Ad Hoc & Sensor Wireless Networks, Vol. 33, pp. 275–299
Reprints available directly from the publisher
Photocopying permitted by license only

Detection of DDoS Attacks Against Wireless SDN Controllers Based on the Fuzzy Synthetic Evaluation Decision-making Model

QIAO YAN[1], QINGXIANG GONG[1] AND FANG-AN DENG[2]

[1]*College of Computer Science and Software Engineering*
Shenzhen University Shenzhen, Guangdong, China
E-mail: yanq@szu.edu.cn; gongqingxiang@email.szu.edu.cn
[2]*School of Mathematics and Computer Science*
Shanxi Sci-Tech University, Hanzhong, Shanxi, China
E-mail: FanganDeng@snut.edu.cn

Received: March 1, 2016. Accepted: September 12, 2016.

Software Defined Networking (SDN) is a new network architecture that separates the control plane and the data plane and provides logically central control over the whole network. Because SDN controller combines the upper application layer and the underlying infrastructure layer, it may face the problem of single-point failure. If it is made unreachable by a Distributed Denial of Service (DDoS) attacks, the whole network may not work normally. Especially for wireless SDN controllers, due to the secure channel for the control protocol in communication between wireless SDN controller and wireless SDN devices is exposed in the attacker's field of vision, the attack range of DDoS attackers will be expanded. To mitigate this threat, this paper introduces a solution based on fuzzy synthetic evaluation decision-making model that is effective and lightweight in terms of the resources that it uses. Importantly, it takes many factors that can be used to detect DDoS attacks into consideration and makes a comprehensive judgment according to multi-factors. To test the solution, the paper also proposes three kinds of DDoS attacks specialized for SDN network and presents two kinds of DDoS attacks inherited from traditional network. Every attack has been tested with the detection method. Finally, we also make a comparable experiment to show its advantage to other DDoS detection algorithm based on single factor. The results show its efficiency in detecting most of the DDoS attacks.

Keywords: SDN, openflow, DDoS attacks, fuzzy synthetic evaluation decision-making model, entropy

1 INTRODUCTION

Software Define Networking (SDN) is a new network architecture designed to separate the control plane and the data plane, providing logically central control over the network, which is quite different from the distributed traditional networks [1]. In traditional network, once the network is configured with predefined policies, it is difficult to reconfigure the network. Furthermore, configuring the network manually is cumbersome and error-prone, and can not fully utilize the capability of physical network infrastructure. SDN is widely used making these problems solved easily and greatly improve the utilization of network equipment [1]. The most representative use case of SDN is google's B4 data center network which takes google three years to accomplish B4 production deployment [2]. B4's centralized traffic engineering service drives links to near 100% utilization.

SDN can greatly facilitate big data acquisition, transmission, storage, and processing and big data will impact the design and operation of SDN [3]. With the help of big data technology, we can use many multi-objective optimization algorithm [4–8] to solve the consumption of the computing resources of the system in the case of network security and network quality of service (QoS).

Wireless communication technologies have made great progress in recent years [9–21]. In order to enhance the experience of mobile users, [9] proposes a wireless interface scheduling algorithm to select proper wireless interfaces for a set of data-dependent sporadic tasks. [10, 12] aims at solving the problem of wireless interference in the wireless network. A multi hop ad hoc network system is formed by wireless communication technology in wireless sensor networks (WSN). WSN is widely used in military, intelligent transportation, environmental monitoring, medical and health, etc. The energy problem in WSN is mentioned in [14]. The problem of fault tolerance in WSN is mentioned in [22] and the problem of relay node placement in WSN is mentioned in [23]. In [17], how to integrate device-to-device communications in the framework of SDN and wireless network virtualization [24] is studied.

Distributed Denial of Service (DDoS) flooding attacks is the main method to destroy the availability of the server or the network. Many attackers in different locations continuously send a great deal of packets at the same time, which is out of the target device's processing ability, making the legitimate user out of service.

At the same time, DDoS attacks are constantly growing. The growth rate is amazing, especially with the combination of cloud computing environment. The characteristics of cloud computing environments [25, 26] (e.g,.

on-demand self-service, broad network access and rapid elasticity, resource pooling, rapid elasticity and Measured Service, etc.) result in the DDoS attacks in cloud computing environment continue to grow and become more difficult to cope with.

What will happen when SDN meets DDoS? The SDN controller can be taken as the brain of the network, which holds the information of the whole network and makes decisions to deal with the network flows. But it is now facing the problem of a single point of failure if it is made unreachable by a Distributed Denial of Service (DDoS) Attack.

Different from wired SDN network, the wireless SDN network is more complex due to the the physical links between the devices are not deployed a priori, but depend on radio configuration. [27] presents a vision of a software-defined multi-technology network architecture (SDN@home) which enables Wireless protocols and features are no longer tied to specific technologies but can be used by general-purpose wireless SDN devices. However, because the secure channel for the control protocol in communication between wireless SDN controller and wireless SDN devices is exposed in the attacker's field of vision. This enables attackers can launch DDoS attacks towards SDN network from multiple layers (infrastructure layer or control layer). Attackers can launch DDoS attacks towards wireless SDN controller or any wireless SDN devices by forged traffic flows or control and management signals instead of launching the attacks through wired interface of the certain SDN device.

Some solutions have been proposed to defeat DDoS attacks towards SDN network recently. Nhu-Ngoc Dao *et al.* [28] proposes the approach of source based IP filtering technique to defeat DDoS attack. It works well when the attack traffic is not very huge. But to use it, two parameters for the detection method need to be initiated first by surveying the network. The effect of the method maybe affected by the artificial parameters. Lim, S *et al.* [29] propose a DDoS blocking application which is a lightweight application running on the popular SDN controller, POX [30]. The application tries to distinguish the bot and the legitimate host, logically moves its service from the attacked address to a redirected address. However, the application proposed can only apply in defending DDoS attacks target a web server, it lacks generality.

Entropy is a mathematical formula of information measure, which indicates the probability of an event happening with respect to the total number of events.

Compared with other DDoS detection methods [31], detecting by entropy is proved to have many advantages, such as more simple, higher sensitivity, lower rate of false positives, no additional network traffic, no extra hardware cost, etc. Seyed Mohammad Mousavi *et al.* [32] propose a solution to detect

DDoS attacks based on the entropy variation of the destination IP address. But the method only takes one factor into consideration, ignoring that there are many factors can be used to identify DDoS attack.

In this paper, we present a new method which is based on fuzzy synthetic evaluation decision-making model. The detection method takes many characteristics of network flows into consideration, such as packet rate, the entropy of destination IP address and source IP address, the entropy of destination TCP port to make a comprehensive judgment according to these factors. The detection method is also a lightweight process in case of excessive usage of the resource of the controller, especially when the DDoS attacks target the SDN controller. To test the solution, the paper also proposes three kinds of DDoS attacks specialized for SDN network and introduces two kinds of DDoS attacks inherited from traditional network. Every attack has been tested with the detection method, the simulation result shows its efficiency in detecting most of the DDoS attacks proposed.

