IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 21, 2020, accepted August 31, 2020, date of publication September 3, 2020, date of current version September 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021435

A New Framework for DDoS Attack Detection and Defense in SDN Environment

LIANG TAN[1], YUE PAN[2], JING WU[2], (Member, IEEE), JIANGUO ZHOU[2], HAO JIANG[2], (Member, IEEE), AND YUCHUAN DENG[2]

[1]China Ship Development and Design Center (CSDDC), Wuhan 430064, China
[2]School of Electronic Information, Wuhan University, Wuhan 430072, China

Corresponding author: Jing Wu (wujing@whu.edu.cn)

This work was supported in part by the National Key Research and Development Project under Grant 2017YFC0503801.

ABSTRACT While software defined network (SDN) brings more innovation to the development of future networks, it also faces a more severe threat from DDoS attacks. In order to deal with the single point of failure on SDN controller caused by DDoS attacks, we propose a framework for detection and defense of DDoS attacks in the SDN environment. Firstly, we deploy a trigger mechanism of DDoS attack detection on data plane to screen for abnormal flows in the network. Then, we use a combined machine learning algorithm based on K-Means and KNN to exploit the rate characteristics and asymmetry characteristics of the flows and to detect the suspicious flows determined by the detection trigger mechanism. Finally, the controller will take corresponding actions to defense against the attacks. In this paper, we propose a new framework of cooperative detection methods of control plane and data plane, which effectively improve the detection accuracy and efficiency, and prevent DDoS attacks on SDN.

INDEX TERMS Software defined network, distributed denial of service (DDoS), collaborative detection, traffic characteristics, detection trigger.

I. INTRODUCTION

Software defined network (SDN) has become a revolutionary network paradigm. It can meet the growing demands of future networks, and it is increasingly used in data centers and operator networks. However, it still faces some basic security challenges, such as Distributed Denial of Service (DDoS) attacks. The controller of SDN will be separated from the rest of the network and lose the centralized control of SDN when it is under DDoS attacks. Therefore, the main advantage of SDN, i.e. centralized control of the network, can also be threatened by DDoS attacks and it is one of the most important security threats in SDN. In order to apply SDN to data centers and cloud computing environments more reliably and promote the development of future networks while ensuring network security, it is particularly important to study DDoS detection and defense technologies in SDN environments.

During the process of DDoS attack on SDN controller, a large number of packets are sent to the target network. If the source IP addresses and destination IP addresses of these packets are forged, the switches will find no matching flow entry and regard these unmatched flows as new flows. Then, the switch will send a packet_in message to the controller or directly forward the packet. The forwarding paths of these packets are determined by controller in SDN. A large amount of DDoS attack flows is hidden in legitimate traffic, which occupy the resources of controller continuously, thereby exhausting the controller's resources and making the controller unable to process newly arrived legitimate packets. As a result, the controller shuts down and the SDN architecture is lost. Disappointedly, the same challenge still exists even if there is a backup controller [2].

By analyzing the way how DDoS attacks work, it can be seen that due to the characteristics of SDN itself and the evolution of DDoS attacks, the methods of DDoS attacks in SDN environment has different characteristics from traditional networks. By analyzing the principle of the DDoS attacks on the SDN controller, we conclude the differences between them in three points:

1) Different goals. In traditional networks, DDoS attacks target one or more destination servers or network links in the network. In SDN, DDoS attacks are aimed at the controller. The purpose is to exhaust the controller's resources and cause the single point of failure in SDN.

2) Different characteristics of attack packets. In traditional networks, because the attack target is usually the terminal server of the network, the destination IP addresses of the data

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

packets is real. In SDN, to launch a DDoS attack, the attacker need to forge the destination IP addresses so that causing the controller to fall into continuously processing with new flows and exhausting the controller's resources.

3) Different results. The server resources were exhausted and unable to provide services for legitimate users in the traditional network if it suffers a DDoS attack. However, the DDoS attack eventually causes the controller to lose contact with data plane and fail to provide services for forwarding data packets in SDN.

DDoS detection and defense methods in traditional network environment are at a relatively mature stage. However, DDoS detection and defense technology in SDN environment is still in a relatively weak stage because SDN is a new network architecture. In SDN environment, DDoS attack detection and defense methods are mostly implemented by transplanting the methods in traditional networks without taking the characteristics of DDoS attack traffic in SDN environment and the advantages of SDN into account. Due to the characteristics of the SDN architecture, existing DDoS detection methods are deployed on the SDN controller. The controller needs to continuously collect information from the traffic on switches to determine whether a DDoS attack occurs, which increases both the workload of the SDN controller and the communication overhead of the SDN southbound channel. Therefore, we design a new DDoS detection and defense framework which fully considers the accuracy and efficiency of the algorithm simultaneously compared with previous methods. Specifically, the accuracy and efficiency of our method are high, and the two-level detection architecture significantly reduces the overhead of the controller's southbound interface, which is more straightforward and effective than previous methods [11]. Our contributions are summarized as follows:

1) We deploy a DDoS detection trigger mechanism on the data plane. This mechanism uses the CPU resources of switches to count the sending rate of *packet_in* messages on switches. Once it finds there may be a DDoS attack, it alerts the controller to detect the abnormality so that the controller can response the detection trigger mechanism rapidly.

2) We deploy a machine learning based detection mechanism on the controller. After receiving the alarm, the controller extracts the traffic and uses a combined machine learning algorithm based on the K-Means and KNN to detect the suspicious traffic reported by the detection trigger mechanism.

3) We design a data plane and control plane collaborative defense method, which takes different actions to attack flows and legitimate burst flows and releases the resource occupied by attack flows immediately.

3) We conduct comprehensive experiments to prove the effectiveness and efficiency of our methods. Experimental results in the simulation network show that the method performs well in terms of accuracy and efficiency.

The rest of the paper is organized as follows. In Section 2, we summarize the related work in this field. In Section 3, the workflow and implementation of the DDoS attack detection method, including detection trigger module, flow feature extraction module and attack detection module, and DDoS attack defense mechanism are elaborated in detail. In Section 4, we discussed the data plane and control plane cooperative detection and defense framework in detail, then describe the implementation and experimental results. Finally, we summarized this paper and discuss future work in Section 5.

II. RELATED WORK

A. RESEARCHES ON DDoS DETECTION AND DEFENSE METHODS

In SDN, most of the existing DDoS attack detection methods are transplantation or transformation of methods used in traditional networks. The algorithm based on statistical information entropy is a common DDoS detection method. This method can quickly process a large amount of traffic data with little cost of calculation, but its accuracy relies on the selection of the threshold and it has certain one-sidedness. Kalkan et al [4] proposed an entropy-based joint scoring system (JESS) to detect and mitigate DDoS attacks. It uses joint entropy as a tool to detect DDoS attacks without significantly increasing the workload on the switches. Lima *et al.* [5] introduced a method to more effectively protect the network from DDoS attacks through statistical analysis of traffic entropy, and built a model in the Mininet for verification. Kumar *et al.* [6] provided a solution that can effectively detect and mitigate SYN flood attacks in SDN. It starts with the entropy calculation of the destination IP addresses, then uses a set of selected TCP flags as random variables, and finally identifies the attacker through an adaptive threshold.

