is triggered. These applications cannot afford to block legitimate traffic unnecessarily	**Moderately agile** As the continuity of services is more important such applications require very low false-positive rate of attack detection. The security solution takes the reactive approach	LR filter

The applications providing crucial services, such as Smart Grid, have stringent requirement for cyber security therefore need an attack detection mechanism having very low false-negative rate. Understandably, these services cannot afford a downtime therefore an extremely agile security solution is needed that takes proactive approach. The detection and mitigation solutions must go an extra mile to fulfill the requirements of applications in catastrophic category. In contrast, applications providing information services to citizens such as weather, news, and sports have less stringent cyber security requirements. These applications needs a solution with very low false-positive rate hence a moderately agile security solution seems appropriate in this case as shown in Table 6.

Motivated by the use of exponentially weighted moving average (EWMA) filters [67,68], ProDefense utilizes such filters to customize the attack detection. Equation (1) shows the modified form of equation of EWMA filter used in ProDefense.

$$\mathrm{PT}_t = \alpha \mathrm{PT}_{t-1} + (1-\alpha)\,\mathrm{CT}_t + c \tag{1}$$

where: $\mathrm{PT_t}$ = predicted traffic, $\mathrm{CT_t}$ = current traffic, α = gain, c constant dependent on traffic characteristics.

Typically, these filters are either able to react quickly in case of attack or respond slowly, masking the transients depending upon the value of α known as the gain. Lower values for α will generate DDoS alert almost instantly. We present three different variants of these filters. These are highly reactive (HR) filter, intermediate reactive (IR) filter, and least reactive (LR) filter for the applications in catastrophic, critical, and marginal categories respectively.

HR filter takes the value of $\alpha = 0.1$ in Eq. (1) which causes the effect of current traffic rate to be dominant. That is, HR filter immediately generates security alerts. This filter can generate more false positive as little spikes in network traffic are also taken as threats. This type of filter provides early warning mechanism which is suitable for highly critical networks such as smart grid.

An IR filter takes the value of $\alpha = 0.5$ in Eq. (1). This makes it to takes into account both the current rate of traffic and the previous rates. For most networks, this filter will be appropriate as it equally takes into account current and previous traffic rates

The LR filter is the most stable filter that reacts much slowly to attacks as it takes the value of $\alpha = 0.9$ in Eq. (1). This filter is more useful where cost of false positive is higher. This filter is a deliberate attempt to lower down the false-positive rate. In case of smart city application this filter is suitable for applications that require less agility.

The design of our SDN-oriented ProDefense framework is modular. It has three major components, namely traffic flow collector, policy engine, attack detector, and mitigation engine as depicted in Fig. 3. The mitigation engine, adaptive filters, and threshold detector functions are decoupled and can be adapted to future security requirements. Traffic flow collector collects the flow entries from OpenFlow switches. The policy engine is used to define attack detection policy and mitigation policy. For attack detection, policy engine is

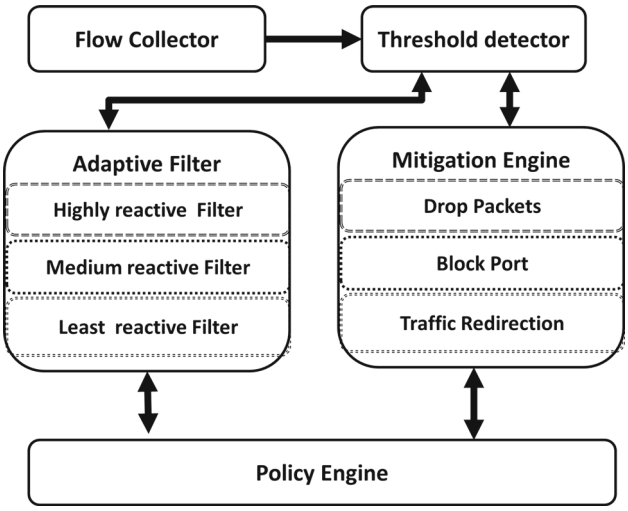

Fig. 3 ProDefense framework

used to configure the type of filter to be used by the network for detecting attacks. Three types of filter can be configured, that is HR filter, IR filter, and LR filter. For attack mitigation, policy engine defines the defense strategy. ProDefense supports many types of defense strategies including dropping packets, blocking ports, and traffic redirection. The attack detector module takes the input from Traffic Flow collector and generates security alerts with respect to the policy defined in Policy Engine. These security alerts trigger the mitigation module for taking appropriate action.