This paper is organized as follows. The related work and background about SDN and DDoS attacks are reviewd in Section 2. Section 3 presents how attackers can exploit SDN network vulnerabilities to launch DDoS attacks and three DDoS attacks towards SDN network are proposed and two kinds of DDoS attacks inherited from traditional network are introduced. Section 4 presents the DDoS detection method based on fuzzy synthetic evaluation decision-making model. Finally, Section 5 presents the simulation results, comparable experiment, discussion and something about the method applied in the real environment followed by the conclusion in Section 6.

2 RELATED WORK AND BACKGROUND

From the evolution process of SDN, we can realize that although every manufactory or organization has different views on protocol standard and network architecture when taking their interest into consideration, their understanding of the core idea of SDN is similar. They all think the core idea of SDN is separating the control plane and the data plane and using the central controller to achieve the task of network programming. The controller communicate with the upper application layer and the underlying infrastruction layer with the north bound interface protocol and the south bound interface protocol. One of the most famous south bound interface protocols is OpenFlow [33].

For SDN-based networks, there are a number of challenges that need to be resolved, especially in security matter. In OpenFlow SDN network, switches do not process incoming packet as traditional way. They simply look for if the incoming packet is matched in their forwarding tables (Flow Table) or not. If there is no match, packets are sent to the controller for processing. Then the

controller installs a flow rule on the switch, the following packets can be forwarded through the rule installed. In this situation, the SDN controller plays the important role in the whole network, it can easily become a potential target of attackers. Attackers may forge a lot of packets that cannot be matched by the flow table in the OpenFlow-enabled switch, and make the controller busy dealing with the useless flooding in packets. It is a good chance for an attacker to deplete the resources and threaten the network availability.

In [34], DDoS attacks towards traditional networks and some defense mechanisms are proposed. The motivation of attackers in launching DDoS attacks is involved. The paper categorizes DDoS flooding attacks into two types based on the protocol level that is targeted: network/transport-level attacks and application-level attacks. They also categorizes the defense mechanisms for DDoS flooding attacks based on the location where prevention, detection, and response to the DDoS flooding attacks occur and based on the time when they prevent, detect, and respond to DDoS flooding attacks. After comparison, they proposed to defend DDoS with a comprehensive distributed and collaborative defense solution.

In [25], Infrastructure layer DDoS attacks, control layer DDoS attacks and application layer DDoS attacks are proposed. For application layer, attackers can launch DDoS attacks by attacking application or northbound API. Lauching Control layer DDoS attacks by attacking controller or northbound API or southbound API or westbound API or eastbound API and launching infrastructure layer DDoS attacks by attacking switch or southbound API. Also some available solutions towards each layer's DDoS attacks are put forward.

There have been proposed solutions to defeat DDoS attacks towards SDN network recently. Nhu-Ngoc Dao *et al.* [28] propose the approach of source based IP filtering technique to defeat DDoS attacks. The approach try to distinguish three kinds of Users. The malicious user who has fix source IP address and injects spoofed packets to the switch infinitely. The DDoS attacking user sends spoofed packets to the switch infinitely. The frequent user acts as normal user. The method distinguishes them and processes differently according to different users. It works well when the attack traffic is not very huge. But to use it, we need to survey the network first and initiate two parameters for the detection method, the minimum number of packets per connection of a frequent user is denoted by n and the average number of connections which the frequent users establish is denoted by k. The effect of the method may be affected by the artificial parameters. When involved with the behavior of the artificial, the uncertainty of the detecting result will increase.

Lim, S *et al.* [29] propose a DDoS blocking Application which is a lightweight application running on the popular SDN controller, POX [30]. The application tries to distinguish the bot and the legitimate host, and

redirects the legitimate clients to the real web server port, logically moves its service from the attacked address to a redirected address. The legitimate clients are asked by the server to access the service at the redirected address. The application proposed can only apply in defending DDoS attacks target a web server, it lacks generality.

Seyed Mohammad Mousavi *et al.* [32] propose a solution to detect DDoS attacks based on the entropy variation of the destination IP address. Although it is a lightweight and effective detection method. But detecting DDoS attacks, we cannot only take one factor into consideration, since there are many factors can be used to identify DDoS attacks. The detection method lacks of comprehensive consideration of multi-factors.

3 HOW TO ATTACK SDN NETWORK BY DDoS?

This part, we first introduce how to perform the DDoS attacks towards OpenFlow SDN network and then put forward five different kinds of DDoS attacks towards SDN network (e.g., SDN controller and SDN OpenFlow-enabled switch).

Launching a DDoS attacks towards SDN network, the first step we need to do is to identify if the network we want to attack is SDN architecture or not. Recall that, in most of the traditional networks the network device just like switch or router, the control plane and the data plane are coupled, so the Forwarding Information Base (FIB) is pre-configured. Therefore, it directly process the packets and needs no extra time to create a new flow entropy for a new incoming packet. While in SDN, if the first new incoming packet is not matched by the flow table in the switch, it takes more time for the packet be forwarded to the SDN controller for further processing compared with the following packets. Based on this knowledge, attackers can identify SDN network or traditional networks by checking the different response time between the first packet and the following packets [35]. We can use the SDN scanner proposed in [35]. After collecting the samples using SDN scanner, we can fingerprinting a SDN network by employing statistical testing methods, such as t-test or other more advanced statistics or machine learning techniques to improve the accuracy.

After identifying the SDN architecture, the attackers can inject a great deal of random packets into SDN network. Because the fake packets are not matched by the flow table in the switch, the SDN controller will busy dealing with a great number of new incoming packets and generates corresponding flow entries harder and harder, furthermore more and more newly installed flow entries will occupy the whole flow table in the switch very soon. Then the legitimate packets cannot be process normally.

According to the packet processing procedure of OpenFlow SDN architecture networks, we know that DDoS attacks towards SDN network can be more flexible and diverse due to the different network packet processing of SDN network architecture compared with traditional one. Therefore, we proposed two kinds of DDoS attacks to SDN network according to its different targets.

Before presenting the DDoS attack methods, we have to make it clear that the relationship between the SDN's three layers and how they interact with each other when the DDoS attack occurs. Regardless of the attacker's target is SDN controllers or SDN switches, eventually DDoS attacks flow will be sent to the SDN controller in large amounts due to not being matched with the flow table of the switch, so as to play the effect of DDoS attacks on the SDN controller. Since SDN controller is responsible for the packet forwarding decision of the whole SDN network, the SDN network may be faced with the problem of single point failure. However, the most important prerequisite to solve this problem is to detect whether there is a DDoS attack towards SDN network.