The network anomaly detection algorithm based on machine learning can also be effectively applied to DDoS attack detection in SDN [41]. Machine learning algorithms can automatically build classification models based on training data, and classify traffic based on the features of flows. Literature [7] proposed a lightweight DDoS attack detection algorithm based on traffic features. The algorithm uses a NOX controller to process switch information and performs traffic analysis based on self-organizing map (SOM). SOM is an unsupervised, competitive learning artificial neural network (ANN), which can achieve lightweight DDoS detection. In addition, the k-nearest neighbor algorithm (KNN) is also a simple and effective machine learning algorithm, which classifies flows by measuring the abstract distance between traffic feature vectors. Peng *et al.* [14] proposed the abnormal traffic detection algorithm, DPTCM-KNN. This algorithm can effectively improve the accuracy of abnormal flow detection, and at the same time reduce the false alarm rate in the process of DDoS detection. Although many researchers have proposed various solutions based on machine learning

algorithms for DDoS detection in SDN, these methods still have problems in accuracy and efficiency.

Therefore, some researchers have proposed original DDoS detection methods to improve the accuracy and efficiency of detection, and have achieved certain results.

Cui *et al.* [1] improved the detection mechanism and introduced attack detection trigger mechanism for the first time, which can respond to DDoS attacks faster and reduce the workload of controllers and switches. This method effectively solves the limitation of fixed detection, but the trigger mechanism is deployed on the control plane which increases the workload of the controller.

Zang *et al.* [9] started from the flow features and proposed a finer-grained and more comprehensive flow index set. The author extracted 9 single attributes and 39 dual attributes from multiple dimensions such as time, space, category and intensity to form the IP address traffic behavior features spectrum. The fine-grained traffic features greatly improve the detection accuracy.

Xu *et al.* [10] improved the DDoS detection algorithm. They proposed a DDoS detection method based on K-FKNN and a module detection system to improve detection efficiency and accuracy.

In addition, some researchers reduced the channel overhead between the data plane and the control plane by improving the flow data collection method, and improved the detection efficiency. In literature [11], the author uses sFlow technology to sample network traffic, which is compared with SDN's own flow table collection method based on OpenFlow (OF) protocol and effectively reduces the overhead of control plane.

B. SDN AND DATA PLANE PROGRAMMABLE TECHNOLOGY

In SDN, the data plane is responsible for processing and forwarding the packets and collecting status of the switches. The core equipment of the data plane is switches which is different from the forwarding equipment in the traditional network. The switches used in the SDN data plane only retains the forwarding function and focuses on the high-speed forwarding of packets. Because the data plane and control plane in SDN are completely separated, the forwarding strategy of all packets is delivered by the controller to the switch through the southbound interface protocol, and the management and configuration of the entire network is also the responsibility of the controller. This reduces the complexity of the forwarding equipment and greatly improves the efficiency of network management and control.

The existing software defined network architecture can provide users with the controllability of the control plane and application plane, which makes the entire network flexible and extensible. In traditional SDN, the packet parsing and forwarding process on switches is determined by the device forwarding chip, so switches do not have scalability in the forwarding protocol. And the cost of developing a forwarding chip that supports a new protocol or an extensible protocol is very high. Therefore, this mode of binding switch functions and protocol support to hardware limits the innovative development of the network.

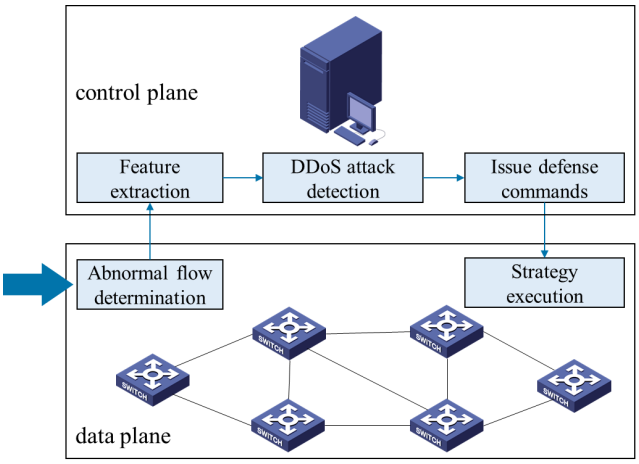


FIGURE 1. How trigger mechanism, detection, and mitigation coordinate in SDN environment.

Therefore, it becomes the trend of the new generation of SDN that the devices on the forwarding plane also have certain programmability so that developers can define global networks and network devices through software. The emergence of the P4 language provides developers with network programmable capability which breaks the restrictions of hardware devices on the underlying network, so that the parsing and forwarding process of packets can also be configured through programs and the overall network devices on the network can be truly open to users from top to bottom.

III. DDoS DETECTION AND DEFENSE IN SDN ENVIRONMENT

The DDoS detection and defense framework we proposed includes the trigger mechanism for DDoS attacks detection on data plane, the combined machine learning algorithm based on K-Means and KNN on the controller for detecting the suspicious flows found by the detection trigger mechanism and the DDoS attack defense mechanism. Overview of the framework is shown in Fig. 1. In the following of this section, we will discuss the implementation of the framework in detail.

A. DDoS DETECTION TRIGGER MECHANISM ON PROGRAMMABLE DATA PLANE

Many researchers have proposed effective DDoS detection methods. However, most existing DDoS attack detection methods use fixed time intervals to detect network traffic. This method performs detection through periodic triggers, in this case, the detection cycle period greatly affects the efficiency of DDoS attack detection and the performance of controller. If the detection period is too long, the response time will be very long too, then the controller and switches have to process a large number of packets which increases the workload of the controller and switches. It also increases the pressure on subsequent DDoS attack defense if the network

cannot respond DDoS attacks in a timely manner. On the contrary, if the detection period is too short, the controller will start attack detection frequently which will occupy a lot of resources of the controller and increase the workload of the controller. It will surely affect other tasks running on the controller and increase the communication overhead between the controller and switches. Apparently, there exists deficiencies in the methods of periodically triggered DDoS attack detection in the SDN environment. In order to solve this problem, we introduce a detection trigger mechanism in this paper which determines the start time of DDoS attack detection through the trigger mechanism. This method realizes the rapid response of the detection module and breaks the limitations of the regular trigger detection method.

In the past, some researchers proposed the idea that deploying a detection trigger mechanism on the controller, but this will increase the workload on the controller and occupy the resources of the controller and the southbound channel. In SDN, the switches only exist as a forwarding device of the data plane, and the resources on them are only used to store and forward rules and send messages to the controller. No complicated work is assigned to them, which means there are a lot of idle computing resources on SDN switches, and reasonable use of the resources can effectively reduce the detection pressure of the controller without affecting the performance of the switches. Therefore, deploying the detection trigger mechanism on the switches can improve the detection performance while reducing the workload of the controller.

In addition, the controller detects DDoS attacks later than switches because the attack traffic will reach the switches first. Therefore, deploying the detection trigger mechanism on the data plane can reduce the controller resource occupation and the communication overhead between the controller and the switches. Considering all the advantages, in this work, we deploy the detection trigger mechanism on the data plane. The CPU resources on the switches are used to count the data packets passing through them, and the packet arrival rate are used as the detection trigger standard.