The ProDefense framework has been designed to meet the requirements of an effective DDoS defense mechanism (Sect. 2.1). The framework is expected to immediately respond to attacks and ensure continuity of operations through integrated detection and mitigation mechanisms. This is specifically useful in the context of smart city, where applications requirements may vary. In addition, the framework allows customization of policies through filters to fulfill the need of various applications. The key features of ProDefense can be summarized below:

(a) ProDefense can filter attacks traffic by sending suspicious traffic to controller for inspection and allowing the legitimate flows to pass through the network without disruption.
(b) It is capable to monitor all traffic flows. This would allow the framework to detect anomalous behavior of both internal and external nodes.
(c) ProDefense utilizes distributed controllers. This would allow the framework to meet scalability, performance, and reliability requirements.
(d) ProDefense does not require any additional hardware and can be easily deployed as an application running on top of existing SDN controller.
(e) It is a lightweight application which does not require high computation.
(f) The adaptive filter module employed by ProDefense helps in rapid detection of attacks. Further, it allows meeting application-specific requirements.
(g) The ProDefense framework supports various mitigation strategies such as blocking port, diverting flows, and controlling bandwidth. Using various configuration of filters, ProDefense can be customized to achieve essential accuracy of attack detection.

The ProDefense framework is a novel approach toward securing smart city data centers against DDoS attacks. Application-specific customization serves works as an early warning system to protect critical city services and infrastructure from cyber criminals. Incorporation of mitigation module ensures continuity of city services to maximum extent.

6 Open Research Challenges, Future Research Directions, and Recommendations

SDN promises programmability and centralized control that enables an adaptive and efficient network. SDN has been prominently termed as a technology behind future networks. However, there are several challenges which are needed to be catered. We highlight some of these crucial challenges below where SDN community needs to exploit its research potential.

6.1 Adoption of SDN

Adoption of SDN is one of the foremost challenge that need to be taken into account. A major requirement is the deployment of SDN-enabled hardware, which adheres additional cost. Further, administration of SDN may require enhanced training to design, maintain, and operate networks. Debugging and maintaining SDN-enabled networks may also be non-trivial due to lack of experts [20]. These problems have been addressed using hybrid SDN models [79] that help in partial transition to SDN. These models integrate SDN with traditional network. However, transition from traditional networks to SDNs remain an open research area that need more efforts from the research community.

6.2 Availability of Production-Level SDN Controllers

Availability and reliability of a production-level SDN controller is another major challenge [79]. Despite rapid research growth and technological advancements to build effective SDN controllers, availability of SDN controllers with proven

technical support remains a barrier in deployment of SDN on a large scale. The community needs SDN controllers that can match the performance of their traditional counterparts in terms of maintenance, security, availability and reliability. Major efforts are required in the form of open-source software as well as commercial license products. Development of distributed SDN controllers meeting the requirements of scalability, availability, and reliability of large-scale data centers is also an issue, which require immediate attention from the research community. Further, the distributed SDN controller architecture—where number of SDN controllers are created to enable automatic failover and load balancing, requires consistent updates throughout the network. The issues related to synchronization of these controllers while maintaining the low overhead is a matter of open debate.

6.3 Security of SDN Controller

Inherently, SDN has a centralized control plane, which empowers it for improved control and management. However, this central management feature can also become a security threat for SDN such that the SDN controller is also a vulnerable target for attackers. This is specifically true in the absence of a robust and secure controller platform [17] In this context, conducive efforts are needed in exploring innovative methods for ensuring security of an SDN controller. Similarly, communication channels between switches and controller must also be secured. Any attack on the communication link between the controller and the switch can cause a considerable damage to the network [17]. Development and implementation of effective security specification for the controller–switch interface is also the subject of concern for the SDN community.

6.4 Real-Time Network Traffic Monitoring Overhead

Monitoring of network traffic is vital for network management. It also provides basis of network security solutions. Specifically, DDoS attack-based security solutions requires real-time monitoring of network traffic to generate security alerts leading to timely mitigation of attacks. Existing monitoring solutions such as *sflow* [34] imposes considerable overhead that reduces efficiency of security applications in large-scale networks. Though frequent collection of statistics from network devices to controller increases the overhead it results in high accuracy. Trade-off between reducing this overhead, while maintaining high accuracy, is an important direction.