3.1 DDoS attacks towards SDN controller

In SDN architecture networks, the SDN controllers can be taken as the brain of the whole network due to its key position in combining the intelligence of the upper application layer and the performance of the underlying infrastructure layer. In this case, the SDN network may take a risk of a single point failure if the SDN controller being taken as a target of the DDoS attackers. Even if we deploy multiple controllers in the cloud [36], for the control layer is taken as the target of the attack, the attackers will continue to attack the next working controller, which will affect the performance of the SDN network. According to the characteristics of SDN network architecture, several threat vectors that may enable the exploit of SDN vulnerabilities to launch DDoS attack towards SDN network [37]. The following methods can launch DDoS attacks towards SDN controller:

Attacks inherited from traditional network

The following two attack methods are the DDoS attacks inherited from the traditional network, we can apply them to launch DDoS attack to SDN network too. Due to the SDN controller is the core of the whole network, the damage is greater than traditional DDoS attacks.

1. *Forged Source IP TCP FLOOD Attack*:
   Theory: Due to the OpenFlow-enabled switch use the flow table to match the incoming packets, the unmatched packets will be forwarded to the controller, attackers can easily launch DDoS attacks towards SDN controller

by sending a great deal of unmatched forged Source IP packets making the controller busy processing malicious attack packets and the legitimate packets cannot be processed in time. In this case, most of the unmatched forged packets will flood to the controller request for processing causing SDN controller busy processing forged flows, while a portion of the attack flow will be successfully installed on the flow table of the switch. According to the characteristics of traditional network TCP flood attack, these flows play the effect of the traditional network TCP flood attack on the server in SDN network. This attack will make the target server holding many half-open TCP connection and make the server out of work too.

2. *Forged Destination Port UDP FLOOD Attack*:
   Theory: The same as Forged Source IP TCP FLOOD Attack. By sending many forged packets to the switch, the unmatched packets will be forwarded to the controller causing the SDN controller busy processing malicious packets and cannot process legitimate packets in time. Most of the unmatched forged packets will flood to the controller causing DDoS attacks against SDN controller. while a portion of the attack flow successfully installed on the flow table of the switch will cause traditional UDP flood attack to the server in SDN network. According to the characteristics of traditional network UDP flood attack, when the server receive UDP packets, it has to hand the packets to the upper application for further process, if no suitable application found, the server will send ICMP packet back to the forged source hosts means the suitable application for the UDP packet is not found. If more and more forged UDP packets target a server will make the server busy doing this kind of work, making the server or the data communication link out of work too.

Attacks based on the new features of SDN

The following two attack methods are the DDoS attacks based on the new features of SDN, that is the controller is responsible for handling the first network flow. In this case, we propose the attack method as follows.

1. *Malformed Packets FLOOD Attack*:
   Theory: The data to be sent must be divided into packets and encapsulated through seven layers, from the upper application layer to the underlying data link layer. When forging packets, attackers can launch DDoS attacks towards SDN controllers by making the data link layer encapsulate wrong to form the malformed packets which of course cannot be matched by the OpenFlow-enabled switch and be sent to the SDN controller, thus causing DDoS attacks towards SDN controller.

2. *Uncommon Protocol Packets FLOOD Attack*:
   Theory: In order to generate packets that cannot be matched by the flow table in OpenFlow-enabled switch, attackers can launch DDoS attacks towards SDN controllers by generating some Uncommon Protocol Packets, just like Extensible Authentication Protocol (EAP) making the uncommon protocol packets cannot be matched and forwarded to the SDN controller continuously.

3.2 DDoS attacks towards OpenFlow-enabled Switch

Theory: Due to the OpenFlow-enabled switch use the flow table to match the incoming packets, the header of the unmatched packets will be sent to the controller to request for further processing. In this case the payload of the unmatched packets will left in the switchs memory until a new flow rule sent back from controller and installed. It is easy for attacker to launch DDoS attacks by sending a great of forged high payload packets, this will set up many new and unknow rule in the switch and the payload of unmatched packets will occupy the memory of a switch further affect the performance of the switch. The SDN controller will be affected greatly by the request of a great of forged high payload packets too. Thus making the legitimate flows cannot be processed in time.

4 DETECTION METHOD BASED ON FUZZY SYNTHETIC EVALUATION DECISION-MAKING MODEL

4.1 Fuzzy synthetic evaluation decision-making

Fuzzy synthetic evaluation method is a comprehensive evaluation method based on fuzzy mathematics. Based on the theory of membership degree in fuzzy mathematics, the synthetic evaluation method is used to transform the qualitative evaluation into quantitative evaluation. That is to say, fuzzy synthetic evaluation method is used to make an overall evaluation of things or objects which are restricted by many factors. It has the characteristic of clear result and strong systematicness, so it is suitable for solving various kinds of uncertainty problems such as the detection of DDoS attacks.

4.2 Mathematical model

The mathematical model of fuzzy comprehensive decision making is made up of three factors:

1. Factor set $U = \{u_1, u_2, \cdots, u_n\}$.
2. Judgment set $V = \{v_1, v_2, \cdots, v_m\}$.

3. Judgment of single factor and built of fuzzy relation matrix $R = (r_{ij})_{n\times m}$.

$$\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{pmatrix}$$

R is also called single-factor judgment matrix. r_{ij} suggests the degree to which the factor u_i in Factor set U is affiliated with v_j in Judgment set. (U, V, R) makes a comprehensive decision model, U, V, R are the three essential parts of this model.

4. Comprehensive judgment. For weight matrix of multiple factors in Factor set U, $W = \{w_1, w_2, \cdots, w_n\}$, $\sum_{i=1}^{n} w_i = 1$. Taking max-min compositional operations, that is using model $M(\bigwedge, \bigvee)$, we can get the comprehensive judgment $\underset{\sim}{B}$

$$\underset{\sim}{B} = W \circ R$$

4.3 DDoS detection method based on synthetic evaluation decision-making

1. ***Detection Principle***:

We cannot tell a network is being 100% DDoS attack or no DDoS attacks. For example. When DDoS attacks, the network service suffered slight influence and still can provide normal service. In this case, we can say 15% DDoS happen instead of saying there is not any DDoS attacks. DDoS attacks is a vague description, this is a question in Fuzzy math field.

No matter in SDN network or traditional network, some characteristics of network flows will be quite different when DDoS attacks happens. We can apply some representative characteristics of network flows, these characteristics are the factors in Factor set U and they are the main factors to judge whether DDoS happen or not.

Since there are more than one of the main factors to judge DDoS attacks, the problem is how to how to quantify each factor? For example the destination IP address can be taken as one factor to judege DDoS attacks, but how to quantify it to deacribe DDoS attacks? The answer is entropy. It is known to us that entropy indicates the probability of an event happening with respect to the total number of events. In this case, we can quantify this factor by applying the entropy of destination IP address. That is to say we can quantify every factor in the Factor set U.