The detection trigger mechanism in this paper is deployed on the data plane and is implemented by counting packet_in messages on switches. When launching a DDoS attack on the SDN controller, the attacker will send a large number of packets with forged IP addresses to the network. After the data plane switches receive these packets, the source IP addresses and destination IP addresses of these packets cannot match the flow entries in switches, and switches cannot determine the paths of these packets. Consequently, the switches have to send requests to the controller to get the forwarding paths of these packets, so packet_in messages will be generated and sent to the controller. After receiving the packet_in messages, the controller will process the messages and take responses, such as adding a new flow entry to the switches, sending packet_out messages to inform the switches to perform proper operations, etc. If no action is taken by the controller, the switches will continuously send packet_in messages to the controller as shown in Fig.2.

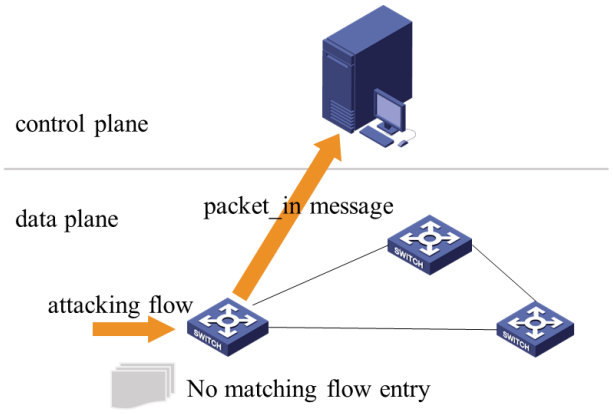


FIGURE 2. Schematic diagram of DDoS attack.

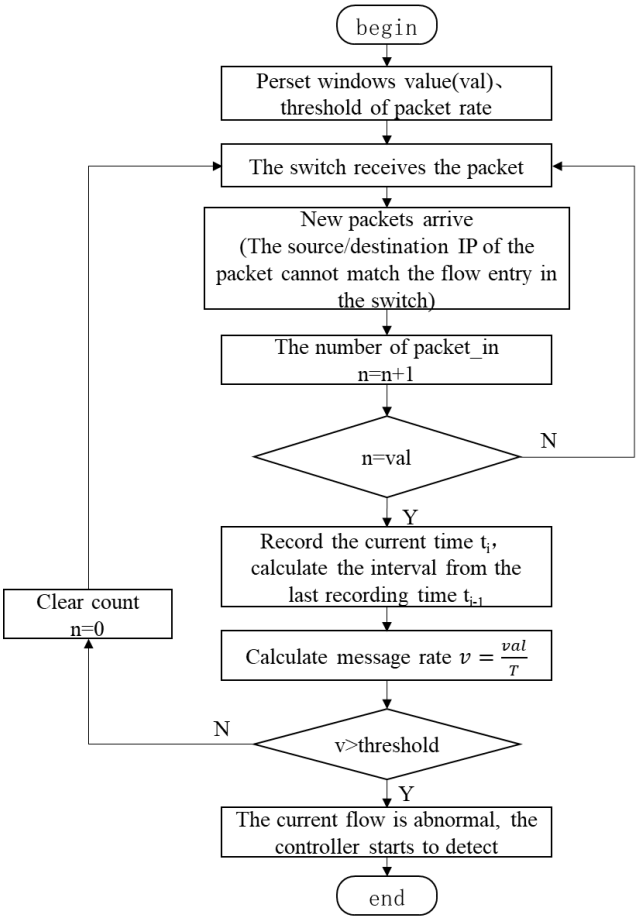


FIGURE 3. Detection trigger mechanism process.

Therefore, when there is a DDoS attack against the controller, the number of packet_in messages sent by the switches will increase sharply, so the abnormal rate of packet_in messages can be used as a sign of the start of DDoS attacks. When the number of packet_in messages increase sharply, it means that there are suspicious flows appearing on the switches, and the controller will start the process of attack detection. The process of detection trigger algorithm we proposed in this paper is shown in Fig.3.

B. FLOW FEATURE EXTRACTION

In the classification based DDoS detection method, the flow features determine the basis of classification of the attack flow and normal flow, so the selection of features greatly affects the accuracy of the algorithm. In this section, we make a detailed analysis of the way how an attacker launching a DDoS attack on the SDN controller, and find the features that can distinguish the attack flows more accurately.

First of all, the goal of DDoS attacks is to crash the network equipment or congest the links which makes the network cannot provide users with normal services. Therefore, for all types of DDoS attacks, in order to achieve the effect of exhausting resources of network devices or occupying bandwidth of links, the attacker needs to send a large amount of traffic to the network in a short time, and the average number of bytes that pass through the switches per unit time will be much higher than normal. Based on these, we find the average byte stream rate and stream duration are the primary indicators that distinguish DDoS attack flows from normal flows.

Second, traffic asymmetry is also an important feature of DDoS attacks. If the source address of a flow is host A and the destination address is host B, then the flow with source address B and destination address A is a symmetric flow of the former flow. Normally, communication between hosts in the network is often bidirectional, so the percentage of symmetric traffic is very high. When a DDoS attack occurs, the number of single flows will increase dramatically, and the percentage of symmetric flows will decrease significantly.

Third, in real world, the surge of network services will cause traffic bursts and generate a large number of packets. This kind of legal burst traffic is very similar with the traffic caused by DDoS attacks. Therefore, in order to ensure the normal service of legitimate users, it is necessary to distinguish between normal traffic surge and DDoS attack traffic. To address this problem, we introduce the rate of change of asymmetric traffic. For legal burst traffic, the symmetric flows will increase accordingly when the traffic surge in a short time and the rate of asymmetric flows will barely change. While the DDoS attack traffic requires no response from the server, the asymmetric traffic rate will increase significantly when an attack occurs.

Finally, in order to crash the network quickly, an attacker needs to send a large number of packets in a short time. To achieve this goal, the number of packets in the attack flows tend to be small. However, normal flows need to contain a lot of packets to transmit valid information. If the number of flows with a small number of packets is too large, it is very likely that there is a DDoS attack flow in the traffic. Therefore, the percentage of flows with a small number of packets can effectively reflect the network traffic status. At the same time, this feature can also reflect the strength of DDoS attacks, which can be followed by different levels of defense.

Based on the analysis presented above, the traffic features we select for DDoS attack detection are as follows:

1) Average bytes per unit time at time Tn: the average number of bytes of flows per unit time.

$$br = \frac{b_{Tn} - b_{T_{n-1}}}{T_n - T_{n-1}} \tag{1}$$

2) Average durations per flow: the average duration of each flow.

$$adf = \frac{\sum_{i=0}^{flow\_nums_T} dur_i}{flow\_nums_T} \tag{2}$$

3) Percentage of pair-flows: The percentage of the symmetric flow to the total flows per unit time.

$$ppf = \frac{\sum_{i=0}^{flow\_nums_T} N_{pair\_flows}}{\sum_{i=0}^{flow\_nums_T} N_{pair\_flows} + \sum_{i=0}^{flow\_nums_T} N_{single\_flows}} \tag{3}$$

4) Variation rate of single flows: Change of the rate of asymmetric flow per unit time.

$$vrsf = \frac{flow\_nums_T - \sum_{i=0}^{flow\_nums_T} N_{\text{single\_flows}}}{T_n - T_{n-1}} \tag{4}$$

5) Percentage of flows with a few packets: The percentage of flows with a small number of packets in all flows.

$$pfsp = \frac{\sum F_i \left(N_p < V\right)}{flow\_nums_T} \tag{5}$$

In practice, we use in-band telemetry (INT) technology in SDN to obtain information about packets in the network. When packets pass through a programmable switch, the programmable switch will write the predefined information, which includes switch ID, queuing delay, number and timestamp of egress port, length of the packet and five-tuple of {source IP address, destination IP address, source port number, destination port number, protocol type}, etc., into the header of these packets [42]. We use this information and the formulas presented above to calculate the desired metrics.