6.5 Overwhelming Controller with Network Traffic

SDN controllers are also susceptible to malicious attacks. SDN controllers can be overwhelmed merely by sending unknown packets to OpenFlow switches [80,81]. Within SDN, whenever a packet reaches an OpenFlow switch, it is forwarded according to the rule specified in the flow table of the switch. In case no matching flow entry is found, the packet is forwarded to the controller for further action. The controller then decides how to handle the packet and installs a new rule in the OpenFlow switch, if required. With large volume of such traffic coming in the SDN controllers direction can not only bring down the controller, but also absorb high bandwidth of the communication channel between switches and controller. SDN controller can become a performance bottle neck and may not work efficiently. Improving the efficiency and performance of SDN controllers so that they can scale up with increasing number of OpenFlow switches is a complex task that needs to be addressed effectively.

6.6 Identifying Malicious Traffic

Most of the techniques used to detect DDoS attacks are based on some threshold value. The alerts are generated when network traffic crosses the maximum allowed threshold limit. Therefore, if illegitimate traffic remains below the threshold, it remains undetected. Sophisticated attacks that resembles the traffic patterns of legitimate users are also difficult to detect. These include flash crowd-based attacks [82] that can mimic legitimate behavior in order to bypass the defense mechanism easily. Typically, threshold-based mechanisms activates mitigation mechanisms when network traffic exceeds a specific value. This in turn causes the degradation or blocking of services to legitimate users. Hence, more reliable and effective mechanisms are required to differentiate between legitimate and illegitimate traffic so that malicious only users can be identified and blocked.

6.7 Vulnerabilities Due by SDN Applications

SDN separates control plane from data plane which facilitates high level network abstraction and programmability. This key feature opens the SDN network to applications that can be used to implement innovative network functions. This can lead to installation of applications with conflicting rules [83]. Also malicious applications can be installed that can bring down the controller. Various mechanisms that can effectively inspect SDN applications before executing them are needed. Likewise, securing controller from vulnerabilities caused by application is an area of further study

The above-mentioned analysis provides a holistic view about SDN and associated challenges in the adaptation. Understandably, professionals and practitioners are required to meet the growing demand and challenges of evolving SDN domain. It also highlights a few significant research directions. Most importantly, the research community should play an active role in ensuring the security of SDN platform.

These include security of the controller and associated links.. Thorough efforts are also required to incorporate and adopt SDN-based security solutions. In this context, significance of cost-effectiveness is also important in global adaptation of SDN. The developer community should play a leading role to meet these expectations. The role of computer scientists is also important in determining balance between a highly reactive threshold-based detection scheme vs a conservative framework which may allow sudden increase in network traffic.

7 Concluding Remarks

The increased reliance on cyber physical systems and advancements in networking and cloud technologies have highlighted the need for protection of network and computing infrastructure against DoS attacks. However detection and mitigation of DDoS attacks has remained an unaccomplished task. Realizing this open-ended problem, we have made several noticeable contributions.

In this paper, we have discussed major requirements and challenges in meeting this uphill task of prevention against DDoS attacks. An effective DDoS attack detection system requires the ability to rapidly respond to increasing malware traffic. Furthermore, customizability and ease of management in application of rules for detection and prevention against DDoS attacks is also required. Considering the emerging potential of SDN in meeting network-wide requirements of flexibility, management, and adaptability, we assessed it's capabilities against mitigation and prevention of DDoS attacks. We classified existing SDN-based solutions according to various detection and mitigation techniques and highlighted pros and cons against each category. Our analysis revealed that existing SDN-based solutions for DDoS detection and mitigation failed to meet application-specific requirements for DDoS attack detection. In that, traffic threshold for DDoS attack detection can vary for each application; however, existing solutions do not implement any mechanism to cater this important need and implement same level of threshold for each application. Further, existing solutions utilizing SDN for DDoS attack detection and mitigation incorporates a single controller. This not only creates a bottleneck for the network traffic but also leads to single point of failure for the SDN network. Although, SDN can incorporate distributed platform but existing solutions have not realized it to its full potential.