When DDoS attacks occur, it needs a comprehensive evaluation method to make a comprehensive evaluation of the DDoS attacks which can be judged by many factors. Fuzzy synthetic evaluation decision-making model is what we need. With the help of the model, we can make a comprehensive evaluation of the DDoS attacks towards SDN network over many factors.

2. ***Detection Method***

As the description of mathematical model in 4.2. Firstly we should find some representative characteristics of network flows when DDoS attack to make up Factor set U. Because when DDoS attack happens the flow request rate will become greater and the destination IP address will become more converged and the source IP address will become more dispersed. Considering some DDoS attacks only target an special application, the destination TCP port will become more converged too. We choose Factor set $U = \{FRR, DstPort, DstIP, SrcIP\}$.

- *FRR(Flow Request Rate)*: The number of packets go to the SDN controller per second.
- *DstPort*: Convergence degree of destination TCP port, described with entropy.
- *DstIP*: Convergence degree of destination IP address, described with entropy.
- *SrcIP*: Divergence degree of source IP address, described with entropy.

Also, in order to make an objective judgment we need to set an weight matrix $W = \{w_1, w_2, w_3, w_4\}, \sum_{i=1}^{n} w_i = 1$ which means the weight of *FRR, DstPort, DstIP, SrcIP* is w_1, w_2, w_3, w_4. We can set the value according what we need.

Secondly, for Judgement sets $V = \{v_1, v_2, \cdots, v_i, \cdots\}$, v_i means the judgment result of the ith T (T is the interval time for detection), which contains the *FRR, DstPort, DstIP* and *SrcIP*.

Thirdly, building up single-factor judgment matrix $R = (r_{ij})_{4\times k}, k = 1, 2, 3, \cdots k$ means kth T. The matrix R can be made by the relationship of U and V can be represented by membership function. Every element in Factor set U has its own membership function, as follows:
The membership function of *FRR*:

$$\underset{\sim}{C}(frr) = \begin{cases} 1 & frr \geq FRR_3 \\ \frac{frr - FRR_1}{FRR_3 - FRR_1} & FRR_3 > frr > FRR_1 \\ 0 & frr \leq FRR_1 \end{cases} \tag{1}$$

The membership function of *DstPort*:

$$\underset{\sim}{C}(edp) = \begin{cases} 1 & edp \le EDP_3 \\ \frac{EDP_1 - edp}{EDP_1 - EDP_3} & EDP_3 < edp < EDP_1 \\ 0 & edp \ge EDP_1 \end{cases} \tag{2}$$

The membership function of *DstIP*:

$$\underset{\sim}{C}(ed) = \begin{cases} 1 & ed \le ED_3 \\ \frac{ED_1 - ed}{ED_1 - ED_3} & ED_3 < ed < ED_1 \\ 0 & ed \ge ED_1 \end{cases} \tag{3}$$

The membership function of *SrcIP*:

$$\underset{\sim}{C}(es) = \begin{cases} 1 & es \ge ES_3 \\ \frac{es - ES_1}{ES_3 - ES_1} & ES_3 > es > ES_1 \\ 0 & es \le ES_1 \end{cases} \tag{4}$$

The parameters in the formulas above FRR_1, FRR_3, EDP_1, EDP_3, ED_1, ED_3 and ES_1, ES_3 is respectively measured with slight DDoS attacks (level one DDoS attacks) and severe DDoS attacks (level three DDoS attacks). In simulation, we launch the attack of each kind and each attack lasts 15 T, take the average value as the parameter. This phase is called the initialization of DDoS attack detection method, will be describe in 5.3 Test Case.

After the membership function is successfully built, we can describe single-factor judgment matrix R as follows:

$$R = \begin{pmatrix} \underset{\sim}{C}(frr_k) \\ \underset{\sim}{C}(edp_k) \\ \underset{\sim}{C}(ed_k) \\ \underset{\sim}{C}(es_k) \end{pmatrix}, k = 1, 2, 3, \cdots, k \text{ is the } kth\ T$$

Finally, comprehensive judgment. By applying the model of weighted average $M(\cdot, +)$

$$\underset{\sim}{B} = W \cdot R = w_1 \underset{\sim}{C}(frr_k) + w_2 \underset{\sim}{C}(edp_k) + w_3 \underset{\sim}{C}(ed_k) + w_4 \underset{\sim}{C}(es_k),$$

$$k = 1, 2, 3, \cdots, k \text{ is the } kth\ T$$

Detecting DDoS attacks targeted SDN controller or a server in SDN network by running a lightweight application on the SDN controller. The application counts the number of packet flooding into the SDN controller to get the packet rate and statistical occurrences of data packets according to destination IP address, TCP port and source IP address and calculates the entropy of destination IP address,TCP port and source IP address in a slide window time(made up of several interval time T). By applying the fuzzy synthetic evaluation decision-making model described above, we can get the *Comprehensive Judgment Scores* $\underset{\sim}{B}$ of each interval T. The *Comprehensive Judgment Scores* is between 0 and 1, which represents the extent of DDoS attacks, 1 is the highest means the DDoS attack is serious, the SDN controller or the attacked server cannot provide normal service. So, by applying this method, we can monitor network traffic characteristics in real time and use these characteristics to make comprehensive judgment whether DDoS attack happens or not.

5 SIMULATION AND EVALUATION

5.1 Simulation tools and network topology

In order to validate the correctness of the logic in the proposed detection method and evaluate its performance, we choose the SDN network emulator called Mininet [38]. For SDN network simulation Mininet is a good choice, it is a network emulator specialized for SDN network simulation, we can easily model a OpenFlow-enabled switch, virtual hosts, controllers, and links for SDN network. Due to Mininet also support external controller, so we choose the network connected to a external controller POX [30]. The DDoS detection method proposed is designed on the POX and run as an application of POX controller.

To make the simulation more like the real network, we build a small topology that made up of a server and sixty legitimate hosts and a DDoS Attackers Group simulator connected to SDN network (Figure 1).

The DDoS Attackers Group is realized by Scapy [39]. Scapy is a powerful interactive packet manipulation program. It is able to forge or decode packets of a wide number of protocols, send them on the wire, capture them, match requests and replies, and much more. We use it to realize the five DDoS attacks method proposed.