After the switches detected the abnormal traffic, the controller detects the abnormal traffic passing through the switches. The controller will send standard OFPT_FLOW _MOD messages to the switches, which can ask the switch to buffer the packet from the abnormal flow and send it to the controller for analysis. After extracting the basic information of the flow, the controller uses the above formula to calculate the five feature values of the flow and store it in the database.

C. DDoS DETECTION ALGORITHM

For DDoS attack detection, how to balance the accuracy and the efficiency is an important issue to be considered. Although the detection algorithms based on machine learning

achieve high detection accuracy, these methods are more complicated and requires a longer detection time at the same time. To solve this problem, we use a combined machine learning algorithm based on K-Means and KNN to detect DDoS attacks.

The DDoS detection algorithm consists of K-Means-based training data processing module and K-nearest neighbor (KNN)-based traffic detection module. The training data processing module uses K-Means to cluster the data so that we can compare the distance between the detection point and various clustering centers when using the KNN algorithm for detection. If the k nearest points are all classified to normal group or attack group, we can directly determine that the flow is normal flow or attack flow. If the nearest k clusters cannot determine the type of the point to be measured, then calculating the distance between the point and other points in the training data and select the nearest k points from it. In the traditional KNN algorithm, it is necessary to calculate the distance between the point to be measured and all the other training data, and select the k nearest points to determine the type of the point to be measured. Therefore, in the algorithm we proposed, the results of training data processing module can reduce the calculation cost of detection and reduce the impact of special data on the results of detection algorithm. Moreover, the K-Means algorithm is only used in the training phase and does not affect the time of detection. Therefore, the algorithm we proposed in this paper can improve the detection efficiency while ensuring the detection accuracy.

The purpose of using K-Means algorithm is to divide traffic data with similar features into multiple categories, which is convenient for subsequent KNN algorithms to achieve rapid matching of traffic features and can reduce the workload of detection module and the impact of special data on detection results. The pseudo code of K-Means algorithm is shown in TABLE 1.

Based on the results of training data processing module, the cost of calculation in detection module is greatly reduced because the range of training data is narrowed, and the efficiency of this module is improved. In the process of KNN detection algorithm, the first step is to normalize the measured data. Then, through the abstract distance between the detected flow and the cluster centers, the k clusters closest to the detected flow are found. If the nearest k clusters are all in normal or abnormal groups, the detected flow can be identified as normal or abnormal respectively. Otherwise, k points closest to the detected flow are found and the detected flow is determined according to the labels of these points. If normal points in the k nearest points are more than abnormal points, the label of the detected flow is normal. In some way, it reduces the cost of calculation. Furthermore, it improves the efficiency of algorithm and avoids the influence of some special points on the detection results.

D. DDoS DEFENSE MECHANISM

When the controller detects the DDoS attack, it needs to mitigate the attack as soon as possible to reduce the impact of the attack and ensure the normal operation of the controller. It is simple and effective to forward, modify or block the attack traffic and these are commonly used methods for mitigating DDoS attacks. However, a large number of malicious flow entries generated by the attack flow will still exist in the switches even after the attack is successfully blocked, and it will affect the forwarding process of normal traffic because the malicious flow entries keep occupying the storage resources of the switches. Therefore, the malicious flow entries must be deleted after blocking the attack traffic.

TABLE 1. K-Means training data processing pseudo code.

Input: Training data set $D_{\text{train}} = \left(X_1^{\text{label}}, X_2^{\text{label}}, \ldots, X_N^{\text{label}}, X_{N+1}^{\text{label}}, X_{N+2}^{\text{label}}, \ldots, X_{N+M}^{\text{label}}\right)$

The number of normal data is N and the number of abnormal data is M. Each data contains 5 dimensional features

$X_i^{label} = \left(x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}\right)$

Output: Cluster class C= (C1, C2, …, Ck), centroid set Cen, radius set

for $\mathrm{i} = 1 : N + M$ **do**
  for j=1:5 **do**
    Normalized $x_{ij}^{'} = \dfrac{x_{ij} - \min\left\{x_j\right\}}{\max\left\{x_j\right\} - \min\left\{x_j\right\}}$
  end for
end for
Randomly select k initial centers $X_{c1}^{\text{label}}, X_{c2}^{\text{label}}, \ldots, X_{ck}^{\text{label}}$
repeat
  for $\mathrm{i} = 1 : N + M - k$ **do**
    for s=1: k **do**
      Calculate the distance between the sample X_i^{label} and the center of each cluster

$$D_{is}\left(X_i^{\text{label}}, X_{cs}^{\text{label}}\right) = \sqrt{\left(X_i^{\text{label}} - X_{cs}^{\text{label}}\right)^2} = \sqrt{\sum_{j=1}^{5}\left(x_{ij} - x_{cj}\right)^2}$$

      According to the distance D_{is}, the sample X_i^{label} is classified into the nearest cluster
    end for
  end for
  for s=1: k **do**
    Calculate new cluster center vector $\left(X_{CS}^{label}\right)'$
    if $\left(X_{cS}^{\text{label}}\right)' = X_{cS}^{\text{label}}$ **then**
      keep the current mean vector unchanged
    else
      update the current cluster center vector X_{CS}^{label} to $\left(X_{CS}^{label}\right)'$
    end if
  end for
until the current cluster center vector is not updated
for s=1: k **do**
  calculate the radius of each cluster $r_S = \dfrac{\sum_{i=1}^{N_S} D_{iS}}{N_S}$
end for

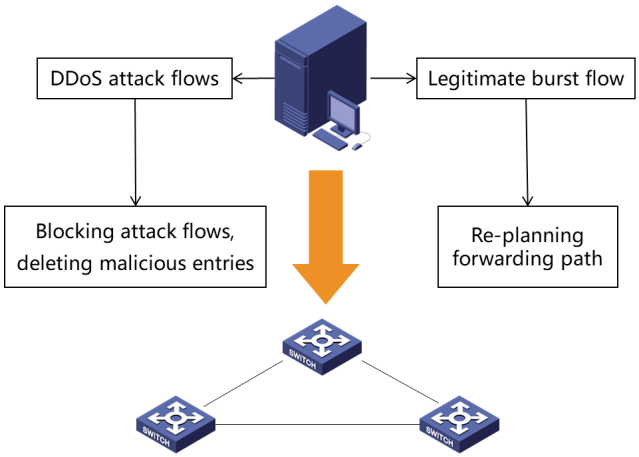


FIGURE 4. The process of DDoS defense.

This vital issue is rarely considered in previous work. In this work, we try to delete the malicious flow entries to release the resource occupied by them.