We have addressed the two problems by proposing an efficient system for DDoS attack detection and mitigation. Our proposed framework, ProDefense, incorporates application-specific criteria for network traffic threshold. This permits implementation of customizable criteria for detection of DDoS attacks. In addition, ProDefense utilizes distributed controller platform which allows load balancing and reduces possibilities for controller failure.

ProDefense is a major step in resolving the above-mentioned issues. We anticipate that ProDefense can potentially be utilized in a wide range of systems including cyber physical systems, smart grid, and e-governance. All such applications exhibit varying degree of tolerance for network attacks. ProDefense would require to implement consistency among all the controllers in order to ensure rapid incorporation of network rules. ProDefense framework can be implemented in any programming or scripting language that is compatible with the SDN controller being used. However, we are implementing the key modules of ProDefense such as threshold detector and adaptive filter module using node.js. We selected node.js because it is an asynchronous IO model which supports parallelism, allowing the SDN application to interact with distributed controllers and monitoring systems without blocking [76]. This will result in a fast and consistent response time. For implementation of flow collector module, we are using sflow [44] to collect and monitor network traffic statistics. Using sflow provides a number of advantages. It is a lightweight sampling technology with no performance overhead in switches and routers. Moreover, the sFlow architecture not only scales better but it is also supported by most switch vendors [77]. As a future step of our research, we are planning to perform comprehensive experiments using ProDefense to study the effectiveness of the framework in protecting Smart city applications. Lastly, SDN is not a silver bullet solution to all network security problems. In this paper, we also highlighted open research issues, challenges, and recommendations related to SDN-based DDoS attack detection and mitigation that require further research [78].

References

1. Geng, X.J.; Whinston, A.B.: Defeating distributed denial of service attacks. IT Prof. **2**(4), 36–42 (2000)
2. Ottis, R.: Analysis of the 2007 cyber attacks against Estonia from the information warfare perspective. In: Proceedings of the 7th European Conference on Information Warfare, p. 163 (2008)
3. Bangladesh Bank heist. (2016). https://en.wikipedia.org/wiki/2016_Bangladesh_Bank_heist
4. European renewable power grid rocked by cyber-attack. EurActiv (2012). https://www.euractiv.com/section/energy/news/european-renewable-power-grid-rocked-by-cyber-attack/
5. Musil, S.: Record-breaking DDoS attack in Europe hits 400 Gbps. CNET (2014). http://www.cnet.com/news/record-breaking-ddos-attack-in-europe-hits-400gbps/
6. Paroutis, S.; Bennett, M.; Heracleous, L.: A strategic view on smart city technology: the case of IBM Smarter Cities during a recession. Technol. Forecast. Soc. Chang. **89**, 262–272 (2014)
7. Bawany, N.Z.; Shamsi, J.A.: Smart city architecture: Vision and challenges. Int. J. Adv. Comput. Sci. Appl. **6**(11) (2015)
8. Yadav, V.K.; Trivedi, M.C.; Mehtre, B.M.: DDA: an approach to handle DDoS (Ping flood) attack. Adv. Intell. Syst. Comput. **408**, 11–23 (2016)