5.2 Implementation and verification of the attack method based on the features of SDN

In this part, we try to give the implementation of the DDoS attack method proposed in 3.1, Malformed Packets FLOOD Attack and Uncommon

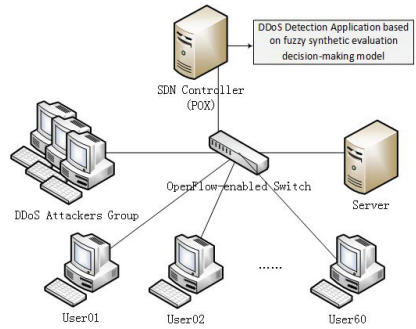

FIGURE 1
Simulation network topology

Protocol Packets FLOOD Attack and verify the feasibility of this kind of attack methods.

Malformed Packets FLOOD Attack

The formation of attack packets:

In simulation, the attack packets is realized by Scapy. We list the main code in forming malformed packets as follows:

```
Packets=UDP()/IP(dst=dstIP,src=srcIP)/
TCP(dport=dst_port,sport=src_port)
```

Due to the code of legitimate packets is:

```
Packets=Ether()/IP(dst=dstIP,src=srcIP)/
TCP(dport=dst_port,sport=src_port)
```

We changed the layer 2 protocol class of packets Ether() into The upper layer protocol class of packets UDP(). In this case, we can form the malformed packets and use to launch DDoS attack against SDN controllers.

Feasibility verifying:

In order to prove the malformed packets sent out by the attacker are really forwarded to the SDN controller when the flow table in Openflow-enabled switch cannot match the malformed packets. In simulation, we launching Malformed Packets FLOOD Attack at different rate. Such as R15% (taken as level one DDoS attacks that is slight attack), R25% (taken as level two DDoS attacks) and R50% (taken as level three DDoS attacks that is severe attack) rate attack (In order to describe the degree of DDoS attacks, we define it. Such as R15% rate attack means the ratio of attack packets with normal packets in a interval time T equals 0.15). As Figure 2 the packet forwarding rate (PFR) means the number of unmatched packets forwarded to the SDN

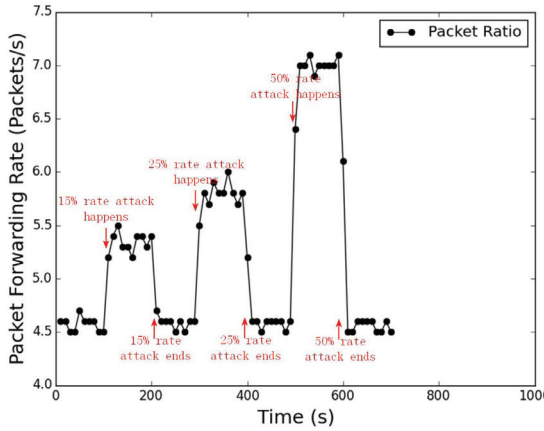

FIGURE 2
Different packet forwarding rate when Malformed Packets FLOOD Attack

controller per second. We list the packet forwarding rate corresponded to different attack rate.

From Figure 2 we can see the packet forwarding rate (PFR) is around 4.6 packets per second when there is not malformed packet attack. The PFR rose up to around 5.4 packets per second when R15% rate malformed packet attack and the change of PFR when the attack rate rose to R25% and R50%. The result verify the effectiveness of this attack.

Uncommon Protocol Packets FLOOD Attack

The formation of attack packets:

In simulation, the attack packets is realized by Scapy. We list the main code in forming malformed packets as follows:

```
Packets=Ether()/EAPOL()/EAP()
```

We try to form attack packets based on a network port authentication protocol - Extensible Authentication Protocol over LAN (EAPoL). Of course the packets cannot be matched and forwarded to the SDN controller causing the DDoS attack against SDN controller.

Feasibility verifying:

Similarly, in order to prove the uncommon protocol packets sent out by the attacker are really forwarded to the SDN controller when the flow table in Openflow-enabled switch cannot match the malformed packets. In simulation, we launching uncommon protocol packets at different rate R15% attack rate, R25% attack rate and R50% attack rate. As Figure 3, we list the PFR

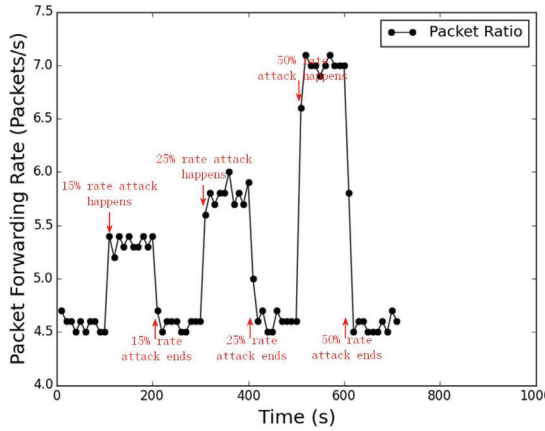

FIGURE 3
Different packet forwarding rate when Uncommon Protocol Packets FLOOD Attack

corresponded to different attack rate. The result verify the effectiveness of this attack.

5.3 Test Case

In order to facilitate the test case, we simplify the Factor set U in 4.3) Detection Method to be {*FRR,DstIP,SrcIP*} and simply set weight matrix $W = \{0.3, 0.4, 0.3\}$ which means the weight of *FRR, DstIP, SrcIP* is 0.3, 0.4, 0.3 based on our experience and the detection interval time T to be 10 seconds. Due to the attack packets launched by DDoS attack methods we introduced in 3.1 Attacks inherited from traditional network and 3.2 DDoS attacks towards OpenFlow-enabled Switch have all the characteristics the detection method needed (Factor Set U), the detection theory is the same, we only simulate the Forged Source IP TCP FLOOD Attack with three different intensities. The simulation may be divided into three phase. The first phase is get the preliminary partition of the attack levels, which help us simulate DDoS attack at different levels. The second phase is the initialization of DDoS attack detection method, the last phased is the DDoS attack detection.

1. ***Get the preliminary partition of the attack levels***
   Task: In order to simulate DDoS attack at different levels (level one, two, three), we get the preliminary partition of the attack levels by the entropy of destination IP address [40].
   Procedure: Firstly, we need to get two parameter, $baseEntropy$ (the average entropy of the destination IP address when no DDoS), d (Maximum deviation of the entropy in n times of T). Then we need to define

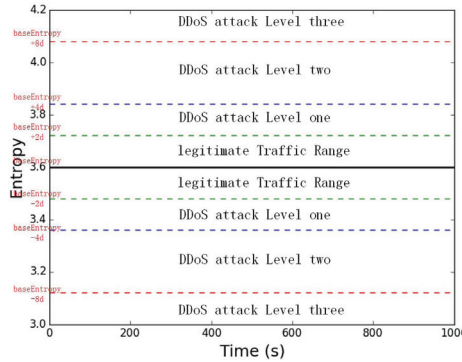

FIGURE 4
Different threat range divided according to the entropy

DDoS attack at different levels. Figure 4 shows the divided threat range in more detail. The black and thick line in the middle represents the average entropy when there is not any DDoS attacks. Any traffic falls on the range marked 'legitimate Traffic Range' will be taken as normal traffic. The threat range of level one is the range marked 'DDoS attack Level one', any traffic falls on this range will be taken as the threat level one. Similarly, any traffics falls on range marked 'DDoS attack Level two' and 'DDoS attack Level three' will be taken as threat level two or three correspondingly. We can use Scapy to adjust packet rate to make the entropy falls continuously on each threat level range, thus we can simulate DDoS attack at different levels.