Furthermore, the network will not only encounter security threats from DDoS attacks, but will also be affected by legitimate burst traffic, such as elephant flows. This flow has many similarities with DDoS attack flows. If the defense methods for DDoS attacks are adopted to the legitimate burst traffic, legitimate users cannot access the network normally. On the other hand, if this kind of network flows are left unchecked, the resources of the network devices or the bandwidth of links will be occupied for a long time, which will also affect the performance of the network. Therefore, we need to take different measures to deal with DDoS attack flows and legal bursts flows. In this work, we prevent DDoS attack flows from entering the network, and redirect the large flows in the network as shown in Fig.4.

When the controller detects the DDoS attack flow, it records the entry port of the attack packets, and then sends the "OFPFC_ADD" message to the switch that found the abnormal flow, and inserts a new flow entry in the flow table of the switch to drop the attack packets. Specifically, in these flow entries, the *Ingress Port* of the *Header* fields is set to the same of the attack packets, the *IP Dst Address* is set to the target address of the attack packets, and the *Actions* field is set to *Drop*. At the same time, the priority of the flow entry needs to be higher than any other flow entries in the switch, so that the switch can match this flow entry first when it receives a DDoS attack packet. Furthermore, it is also necessary to delete malicious flow entries generated by DDoS attacks to release the storage they occupy. We make the controller send the OFPFC_DELETE message to the switch through which the DDoS attack flow passes, and the switch that receives the message will delete the corresponding malicious flow entries.

If the detection trigger mechanism detects an abnormal flow on the switch, but the detection algorithm determines the flow as a normal flow, the flow may be a legitimate burst flow, and the controller needs to formulate a new forwarding path for it to reduce the pressure of the network for forwarding it. First, the update information of the flow table entry is generated, and the original information of forwarding path in the flow table is changed. Then, the controller sends the modification information of the flow entry to the switch that detects the abnormal flow, and the switch forwards the packets from another outgoing port according to the new flow entry. In practice, we make the controller send OFPFC_MODIFY information to the switch to directly modify the original flow table entry.

IV. EXPERIMENT AND EVALUATION

A. IMPLEMENTATION OF THE SYSTEM

As shown in Fig.1, the network traffic will first reach the switches on the data plane, so we deploy a DDoS detection trigger mechanism on the data plane to filter the network traffic passing through each switch and to find the abnormal flows that may be a DDoS attack. The detection trigger mechanism uses the CPU resources on the switches to perform simple rate statistics of the packets, and uses the *packet_in* message rate to determine abnormal flows. If suspicious traffic is found in the network, an alarm signal is generated to notify the controller to perform DDoS detection.

After the controller receives the detection notification, it extracts the features of the traffic passing through the alarm switch and uses the DDoS detection algorithm to determine whether the suspicious flow is an attack flow. If the suspicious flow is proved to be an attack flow, the corresponding defense strategy will be activated. And if the flow is determined to be legitimate burst flow, the flow scheduling method will be activated to re-plan the path of this flow to prevent it from congesting the network.

Finally, the DDoS defense module is deployed on the SDN controller. Using the characteristics of the centralized control of the whole network, the controller can issue corresponding instructions, according to the detection results, to all switches in the network to take defense measures against DDoS attacks to protect itself.

We use ONOS as the SDN controller and use Mininet to simulate the underlying network environment. Specifically, we run two virtual machines on one computer. One of the virtual machines is used to configure Mininet simulation environment as the SDN data plane, including the switch, network links and hosts, the other is configured with ONOS controller environment as SDN control plane. The ONOS controller centrally controls all switches, collecting the information of all devices and links on the data plane through the southbound interface protocol, and obtaining a global view of the network. The topology of the Mininet simulation network is shown in Fig.5.

In the experimental environment, the packet sending and receiving tools are used to simulate the end-to-end traffic. We use Scapy to construct the packets, the random function in Scapy is used to generate random destination IP addresses for the packets to simulate the DDoS attacks on SDN control plane. In addition, in order to detect whether the algorithm can distinguish the legitimate burst flows from attack flows, we use Scapy to send a large number of legitimate packets

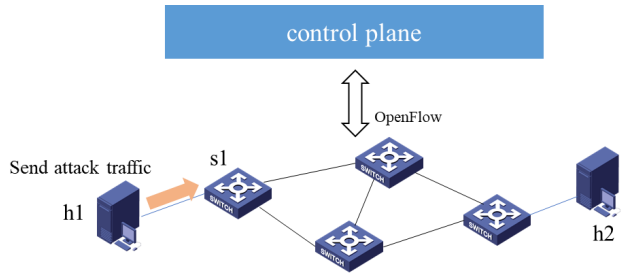


FIGURE 5. Simulation network topology.

at the same time to simulate legitimate burst traffic. With the DDoS attack dataset and the data obtained in the simulation, we construct experiments to verify the feasibility of the algorithm and the effectiveness of the detection and defense mechanisms.

B. EVALUATION WITH BENCHMARK DATASET

We use the commonly used dataset for DDoS attack detection, i.e. NSL-KDD, to evaluate the performance of our method. NSL-KDD dataset is an improvement of the KDD 99 dataset. And it contains 41 characteristics of the network flow such as *duration*, *protocol_type*, *src_bytes*, *dst_bytes*, etc.

Firstly, we divide the data set into three parts, which are used for training, validating, and testing, respectively. Then, we train the K-Means and KNN algorithms with training data, and use validating data to prevent overfitting. Finally, we test the K-Means algorithm, KNN algorithm, and our algorithm, which is a combine of K-Means algorithm and KNN algorithm. Also, we compare the results of these methods with the results of DPTCM-KNN [21] algorithm and KD Tree [23] DDoS detection algorithm. We evaluate the results according to the following indicators:

1) Precision, $\frac{TP}{TP+FN}$. It indicates how many of the samples predicted to be normal are truly normal samples. The higher the detection rate, the higher the accuracy of identifying DDoS attacks.

2) Recall, $\frac{TP}{TP+FP}$. It indicates how much of the normal data in the sample is predicted correctly.

3) False Alarm Rate, $\frac{FP}{FP+TN}$. It is the ratio of the number of normal flows identified as attack flows to the number of all normal flows. The lower the false alarm rate, the better the classification effect.

Among these formulas, TP (True Positive) refers to the probability that attack traffic is recognized as attack traffic, FP (False Positive) refers to the probability that normal traffic is recognized as attack traffic, and TN (True Negative) refers to the probability that normal traffic is recognized as normal traffic. FN (False Negative) refers to the probability that attack traffic is recognized as normal traffic. The higher the accuracy and recall rate, and the lower the false alarm rate, the better the performance of the detection algorithm.