9. Saied, A.; Overill, R.E.; Radzik, T.: Detection of known and unknown DDoS attacks using artificial neural networks. Commun. Comput. Inf. Sci. **172**, 385–393 (2016)
10. Hoque, N.; Bhattacharyya, D.; Kalita, J.: Botnet in DDoS attacks: trends and challenges. IEEE Commun. Surv. Tutor. **99**, 1–1 (2015)
11. Arbor Networks Inc. http://www.arbornetworks.com
12. Arbor networks detects largest ever DDoS attack in Q1 2015 DDoS report. In: Arbor Networks (2015). http://www.arbornetworks.com/arbor-networks-detects-largest-ever-ddos-attack-in-q1-2015-ddos-report
13. Jain, S.; et al.: B4: experience with a globally-deployed software defined WA. ACM SIGCOMM Comput. Commun. Rev. **43**(4), 3–14 (2013)
14. Technol, I.: Secure and Dependable SDNs, Feb 2016 (2015)
15. Shalimov, A.; Zuikov, D.; Zimarina, D.; Pashkov, V.; Smeliansky, R.: Advanced study of SDN/openflow controllers. In: Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia on - CEE-SECR '13 Oct pp. 1–6 (2013)
16. Schehlmann, L.; Abt, S.; Baier, H.: Blessing or curse? Revisiting security aspects of software-defined networking. In: Proceedings of the 10th International Conference on Network and Service Management, CNSM 2014, no. 1, pp. 382–387 (2015)
17. Kreutz, D.; Ramos, F.M.V.; Verissimo, P.: Towards secure and dependable software-defined networks. In: Proceedings of the second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking—HotSDN '13, p. 55 (2013)
18. Wang, B.; Zheng, Y.; Lou, W.; Hou, Y.T.: DDoS attack protection in the era of cloud computing and software-defined networking. In: 2014 IEEE 22nd International Conference on Network Protocols, pp. 624–629 (2014)
19. Thapngam, T.; Yu, S.; Zhou, W.; Beliakov, G.: Discriminating DDoS attack traffic from flash crowd through packet arrival patterns. In: 2011 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2011, pp. 952–957 (2011)
20. Xia, W.; Wen, Y.; Member, S.; Heng Foh, C.; Niyato, D.; Xie, H.: A survey on software-defined networking. IEEE Commun. Surv. Tutor. 17(1), 27–51 (2015)
21. Liao, Q.; Li, H.; Kang, S.; Liu, C.: Application layer DDoS attack detection using cluster with label based on sparse vector decomposition and rhythm matching. Secur. Commun. Netw. **8**(17), 3111–3120 (2015)
22. Stewart, J.M.: Network Security, Firewalls and VPNs. Jones & Bartlett Publishers (2013)
23. DDoS: website-crippling cyber-attacks to rise in 2016. BBC News. http://www.bbc.com/news/technology-35376327
24. Q1 2016 Global DDoS Threat Landscape Report. Incapsula. https://www.incapsula.com/blog/q1-2016-global-ddos-threat-landscape-report.html
25. Bawany, N.Z.; Shamsi, J.A.: Application layer DDoS attack defense framework for smart city using SDN. In: Computer Science, Computer Engineering, and Social Media (CSCESM) (2016)
26. Kreutz, D.; Ramos, F.M.V.; Verissimo, P.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015)
27. Khondoker, R.; Zaalouk, A.; Marx, R.; Bayarou, K.: Feature-based comparison and selection of Software Defined Networking (SDN) controllers. In: 2014 World Congress on Computer Applications and Information Systems (WCCAIS), pp. 1–7. IEEE (2014)
28. Berde, P.; Gerola, M.; Hart, J.; Higuchi, Y.; Kobayashi, M.; Koide, T.; Lantz, B.; Snow, W.; Parulkar, G.; O'Connor, B.; Radoslavov, P.: ONOS. In: Proceedings of the third workshop on Hot topics in software defined networking—HotSDN '14, pp. 1–6 (2014)
29. Linux Foundation. http://www.opendaylight.org
30. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)
31. Coughlin, M.: A survey of SDN security research. In: Future Networks and Services (SDN4FNS), IEEE (2013)
32. Kim, J.; Firoozjaei, M.D.; Jeong, J.P.; Kim, H.; Park, J.-S.: SDN-based security services using interface to network security functions. In: 2015 International Conference on Information and Communication Technology Convergence (ICTC), pp. 526–529. IEEE (2015)
33. Yan, Q.; Yu, F.R.: Distributed denial of service attacks in software-defined networking with cloud computing. IEEE Commun. Mag. **53**(4), 52–59 (2015)
34. Giotis, K.; Argyropoulos, C.; Androulidakis, G.; Kalogeras, D.; Maglaris, V.: Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. Comput. Netw. **62**, 122–136 (2014)
35. Lee, W.; Xiang, D.: Information-theoretic measures for anomaly detection. In: Proceedings of the 2001 IEEE Symposium on Security and Privacy, S&P 2001, pp. 130–143. IEEE (2001)
36. Gu, Y.; McCallum, A.; Towsley, D.: Detecting anomalies in network traffic using maximum entropy estimation. In: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement, p. 32. USENIX Association (2005)
37. Bereziński, P.; Szpyrka, M.; Jasiul, B.; Mazur, M.: Network anomaly detection using parameterized entropy. In: Computer Information Systems and Industrial Management. Springer, Berlin (2014)
38. Nychis, G.; Sekar, V.; Andersen, D.G.; Kim, H.; Zhang, H.: An empirical evaluation of entropy-based traffic anomaly detection. In: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement Conference—IMC '08, p. 151 (2008)
39. Brauckhoff, D.; Tellenbach, B.; Wagner, A.; May, M.; Lakhina, A.: Impact of packet sampling on anomaly detection metrics. In: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, pp. 159–164 (2006)
40. Androulidakis, G.; Chatzigiannakis, V.; Papavassiliou, S.: Network anomaly detection and classification via opportunistic sampling. IEEE Netw. **23**(1), 6–12 (2009)
41. Wang, R.; Jia, Z.; Ju, L.: An entropy-based distributed DDoS detection mechanism in software-defined networking. In: 2015 IEEE Trustcom/BigDataSE/ISPA, pp. 310–317 (2015)
42. Mehdi, S.,A.,S.; Khalid, J.; Khayam, S.,A.,S.: Revisiting traffic anomaly detection using software defined networking. In: Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection, pp. 161–180 (2011)
43. Lakhina, A.; Crovella, M.; Diot, C.: Mining anomalies using traffic feature distributions. ACM SIGCOMM Comput. Commun. Rev. **35**(4), 217 (2005)
44. sflow. http://www.sflow.com
45. Fiadino, P.; Alconzo, A.,D.; Schiavone, M.; Casas, P.: Challenging entropy-based anomaly detection and diagnosis in cellular networks. In: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (2015)
46. Javed, M.; Ashfaq, A.B.; Shafiq, M.Z.; Khayam, S.A.: On the Inefficient Use of Entropy for Anomaly Detection. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 5758 LNCS, no. c, pp. 369–370 (2009)
47. Tsai, C.F.; Hsu, Y.F.; Lin, C.Y.; Lin, W.Y.: Intrusion detection by machine learning: a review. Expert Syst. Appl. **36**(10), 11994–12000 (2009)
48. Sommer, R.; Paxson, V.: Outside the closed world: on using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy, pp. 305–316 (2010)