2. ***Initialization of DDoS attack detection method***
Task:Initializes some parameters of the DDoS attack detection method. The parameters are the parameters of each membership functions mentioned in 4.3 2) Detection Method. We summarizes the parameters used in our experiment in Table. 1.
Procedure: Because the parameters of each membership function is initialized by launching R15% (taken as level one DDoS attacks means slight attack) and R50% (taken as level three DDoS attacks that is severe attack means the SDN controller or the attacked server cannot provide normal service) rate attack (In order to describe the degree of DDoS attacks, we define it. Such as R15% rate attack means the ratio of attack packets with normal packets in a interval time T equals 0.15), these two kinds of attack packets should be launched strictly according to the procedure. After phase one, we know how to simulate the attack of different attack levels. Firstly, when 5 normal T passed, we start to launch level one Forged Source IP TCP FLOOD Attack, the attack lasted for 15 T,

Types of parameter	Description	Value
baseEntropy	The average of the entropy in n times of T when no DDoS. We set $n = 12$ when simulation	3.5848
d	Maximum deviation in n times of T. We set $n = 12$ when simulation	0.0741
FRR_1	Membership function parameter of the flow request rate. request rate when Level one (R15%) attack	5.5667
FRR_3	Membership function parameter of the flow request rate. request rate when Level three (R50%) attack	7.2267
ED_1	Membership function parameter of the aggregation of destination IP address. The destination IP address entropy when Level one (R15%) attack	3.3563
ED_3	Membership function parameter of the aggregation of destination IP address. The destination IP address entropy when Level three (R50%) attack	2.6929
ES_1	Membership function parameter of the divergence of source IP address. The source IP address entropy when Level one (R15%) attack	5.1824
ES_3	Membership function parameter of the divergence of source IP address. The source IP address entropy when Level three (R50%) attack	5.4468

TABLE 1
The initial parameters of the network for detection method

then we get the parameter FRR_1, ED_1, ES_1(The average value of 15 T). When the network back to normal, we begin to launch level three Forged Source IP TCP FLOOD Attack, the attack lasted for 15 T too, then we get the parameter FRR_3, ED_3, ES_3. The initialization is done after we get all the parameters the detection method need in Table. 1.

3. ***DDoS attack detection***
Task: launching Forged Source IP TCP FLOOD Attack with three different levels and calculating *Comprehensive Judgment Scores* $\underset{\sim}{B}$ of each interval T and plotting the result.
Procedure: After phase one and two, we start to test the detection method. When the network came back to normal and 10 normal T passed, we start to launch level one Forged Source IP TCP FLOOD Attack, the attack lasted for 15 T. When the network came back to normal and 10 normal T passed, we then start to launch level two Forged Source IP TCP FLOOD Attack, the attack also lasted for 15 T. When the network came back to normal and 10 normal T passed again, we then start to launch level three Forged Source IP TCP FLOOD Attack, the attack lasted for 15 T too. As Figure 5, we plot the result $\underset{\sim}{B}$ of each interval T.

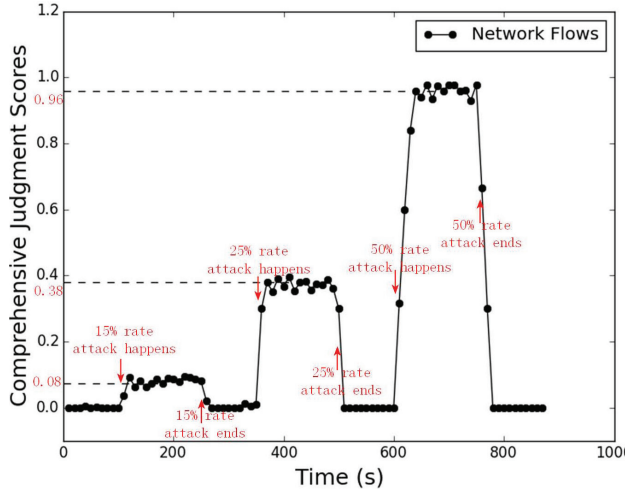

FIGURE 5
comprehensive judgment scores when different levels of DDoS attack

5.4 Evaluation and discussion

1. ***Test case analysis***

Figure 5 shows the change of *Comprehensive Judgment Scores* when different levels Forged Source IP TCP FLOOD Attack happen. From the figure we can see the *Comprehensive Judgment Scores* for DDoS is very low (nearly equals to zero) when there is no any DDoS attacks. When level one (R15%) DDoS attack happens, the *Comprehensive Judgment Scores* rises a little and stays around 0.08 which means the DDoS attack is slight. Compared to level one, level two (R25%), level three (R50%) DDoS attack *Comprehensive Judgment Scores* is around 0.38 and 0.96. Any attack more serious than level three with its *Comprehensive Judgment Scores* around 1 will be taken as the SDN controller or the server out of service and have to take mitigation measures.

Why the comprehensive judgment scores will be different to different levels of DDoS attacks? Due to the elements in the Factor set $U = \{FRR, DstIP, SrcIP\}$ in our test case are all significant indicators for judging DDoS attack and the weight matrix $W = \{0.3, 0.4, 0.3\}$ also be set for a more objective judgment, we apply the fuzzy synthetic evaluation decision-making model to make a comprehensive judge based on multiple factors. When level one DDoS attack, there is a little rise in FRR will become greater and $DstIP$ will become smaller (the destination IP address will become more converged), also $SrcIP$ will become greater too (the source IP address will become more dispersed). But all

these change is very little which leads to the *Comprehensive Judgment Scores* rises a little. Similarly, the change is greater in level two and level three DDoS attack, which makes the final score grows up.

2. ***Attacks the detection method cannot make a comprehensive judgment*** For the DDoS attack method proposed In section 3.1 Malformed Packets FLOOD Attack and Uncommon Protocol Packets FLOOD Attack, the detection method cannot make a comprehensive judgment according to all the elements in the Factor set U. That is because the controller cannot parse the special packet normally, which makes our detection method cannot make the correct statistics to the destination IP address and source IP address. In this case, for Factor set $U = \{FRR, DstIP, SrcIP\}$ in our test case, only FRR can be count, the other two factors will be set to zero and cannot make a comprehensive judgment. But the detection method also can figure out the exist of these attacks according to FRR.