The experimental results are shown in Fig.6. In terms of precision, the average precision of the KNN algorithm and the K-Means algorithm are 95.83% and 95.99%, respectively, while the average precision of our method is 99.03%, which is significantly higher than the KNN algorithm and

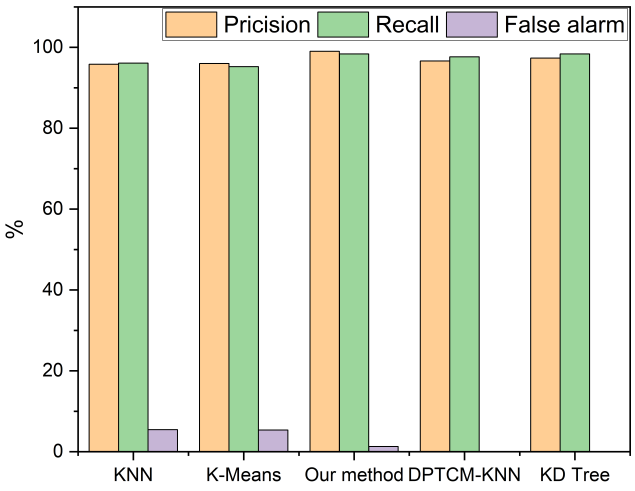


FIGURE 6. Comparison of different methods in terms of precision, recall and false alarm.

the K-Means algorithm. In addition, the average precision of DPTCM-KNN algorithm and KD Tree algorithm is 96.61% [21] and 97.35% [23], respectively. As for recall, the average recall rate of KNN and K-Means algorithm are 96.11% and 95.23%, respectively, while the recall rate of our method reaches 98.35%. In addition, the average recall rate of DPTCM-KNN algorithm and KD Tree algorithm is 97.64% [21] and 98.38% [23], respectively. As far as the false alarm rate is concerned, the average false alarm rate of our method is 1.27%, while the average false alarm rate of KNN and K-Means algorithms are 5.45% and 5.36%, respectively, which is significantly higher. The existence of legal bursty traffic is often ignored by previous DDoS attack detection methods. However, we use the asymmetry characteristics of traffic to detect DDoS attacks, which can effectively distinguish legal bursty flow from the attack flow, so the false positive rate of our method is lower than other methods. It can be seen that the performance of DDoS detection method used in this paper is better than other algorithms in SDN environment.

C. EVALUATION WITH SIMULATED DATA

In this section, we analyze the performance of the DDoS attack detection method in SDN simulation network and evaluates the effectiveness and feasibility of the method in terms of accuracy, recall and misjudgment rate. In addition, we select the detection method based on joint entropy JESS [4] and SOM-based machine learning detection algorithm [8] as comparison.

Most DDoS attack detection algorithms over SDN can be classified as methods based on entropy and methods based on machine learning. The comparison method JESS [4] selected in this work is a DDoS detection method based on joint entropy. This method not only considers the entropy of the target IP address, but also pays attention to the combination of IP address and TCP attributes, and selects the appropriate dynamic attributes for the current attack during the detection process. Although the detection method based on entropy

TABLE 2. Test results of different methods.

Detection method	Accuracy	Recall	False positive
Our method	98.85%	98.47%	0.97%
Entropy method [4]	93.79%	92.80%	6.95%
Distributed-SOM [8]	98.47%	97.79%	1.75%

value will not bring a heavy detection burden to the system, it is difficult to distinguish DDoS attack flows from burst flows because the entropy value only represents the randomness of the data, and the detection accuracy of it is poor than that of methods based on machine learning. The other comparison method, Distributed-SOM [8], is a DDoS detection method based on machine learning. It uses distributed SOM to deal with DDoS flood attacks. This method can effectively improve the detection accuracy and the speed of the system to detect the traffic, while bringing less overhead to the network system.

TABLE 2 is the average of three evaluation indicators detected by the three methods. It can be seen from the table that the accuracy and recall rate of the entropy detection method is only 93.79% and 92.80%, which is lower than the detection method based on the machine learning, so the detection method based on the machine learning can identify DDoS attack flows more accurately. Moreover, the misjudgment rate of the detection method based on entropy value is 6.95%, which is much higher than the 1.75% of the Distributed-SOM method and 0.97% of the detection method we propose. This is because the detection method based on entropy value only depends on the randomness of the data to detect DDoS attacks, and it is easy to determine the sudden change of traffic caused by the surge of network services as the DDoS attack flow, so the accuracy of detection methods based on machine learning is often better than the detection method relying on entropy value.

It can be seen from TABLE 2 that the accuracy and recall rate of the detection method we propose is better than other machine learning based methods. This is because we use a detection trigger mechanism, which can not only improve the detection efficiency, but also simply filtered the network traffic in advance on the data plane so that the controller can only analyze suspicious traffic, and the results of the method we propose are improved compared to other methods. In addition, because the asymmetric feature of the traffic is selected to distinguish between DDoS attack flows and burst flows, the misjudgment rate of our method is lower than other methods and the classification effectiveness of it is better too. The comparative tests discussed above verify the superiority of our method.

In general, we perform binary classification of network flows twice. Firstly, the switches determine whether a flow is a suspicious flow, and then the controller determines whether the suspicious flow is an attack flow or a legitimate burst flow, and take different actions for different results as shown in the Fig.4. It is worth mentioning that our method is general to different types of DDoS attacks, because the five features used to classify network flows mentioned above, i.e. average bytes, average duration, percentage of pair-flows, variation rate of single flows and percentage of flows with a few packets, are all protocol-independent.

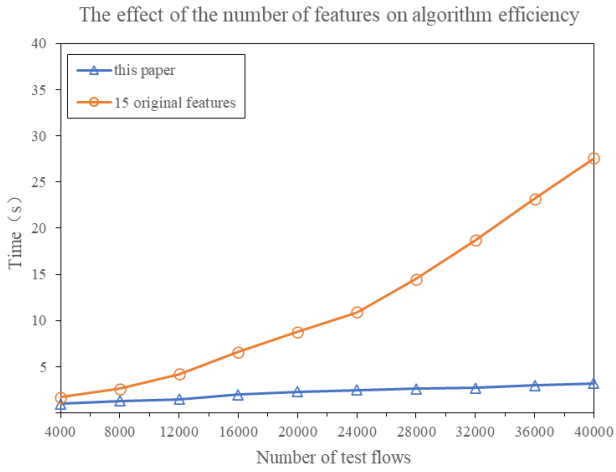


FIGURE 7. Algorithm efficiency with different number of features mechanisms.

In classification algorithms, the number of features greatly affects the efficiency of the detection algorithm. The purpose of selecting more traffic features is to distinguish between normal flows and attack flows more accurately, and to improve the accuracy of the detection algorithm. However, if the dimension of the feature is too high, the delay of detection will be greatly extended, and the more sample data, the longer the detection time is required, therefore, the worse the algorithm efficiency. In this paper, the feature of rate and asymmetry of network traffic are selected to distinguish between normal flows and attack flows. The flow rate and asymmetry have a large difference between normal flows and DDoS attack flows. The original features of the dataset can better distinguish the normal flows and the attack flows, and reduce the detection efficiency while ensuring the accuracy of the algorithm. Thus, the original features in the DDoS detection dataset are used to detect traffic [10]. In order to improve the accuracy of the algorithm, 15 features are used, but at the same time, it also increases the detection delay of the algorithm. The efficiency compared with our method is shown in the Fig.7.

In simulation, the SD-Anti-DDoS method [3] is compared with the method proposed in this paper in terms of the overhead of controller resource. In SDN, the CPU utilization of the controller can be used to indicate the occupancy of the controller's computing resources. As shown in Fig.8., we send DDoS attack flows to the network in the period between 10s and 20s. Before 10s, the DDoS attack flows did not enter the network, and the CPU utilization of the controller in both methods was at a low level. But the CPU utilization of our method is only about 15%, while the CPU utilization of SD-Anti-DDoS method is about 35%.