49. Mukkamala, S.; Janoski, G.; Sung, A.: Intrusion detection using neural networks and support vector machines. In: Proceeding of the 2002 International Joint Conference on Neural Networks, vols. 1–3, pp. 1702–1707 (2002)
50. Kruegel, C.; Mutz, D.; Robertson, W.; Valeur, F.: Bayesian event classification for intrusion detection. In: Proceedings—Annual Computer Security Applications Conference, ACSAC, pp. 14–23 (2003)
51. Kayacik, H.G.; Zincir-Heywood, A.N.; Heywood, M.I.: A hierarchical SOM-based intrusion detection system. Eng. Appl. Artif. Intell. **20**(4), 439–451 (2007)
52. Mabu, S.; Chen, C.; Lu, N.; Shimada, K.; Hirasawa, K.: An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. IEEE Trans. Syst. Man Cybern C Appl. Rev. **41**(1), 130–139 (2011)
53. Abduvaliyev, A.; Pathan, A.-S.K.; Zhou, J.; Roman, R.; Wong, W.-C.: On the vital areas of intrusion detection systems in wireless sensor networks. IEEE Commun. Surv. Tutor. **15**(3), 1223–1237 (2013)
54. Xu, Y.; Liu, Y.: DDoS attack detection under SDN context. In: IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, pp.1–9. IEEE (2016)
55. Braga, R.; Mota, E.; Passito, A.: Lightweight DDoS flooding attack detection using NOX/OpenFlow. In: LCN '10 Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks, pp. 408–415. IEEE, Washington (2010)
56. Dotcenko, S.; Vladyko, A.; Letenko, I.: A fuzzy logic-based information security management for software-defined networks. In: 16th International Conference on Advanced Communication Technology (ICACT), pp. 167–171. IEEE (2014)
57. Schechter, S.E.; Jung, J.; Berger, A.W.: Fast detection of scanning worm infections. In: International Workshop on Recent Advances in Intrusion Detection. Springer, Berlin, Heidelberg (2004)
58. Chung, C.-J.; Khatkar, P.; Xing, T.; Lee, J.; Huang, D.: NICE: Network intrusion detection and countermeasure. IEEE Trans. Dependable Secure Comput. **10**(4), 198–211 (2013)
59. Dillon, C.; Berkelaar, M.: OpenFlow (D) DoS Mitigation. Technical Report (Feb 2014). http://www.delaat.net/rp/2013-2014/p42/report.pdf (2014)
60. Yen, T.-F.; Reiter, M.K.: Traffic aggregation for malware detection. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 207–227. Springer, Berlin, Heidelberg (2008)
61. Shin, S.; Porras, P.; Yegneswaran, V.; Fong, M.; Gu, G.; Tyson, M.; Texas, A.; Station, C.; Park, M.: Fresco: modular composable security services for software-defined networks. In: Network and Distributed System Security Symposium, pp. 1–16. (2013)
62. Gu, G.; Perdisci, R.; Zhang, J.; Lee, W.: BotMiner: clustering analysis of network traffic for protocol- and structure-independent Botnet detection. In: USENIX Security Symposium, vol. 5, no. 2, pp. 139–154 (2008)
63. Jin, R.; Wang, B.: Malware detection for mobile devices using software-defined networking. In: GREE '13 Proceedings of the 2013 Second GENI Research and Educational Experiment Workshop, pp. 81–88. IEEE, Washington (2013)
64. Twycross, J.; Williamson, M.M.: Implementing and testing a virus throttle. In: Proceedings of the 11th USENIX Security Symposium, pp. 285–294 (2003)
65. Williamson, M.M.: Throttling viruses: restricting propagation to defeat malicious mobile code. In: Proceedings—18th Annual Computer Security Applications Conference, pp. 61–68. IEEE (2002)
66. Lim, S.; Ha, J.; Kim, H.; Kim, Y.; Yang, S.: A SDN-oriented DDoS blocking scheme for botnet-based attacks. In: Sixth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 63–68. IEEE (2014)