5.5 Comparable Experiment

In order to reflect the detection efficiency of our proposed algorithm, in this section we compare it with the DDoS detection algorithm based on the entropy variation of the destination IP address proposed in [32]. Due to the algorithm for comparison only base on the entropy of destination IP address is less than a predetermined threshold to determine the existence of DDoS attacks. According to the shortage of the method, we propose the following two kinds of DDoS attacks, which target the whole SDN network (User01, User02 $\cdots$ User60 and Server) instead of the single server.

1. *Malicious user DoS attacks*: Malicious user with fixed IP address launches DoS attack against the entire SDN network.
2. *Forged IP DDoS attacks*: Attackers forge the source IP address and launch DDoS attacks against the entire SDN network.

We try to verify our proposed algorithm and the DDoS detection algorithm proposed in [32] by launching these two kinds of DDoS attacks. We use the same environment, the same Factor set and weight matrix W as the Test case. We simulate the two kinds of DDoS attacks above with three different intensities (R%15,R%25,R%50). The attack time and the duration time of attack are the same as 5.3 3 DDoS attack detection. Finally we get the detection results of the two detection algorithms for these two kinds of DDoS attacks.

Figure 6 shows the detection results of the two algorithms under Malicious user DoS attacks with three different intensities (R%15,R%25,R%50). We plot the *Comprehensive Judgment Scores* of our detection method and the Average Scores (Average value of the *Comprehensive Judgment Scores* during attack phase. We give up the start point and end point of the attack). From

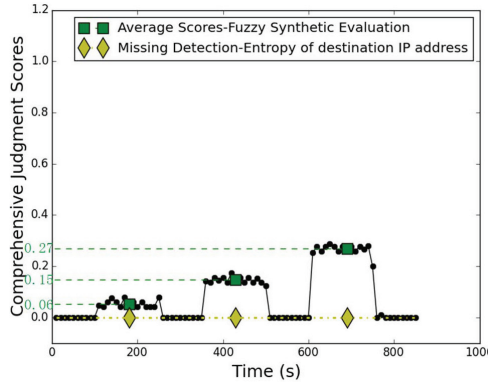

FIGURE 6
Detection Result when Malicious user DoS attacks

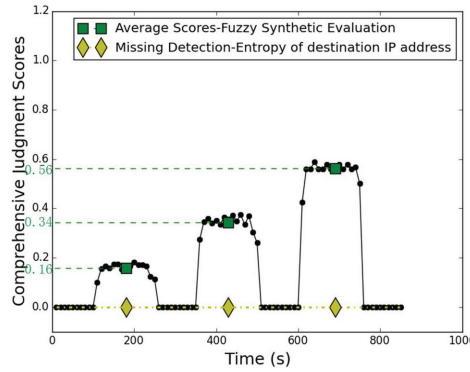

FIGURE 7
Detection Result when Forged IP DDoS attacks

Figure 6, we can clearly see that this attack bypassed the DDoS detection algorithm proposed in [32]. However, our proposed detection algorithm can detect the attack to a certain extent. Similarly, for Forged IP DDoS attacks. From Figure 7, we can see that our proposed algorithm can figure out a Comprehensive Judgement Scores to inform us of the existence of DDoS attacks, while the algorithm proposed in [32] missed the detection.

5.6 About the method applied in the real environment

1. ***Setting the weights of evaluation factors***
   In this part, we are going to discuss about how to determinate the weight matrix. In order to facilitate the test method, the default weight matrix is set to be $W = \{FRR, DstIP, SrcIP\} = \{0.3, 0.4, 0.3\}$. The weight

matrix can be determined in the real environment by applying Analytic Hierarchy Process (AHP) [41].

First of all, launching a variety kinds of DDoS attacks against the server in SDN network by applying the typical historical data of the DDoS attacks, we can get judgment matrix of evaluation factor. Secondly, by applying Bidirectional Analytic Hierarchy Process proposed in [41] to get a better weight matrix W. For example we have three kinds of DDoS attacks named $DDoS_1$, $DDoS_2$, $DDoS_3$ relatively, and three factors in Factor set U named $factor_1$, $factor_2$, $factor_3$ relatively. We conduct the traditional Analytic Hierarchy Process method and gain a group of weights. Then, we exchange the factors in the bottom layer and the DDoS types in the middle layer and conduct the reversed Analytic Hierarchy Process method and gain another group of weights. Finally, we can calculate the mean of the two groups of weights and gain a better result of weight matrix.

2. ***Initialization of the method parameters***
Based on the idea of active defense. In real environment, the initialization of DDoS attack detection method need to be determined in the training phase, which means searching for the typical historical data of the DDoS attacks and launching a variety of DDoS attacks against SDN network server are the work must be done firstly. And then based on your network or server's ability to tolerate DDoS attacks, divide the attack into three levels and use these three levels of attack data to initialize the parameters of the DDoS detection method and then apply into the real environment.

6 CONCLUSION AND FUTURE WORK

The paper proposes a DDoS detection method based on on fuzzy synthetic evaluation decision-making model. For testing the method, the paper also proposes three kinds of DDoS attacks specialized for SDN network and introduces two kinds of DDoS attacks inherited from traditional network. We tested every attack method with different attack intensity. We also make a comparable experiment to show its advantage to other DDoS detection algorithm based on single factor. Our detection method is proved to be lightweight and effective to most of the DDoS attacks.

For future work, we need to improve our detection method to make it more sensitive to some special DDoS attack packets, such as malformed packets and uncommon Protocol Packets. The discrimination of DDoS attacks and flash crowds in SDN network will be our next research direction too. I hope the method can be tested in real network for further confirm its performance and efficiency.

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. This work is supported by the National Science Foundation of China (Protecting security of mobile payment:system model and supporting techniques. Grant No. 61672358).

REFERENCES

[1] Kreutz, D., Ramos, F. M. V., Verłssimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (Jan 2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.

[2] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., *et al.* (2013). B4: Experience with a globally-deployed software defined wan. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 3–14. ACM.

[3] Cui, L., Yu, F. R., and Yan, Q. (January 2016). When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE Network*, 30(1):58–65.

[4] Liang, Z., Song, R., Lin, Q., Du, Z., Chen, J., Ming, Z., and Yu, J. (2015). A double-module immune algorithm for multi-objective optimization problems. *Applied Soft Computing*, 35:161–174.

[5] Lin, Q., Zhu, Q., Huang, P., Chen, J., Ming, Z., and Yu, J. (2015). A novel hybrid multi-objective immune algorithm with adaptive differential evolution. *Computers & Operations Research*, 62:95–111.

[6] Lin, Q. and Chen, J. (2013). A novel micro-population immune multiobjective optimization algorithm. *Computers & Operations Research*, 40(6):1590–1601.

[7] Chen, J., Lin, Q., and Hu, Q. (2010). Application of novel clonal algorithm in multiobjective optimization. *International Journal of Information Technology & Decision Making*, 9(02):239–266.