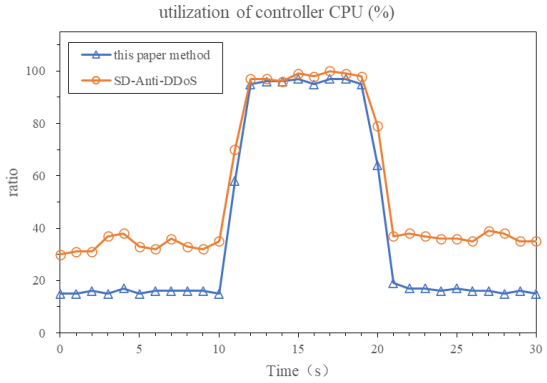


FIGURE 8. Comparison of controller CPU resource consumption.

Between 10s and 20s, the DDoS attack flows enter the network. Both methods find the attack flows and have to process a large amount of data to detect the attack flow, so the CPU utilization of the two methods is maintained at a high level. When the detection is completed, the CPU utilization of our method is restored to the level it used to maintain normal work, while the SD-Anti-DDoS [3] method also requires controller to collect and calculate traffic information on the switches.

Because we deploy the DDoS detection trigger mechanism deployed on the data plane, the detection algorithm on the controller starts to run after the trigger mechanism sent alarm messages, so the detection method proposed in this paper does not occupy the controller's computing resources before DDoS attacks, and resources of the controller are only used to maintain normal work. In the SD-Anti-DDoS method [3] proposed in other literatures, since the controller needs to continuously collect data and determine whether suspicious traffic occurs, it takes up a lot of computing resources of the controller even when the DDoS attack does not occur, so the CPU utilization of the controller is higher than that of our method.

In order to verify the effectiveness and feasibility of the method of defense, we test the defense method in SDN simulation environment. In this paper, the traffic features selected during DDoS detection have a large difference between the normal situation and the situation under DDoS attacks. It can be clearly seen whether there is an attack flow and the intensity of the attack flow in the network, so we verify whether the defense method can achieve the goal of preventing the attack flow by changing several features in this paper.

In the SDN simulation environment we established, the host h1 sends DDoS attack packets to the network at the 10th second. Various indicators have changed significantly when the DDoS attack started, indicating that a DDoS attack flow has appeared in the network. At the same time, the detection trigger mechanism on the data plane finds that there is a suspicious DDoS attack flow on the switch at the beginning of the DDoS attack, that is, at the 10th second, and alerts the controller. After receiving the warning, the controller extracts the traffic features passed through the switch and

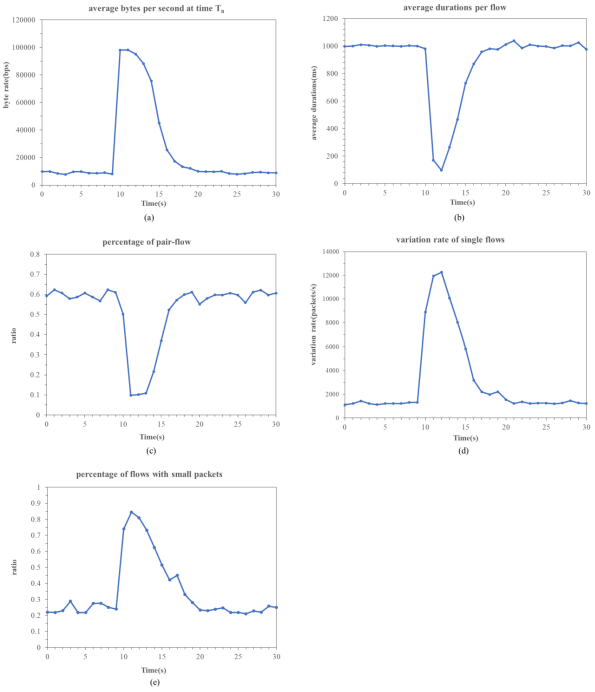


FIGURE 9. Index changes after defense of the controller.

detects it. After detecting whether it is a DDoS attack flow within a few seconds, the controller adopts corresponding attack defense measures and issues corresponding commands to the switches. Therefore, the indicators gradually return to normal level after the DDoS attack occurs as we can see in Fig.9, which proves the effectiveness of the defense method used in this work.

V. CONCLUSION AND FUTURE WORK

Although SDN has many advantages, it also faces the threat of DDoS attacks, the most common security threat in the network. As an advantage of SDN, centralized control also makes the controller in SDN more vulnerable to security threats from DDoS attacks. In response to this problem, in this paper, we analyze the detection and defense mechanism of DDoS attacks over SDN, which combines SDN's own advantages and machine learning algorithms, and adopts a more targeted method to detect and defend against DDoS attacks in the SDN controller. Experiments are constructed to prove that the detection methods proposed in this paper can achieve good results. Moreover, the detection trigger mechanism can effectively detect the occurrence of abnormal flows and save resources of the controller. The adopted defense strategy can also effectively mitigate DDoS attacks.

However, the burden of the controller increases and the efficiency of DDoS detection decreases when the network is under larger-scale network traffic. Therefore, in the future, we will try to exploit the technology of streaming computing to reduce the burden of a single controller to ensure the efficiency of DDoS detection and network quality under large-scale network traffic.

REFERENCES

[1] M. P. Singh and A. Bhandari, "New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges," *Comput. Commun.*, vol. 154, pp. 509–527, Mar. 2020.

[2] Y. Xu and Y. Liu, "DDoS attack detection under SDN context," in *Proc. IEEE INFOCOM-35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[3] Y. Cui, L. Yan, S. Li, H. Xing, W. Pan, J. Zhu, and X. Zheng, "SD-anti-DDoS: Fast and efficient DDoS defense in software-defined networks," *J. Netw. Comput. Appl.*, vol. 68, pp. 65–79, Jun. 2016.

[4] K. Kalkan, L. Altay, G. Gur, and F. Alagoz, "JESS: Joint entropy-based DDoS defense scheme in SDN," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2358–2372, Oct. 2018.

[5] N. A. S. Lima and M. P. Fernandez, "Towards an efficient DDoS detection scheme for software-defined networks," *IEEE Latin Amer. Trans.*, vol. 16, no. 8, pp. 2296–2301, Aug. 2018.

[6] P. Kumar, M. Tripathi, A. Nehra, M. Conti, and C. Lal, "SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1545–1559, Dec. 2018.

[7] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/openflow," in *Proc. IEEE Local Comput. Netw. Conf.*, Denver, CO, USA, Oct. 2010, pp. 10–14.

[8] T. V. Phan, N. K. Bao, and M. Park, "Distributed-SOM: A novel performance bottleneck handler for large-sized software-defined networks under flooding attacks," *J. Netw. Comput. Appl.*, vol. 91, pp. 14–25, Aug. 2017.

[9] X.-D. Zang, J. Gong, and X.-Y. Hu, "An adaptive profile-based approach for detecting anomalous traffic in backbone," *IEEE Access*, vol. 7, pp. 56920–56934, 2019.

[10] Y. Xu, H. Sun, F. Xiang, and Z. Sun, "Efficient DDoS detection based on K-FKNN in software defined networks," *IEEE Access*, vol. 7, pp. 160536–160545, 2019.

[11] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Comput. Netw.*, vol. 62, pp. 122–136, Apr. 2014.

[12] D. Hu, P. Hong, and Y. Chen, "FADM: DDoS flooding attack detection and mitigation system in software-defined networking," in *Proc. GLOBECOM-IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–7.

[13] S. H. Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2012, pp. 19–24.