67. Roesch, M.: Snort: lightweight intrusion detection for networks. In: LISA '99: 13th Systems Administration Conference, pp. 229–238 (1999)
68. White, J.S.; Fitzsimmons, T.; Matthews, J.N.: Quantitative analysis of intrusion detection systems: Snort and Suricata. Proc. SPIE **8757**, 875704 (2013)
69. Albin, E.; Rowe, N.C.: A realistic experimental comparison of the Suricata and Snort intrusion-detection systems. In: 2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 122–127. IEEE (2012)
70. Ali, S.T.; Sivaraman, V.; Radford, A.; Jha, S.: A survey of securing networks using software defined networking. IEEE Trans. Reliab. **64**(3), 1086–1097 (2015)
71. Chin, T.; Mountrouidou, X.; Li, X.; Xiong, K.: Selective packet inspection to detect DoS flooding using software defined networking (SDN). In: 2015 IEEE 35th International Conference on distributed Computing Systems Workshops (ICDCSW), pp. 95–99. IEEE (2015)
72. Xing, T.; Huang, D.; Xu, L.; Chung, C.J.; Khatkar, P.: SnortFlow: a OpenFlow-based intrusion prevention system in cloud environment. In: Proceedings—2013 2nd GENI Research and Educational Experiment Workshop, GREE 2013, pp. 89–92 (2013)
73. Von Ahn, L.; Blum, M.; Hopper, N.J.; Langford, J.: CAPTCHA: using hard AI problems for security. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 294–311. Springer, Berlin, Heidelberg (2003)
74. Piedrahita, A.F.M.; Rueda, S.; Mattos, D.M.F.; Duarte, O.C.M.B.: FlowFence: a denial of service defense system for software defined networking. In: 2015 Global Information Infrastructure and Networking Symposium (GIIS), Guadalajara, pp. 1–6. (2015)
75. Suciu, G.; Vulpe, A.; Halunga, S.; Fratu, O.; Todoran, G.; Suciu, V.: Smart cities built on resilient cloud computing and secure internet of things. In: 2013 19th International Conference on Control Systems and Computer Science (CSCS), pp. 513–518. IEEE (2013)
76. Afaq, M.; Rehman, S.; Song, W.-C.: Large flows detection, marking, and mitigation based on sFlow standard in SDN. J. Korea Multimedia Soc. **18**(2), 189–198 (2015)
77. Sqalli, M.H.; Al-Haidari, F.; Salah, K.: Edos-shield-a two-steps mitigation technique against edos attacks in cloud computing. In: 2011 Fourth IEEE International Conference on Utility and Cloud Computing (UCC), pp. 49–56. IEEE (2011)
78. Salman, O.; Elhajj, I.H.; Kayssi, A.; Chehab, A.: SDN controllers: a comparative study. In: Proceedings of the 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1–6. IEEE (2016)
79. Vissicchio, S.; Vanbever, L.; Bonaventure, O.: Opportunities and research challenges of hybrid software defined networks. ACM SIGCOMM Comput. Commun. Rev. **44**(2), 70–75 (2014)
80. Akyildiz, I.F.; Lee, A.; Wang, P.; Luo, M.; Chou, W.: A roadmap for traffic engineering in software defined networks. Comput. Netw. **71**, 1–30 (2014)
81. Akyildiz, I.F.; Lee, A.; Wang, P.; Luo, M.; Chou, W.: Research challenges for traffic engineering in software defined networks. IEEE Netw. **30**(3), 52–58 (2016)
82. Yu, S.; Zhou, W.; Jia, W.; Guo, S.; Xiang, Y.; Tang, F.: Discriminating DDoS attacks from flash crowds using flow correlation coefficient. IEEE Trans. Parallel Distrib. Syst. **23**(6), 1073–1080 (2012)
83. Lee, S.; Yoon, C.; Shin, S.: The smaller, the Shrewder: a simple malicious application can kill an entire SDN environment. In: Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, pp. 23–28. ACM (2016)