[8] Chen, J., Lin, Q., and Shen, L. (2011). An immune-inspired evolution strategy for constrained optimization problems. *International Journal on Artificial Intelligence Tools*, 20(03):549–561.

[9] Niu, J., Gao, Y., Qiu, M., and Ming, Z. (2012). Selecting proper wireless network interfaces for user experience enhancement with guaranteed probability. *Journal of Parallel and Distributed Computing*, 72(12):1565–1575.

[10] Zhang, S. and Liew, S. C. (2010). Applying physical-layer network coding in wireless networks. *EURASIP Journal on Wireless Communications and Networking*, 2010(1):1.

[11] Li, Z., Yu, F. R., and Huang, M. (Jan. 2010). A distributed consensus-based cooperative spectrum sensing in cognitive radios. *IEEE Trans. Veh. Tech.*, 59(1):383–393.

[12] Jinyi, Z., Shutao, X., Jiang, Y., Zheng, H., and Laizhong, C. (2013). Maximum multiflow in wireless network coding. *IEICE Transactions on Communications*, 96(7):1780–1790.

[13] Xie, R., Yu, F. R., Ji, H., and Li, Y. (Nov. 2012). Energy-efficient resource allocation for heterogeneous cognitive radio networks with femtocells. *IEEE Trans. Wireless Commun.*, 11(11):3910–3920.

[14] Lin, X.-H., Kwok, Y.-K., Wang, H., and Xie, N. (2015). A game theoretic approach to balancing energy consumption in heterogeneous wireless sensor networks. *Wireless Communications and Mobile Computing*, 15(1):170–191.

[15] Yu, F. and Leung, V. C. M. (Apr. 2001). Mobility-based predictive call admission control and bandwidth reservation in wireless cellular networks. In *Proc. IEEE INFOCOM'01*, Anchorage, AK.

[16] Yu, F. and Krishnamurthy, V. (Jan. 2007). Optimal joint session admission control in integrated wlan and cdma cellular networks with vertical handoff. *IEEE Trans. Mobile Comput.*, 6(1):126–139.

[17] Cai, Y., Yu, F. R., Liang, C., Sun, B., and Yan, Q. (2015). Software defined device-to-device (d2d) communications in virtual wireless networks with imperfect network state information (nsi). *IEEE Transactions on Vehicular Technology*, PP(99):1–1.

[18] Yu, F. R., Huang, M., and Tang, H. (May 2010). Biologically inspired consensus-based spectrum sensing in mobile ad hoc networks with cognitive radios. *IEEE Network*, 24(3):26–30.

[19] Attar, A., Tang, H., Vasilakos, A., Yu, F. R., and Leung, V. (2012). A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proceedings of the IEEE*, 100(12):3172–3186.

[20] Bu, S., Yu, F. R., Cai, Y., and Liu, P. (Aug. 2012). When the smart grid meets energy-efficient communications: Green wireless cellular networks powered by the smart grid. *IEEE Trans. Wireless Commun.*, 11:3014–3024.

[21] Ma, L., Yu, F., Leung, V. C. M., and Randhawa, T. (Aug. 2004). A new method to support UMTS/WLAN vertical handover using SCTP. *IEEE Wireless Commun.*, 11(4):44–51.

[22] Qiu, M., Ming, Z., Li, J., Liu, J., Quan, G., and Zhu, Y. (2013). Informer homed routing fault tolerance mechanism for wireless sensor networks. *Journal of Systems Architecture*, 59(4):260–270.

[23] Lu, K., Chen, G., Feng, Y., Liu, G., and Mao, R. (2010). Approximation algorithm for minimizing relay node placement in wireless sensor networks. *Science China Information Sciences*, 53(11):2332–2342.

[24] Liang, C. and Yu, F. R. (Firstquarter 2015). Wireless network virtualization: A survey, some research issues and challenges. *IEEE Commun. Surveys Tutorials*, 17(1):358–380.

[25] Yan, Q., Yu, F. R., Gong, Q., and Li, J. (Firstquarter 2016). Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges. *IEEE Communications Surveys Tutorials*, 18(1):602–622.

[26] Yan, Q. and Yu, F. R. (April 2015). Distributed denial of service attacks in software-defined networking with cloud computing. *IEEE Communications Magazine*, 53(4):52–59.

[27] Gallo, P., Kosek-Szott, K., Szott, S., and Tinnirello, I. (May 2016). SDN@home: A method for controlling future wireless home networks. *IEEE Communications Magazine*, 54(5):123–131.

[28] Dao, N.-N., Park, J., Park, M., and Cho, S. (Jan 2015). A feasible method to combat against DDoS attack in SDN network. In *2015 International Conference on Information Networking (ICOIN)*, pages 309–311.

[29] Lim, S., Ha, J., Kim, H., Kim, Y., and Yang, S. (July 2014). A SDN-oriented DDoS blocking scheme for botnet-based attacks. In *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 63–68.

[30] McCauley, M. (2013). About POX. *URL: http://www. noxrepo. org/pox/about-pox/. Online*.

[31] Braga, R., Mota, E., and Passito, A. (Oct 2010). Lightweight DDoS flooding attack detection using NOX/OpenFlow. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 408–415.

[32] Mousavi, S. M. and St-Hilaire, M. (Feb 2015). Early detection of DDoS attacks against SDN controllers. In *Computing, Networking and Communications (ICNC), 2015 International Conference on*, pages 77–81.

[33] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.

[34] Zargar, S., Joshi, J., and Tipper, D. (Fourth 2013). A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. *Communications Surveys Tutorials, IEEE*, 15(4):2046–2069.

[35] Shin, S. and Gu, G. (2013). Attacking software-defined networks: A first feasibility study. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 165–166. ACM.

[36] Li, H., Li, P., Guo, S., and Nayak, A. (Oct 2014). Byzantine-Resilient Secure Software-Defined Networks with Multiple Controllers in Cloud. *IEEE Transactions on Cloud Computing*, 2(4):436–447.

[37] Kreutz, D., Ramos, F., and Verissimo, P. (2013). Towards secure and dependable software-defined networks. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 55–60. ACM.

[38] (2014). Mininet. [Online]. Available: http://mininet.org.

[39] (2014). Scapy. [Online]. Available: http://www.secdev.org/projects/scapy.

[40] Zhang, J. and Qin, Z. (2010). Modified method of detecting DDoS attacks based on entropy. *Jisuanji Yingyong/ Journal of Computer Applications*, 30(7):1778–1781.

[41] Liu, J. (June 2010). Analyze the Influencing Factors of Food Security by Bidirectional Analytic Hierarchy Process. In *Computing, Control and Industrial Engineering (CCIE), 2010 International Conference on*, volume 2, pages 32–33.