[14] Y. Wang, T. Hu, G. Tang, J. Xie, and J. Lu, "SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking," *IEEE Access*, vol. 7, pp. 34699–34710, 2019.

[15] T. Bienkowski. (2018). *No Sooner Did The Ink Dry: 1.7Tbps Ddos Attack Makes History [EB/OL]*. [Online]. Available: https://www.netscout.com/blog/security-17tbps-ddos-attack-makes-history

[16] (2017). *Arbor's Worldwide Security Report [EB/OL]*. [Online]. Available: https://pages.arbornetworks.com/rs/082-KNA-087/images/13th_Worldwide_Infrastructure_Security_Report.pdf

[17] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 77–81.

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

[19] C. Gong, D. Yu, L. Zhao, X. Li, and X. Li, "An intelligent trust model for hybrid DDoS detection in software defined networks," *Concurrency Comput., Pract. Exp.*, vol. 32, no. 16, Aug. 2020.

[20] Y. Yu, L. Guo, Y. Liu, J. Zheng, and Y. Zong, "An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks," *IEEE Access*, vol. 6, pp. 44570–44579, 2018.

[21] H. Peng, Z. Sun, X. Zhao, S. Tan, and Z. Sun, "A detection method for anomaly flow in software defined network," *IEEE Access*, vol. 6, pp. 27809–27817, 2018.

[22] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Secur. Commun. Netw.*, vol. 2018, pp. 1–8, Apr. 2018.

[23] L. Zhu, X. Tang, M. Shen, X. Du, and M. Guizani, "Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 628–643, Mar. 2018.

[24] P. Bosshart, D. Dally, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, 2014.

[25] *The P4 Language Specification, Version 1.1*, P4 Lang. Consortium, 2016.

[26] V. Akilandeswari and S. M. Shalinie, "Probabilistic neural network based attack traffic classification," in *Proc. 4th Int. Conf. Adv. Comput. (ICoAC)*, Dec. 2012, pp. 1–8.

[27] P. Kathiravelu and L. Veiga, "SD-CPS: Taming the challenges of cyber-physical systems with a software-defined approach," in *Proc. 4th Int. Conf. Softw. Defined Syst. (SDS)*, May 2017, pp. 6–13.

[28] P. Manso, J. Moura, and C. Serrão, "SDN-based intrusion detection system for early detection and mitigation of DDoS attacks," *Information*, vol. 10, no. 3, p. 106, Mar. 2019.

[29] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Y. Yau, and J. Wu, "Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1838–1853, Jul. 2018.

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

[31] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[32] K. Kalkan, G. Gur, and F. Alagoz, "Defense mechanisms against DDoS attacks in SDN environment," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 175–179, Sep. 2017.

[33] K.-Y. Chen, A. R. Junuthula, I. K. Siddhrau, Y. Xu, and H. J. Chao, "SDNShield: Towards more comprehensive defense against DDoS attacks on SDN control plane," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2016, pp. 28–36.

[34] B. Yuan, D. Zou, S. Yu, H. Jin, W. Qiang, and J. Shen, "Defending against flow table overloading attack in software-defined networks," *IEEE Trans. Services Comput.*, vol. 12, no. 2, pp. 231–246, Mar. 2019.

[35] A. Bhandari, A. L. Sangal, and K. Kumar, "Characterizing flash events and distributed denial-of-service attacks: An empirical investigation," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 2222–2239, 2016.

[36] G. Kirubavathi and R. Anitha, "Structural analysis and detection of Android botnets using machine learning techniques," *Int. J. Inf. Secur.*, vol. 17, no. 2, pp. 153–167, Apr. 2018.

[37] T. M. Nam, P. H. Phong, T. D. Khoa, T. T. Huong, P. N. Nam, N. H. Thanh, L. X. Thang, P. A. Tuan, L. Q. Dung, and V. D. Loi, "Self-organizing map-based approaches in DDoS flooding detection using SDN," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 249–254.

[38] J. Boite, P.-A. Nardin, F. Rebecchi, M. Bouet, and V. Conan, "Statesec: Stateful monitoring for DDoS protection in software defined networks," in *Proc. IEEE Conf. Netw. Softwarization*, Jul. 2017, pp. 1–9.

[39] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for openflow," in *Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw.* Berkeley, CA, USA: USENIX Association, 2010, pp. 1–6.

[40] A. Wang, Y. Guo, F. Hao, T. V. Lakshman, and S. Chen, "Scotch: Elastically scaling up SDN control-plane using vSwitch based overlay," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2014, pp. 403–414.

[41] B. V. Karan, D. G. Narayan, and P. S. Hiremath, "Detection of DDoS attacks in software defined networks," in *Proc. 3rd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solutions (CSITSS)*, Bengaluru, India, Dec. 2018, pp. 265–270, doi: 10.1109/CSITSS.2018.8768551.

[42] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Secur. Commun. Netw.*, vol. 2018, pp. 1–8, Apr. 2018.

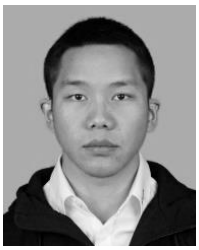
LIANG TAN was born in Huangshi, Hubei, in 1979. He received the master's degree in computer technology from the School of Computer Science and Technology, Huazhong University of Science and Technology. He is currently a Senior Engineer with the China Ship Development and Design Center (CSDDC). He involved in the design and development of ship information systems. He has participated in international seabed regional research and development projects, China Ocean Association Scientific Research Ship Network Information Integration System Research and Development projects, and China Ministry of Industry and Information Technology high-tech ship research projects.

YUE PAN received the bachelor's degree in communication engineering and the master's degree of information and communication engineering from Wuhan University, in 2017 and 2020, respectively. Her research interests include software-defined networks, system security, and big data.

JING WU (Member, IEEE) received the B.Eng. degree in communication engineering and the Ph.D. degree in communication and information systems from Wuhan University, China, in 2002 and 2007, respectively. From 2004 to 2005, she undertook her Postdoctoral Research work at LIMOS, Clermont-Ferrand, France. She is currently an Associate Professor with Wuhan University. Her research interests include wireless communication networks, network simulation, and intelligence data processing.

JIANGUO ZHOU was born in 1965. He received the B.Eng. degree in computer and automation from Sichuan University, China, in 1988, and the M.Eng. degree in radio electronics and the Ph.D. degree in communication and information systems from Wuhan University, China, in 1991 and 2013, respectively. He is currently an Associate Professor with Wuhan University. He has authored over 50 articles in different journals and conferences. His research interests include computer networks and the spatial information networks. He also serves as a Senior Member of the China Institute of Communications.

HAO JIANG (Member, IEEE) received the B.Eng. degree in communication engineering and the M.Eng. and Ph.D. degrees in communication and information systems from Wuhan University, China, in 1999, 2001, and 2004, respectively. From 2004 to 2005, he undertook his Postdoctoral Research work at LIMOS, Clermont-Ferrand, France. He was a Visiting Professor with the University of Calgary, Canada, ISIMA, and Blaise Pascal University, France. He is currently a Professor with Wuhan University. He has authored over 60 articles in different journals and conferences. His research interests include mobile ad hoc networks and mobile.



YUCHUAN DENG received the bachelor's degree in computer science and technology from Sichuan University, in 2019. He is currently pursuing the master's degree in information and communication engineering with Wuhan University. His research interests include software-defined networks, system security, and machine learning.