84. CCTV-based botnet used for DDoS attacks. https://www.ddosattacks.net/a-massive-botnet-of-cctv-cameras-involved-in-ferocious-ddos-attacks Accessed 04 July 2016
85. Sucuri, Inc. Delaware Corporation. https://sucuri.net
86. DDoS Attack on Bank of Greece Website https://www.hackread.com/anonymous-ddos-attack-bank-greece-website-down. Accessed 04 July 2016
87. HSBC Internet Banking Services Down After DDoS Attack. http://www.telegraph.co.uk/finance/newsbysector/banksandfinance/12129411/HSBC-online-banking-service-crashes-again.html. Accessed 04 July 2016
88. HSBC Bank. www.hsbc.co.uk. Accessed 02 July 2016
89. Irish Government Websites temporarily offline due to DDoS-attack. http://www.bbc.com/news/world-europe-35379817. Accessed 04 July 2016
90. Laskar, S.; Mishra, D.: Qualified vector match and merge algorithm (QVMMA) for DDoS prevention and mitigation. Procedia Comput. Sci. **79**, 41–52 (2016)
91. Web Attack Knocks BBC Websites Offline. http://www.bbc.com/news/technology-35204915. Accessed 04 July 2016
92. Thai Government Websites hit by denial-of-service attack. http://www.bbc.com/news/world-asia-34409343. Accessed 04 July 2016
93. Hack attack leaves 1,400 airline passengers grounded. http://www.cnbc.com/2015/06/22/hack-attack-leaves-1400-passengers-of-polish-airline-lot-grounded.html. Accessed 04 July 2016
94. Hacker group 'Anonymous' claims credit for federal cyber attacks. http://ottawacitizen.com/news/politics/federal-computer-servers-cyber-attacked-clement. Accessed 04 Jul 2016
95. Musil, S.: Record-breaking DDoS attack in Europe hits 400 Gbps. CNET http://www.cnet.com/news/record-breaking-ddos-attack-in-europe-hits-400gbps/
96. Cloudflare, Cloudflare Organization. https://www.cloudflare.com. Accessed 08 Aug 2015
97. Wong, F.; Tan, C.X.: A survey of trends in massive DDoS attacks and cloud-based mitigations. Int. J. Netw. Secur. Appl. (IJNSA) 6(3), 57–71 (2014).
98. Zargar, S.T.; Joshi, J.; Tipper, D.; Member, S.: A survey of defense mechanisms against distributed denial of service (DDoS). IEEE Commun. Surv. Tutor. **15**(4), 2046–2069 (2013)
99. Kaufman, C.; Perlman, R.; Sommerfeld, B.: DoS protection for UDP-based protocols. In: Proceedings of the 10th ACM Conference on Computer and communication security—CCS '03 p. 2, (2003)
100. Peng, T.; Leckie, C.; Ramamohanarao, K.: Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Comput. Surv. **39**(1), 3-es (2007)
101. Czyz, J.; Kallitsis, M.; Papadopoulos, C.; Bailey, M.: Taming the 800 Pound Gorilla: the rise and decline of NTP DDoS attacks. In: IMC, pp. 435–448 (2014)