

Mitigating while Accessing: A Lightweight Defense Framework Against Link Flooding Attacks in SDN

Sun Hancun^{1,2}, Chen Xu^{1,3}, Luo Yantian^{1,2}, Ge Ning^{1,2,*}

¹ Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

² Beijing National Research Center for Information Science and Technology, Beijing 100084, China

³ Naval Research Institute, Beijing 100036, China

* The corresponding author, email: gening@tsinghua.edu.cn

Cite as: H. Sun, X. Chen, *et al.*, “Mitigating while accessing: A lightweight defense framework against link flooding attacks in sdn,” *China Communications*, 2024, vol. 21, no. 11, pp. 15-27. **DOI:** 10.23919/JCC.fa.2024-0209.202411

Abstract: Link flooding attack (LFA) is a type of covert distributed denial of service (DDoS) attack. The attack mechanism of LFAs is to flood critical links within the network to cut off the target area from the Internet. Recently, the proliferation of Internet of Things (IoT) has increased the quantity of vulnerable devices connected to the network and has intensified the threat of LFAs. In LFAs, attackers typically utilize low-speed flows that do not reach the victims, making the attack difficult to detect. Traditional LFA defense methods mainly reroute the attack traffic around the congested link, which encounters high complexity and high computational overhead due to the aggregation of massive attack traffic. To address these challenges, we present an LFA defense framework which can mitigate the attack flows at the border switches when they are small in scale. This framework is lightweight and can be deployed at border switches of the network in a distributed manner, which ensures the scalability of our defense system. The performance of our framework is assessed in an experimental environment. The simulation results indicate that our method is effective in detecting and mitigating LFAs with low time complexity.

Keywords: attack mitigation; distributed denial of ser-

vice (DDoS); link flooding attack (LFA); software defined networking (SDN)

I. INTRODUCTION

The global rollout of the fifth-generation (5G) communication technologies has facilitated the prosperity of Internet of Things (IoTs) [1–8]. With limited resources and regulation, IoT devices are vulnerable to exploitation by attackers, which increases the threat of distributed denial of service (DDoS) attacks. Attack traffic in traditional DDoS attacks can be distinguished with high accuracy using machine learning-based methods considering the distinct behavioral characteristics between legitimate and malicious traffic. However, some more covert attack patterns have emerged, and the link flooding attacks (LFAs) [9] have posed great challenges for network security.

Defending against LFAs is difficult due to the following characteristics. First, attackers coordinate a large number of bots to generate low-speed malicious flows in LFAs. The attack traffic is difficult to distinguish from legitimate traffic based on the behavioral patterns of a single flow. Second, the attack traffic is directed towards decoy servers surrounding the target area. As a result, victims in target area do not receive malicious traffic, making it difficult to detect an LFA. Third, in typical LFAs, attackers consecutively

Received: Apr. 30, 2024

Revised: Aug. 27, 2024

Editor: Li Changle

change the attack strategy by changing target links within seconds. Therefore, the defense methods need to be lightweight to address the attacks in real time.

Software defined networking (SDN) architecture opens up a new direction in defending against LFAs [10–13]. The SDN controller can request and analyze traffic statistics in real-time, enabling a quick response to the presence of LFAs. In addition, possessing a global perspective of network, the controller can dynamically develop defense strategies by distributing flow entries to switches in the data plane. However, most existing methods encounters two challenges. First, the limitation of communication and computation resources of SDN controller restricts the complexity of LFA defense methods [14–16]. The proposed method is required to be lightweight to ensure the scalability. Second, most methods reroute the redundant traffic on upstream nodes close to congested links to mitigate the congestion. Limited by local topology and residual bandwidth resources, these methods have limited mitigation capabilities, and new congestion can be easily caused on downstream untar-geted links.

To overcome these challenges, we have proposed a lightweight framework for detecting and mitigating LFAs at the border switches in SDN. In the detection stage, the controller periodically collects traffic data from the data plane and updates the traffic distribution in the network. We employ an entropy-based detection method to ensure the low complexity. Subsequently, we mitigate LFAs by blocking the suspicious flows and rerouting flows passing through target links. Both mitigation strategies can be deployed at border switches to alleviate the computational and communication overhead of the SDN controller. To the best of our knowledge, this is the first work in LFA defense that considers evaluating traffic and deploying defense strategies in a distributed manner at border switches.

Our main contribution is summarized as follows.

1. We propose a novel framework for LFA detection and mitigation at the border switches in SDN. This framework blocks attack flows deploys rerouting schemes at border switches of SDN, which can not only prevent malicious traffic from entering the downstream links and consuming bandwidth resources, but also avoid subsequent secondary congestion by unified reroute path arrangement.
2. We propose a lightweight entropy-based method to detect LFAs and a trust evaluation method to evaluate the behavior of flows. By requesting and analyzing traffic statistics in real-time, the centralized SDN controller can respond quickly when LFA exists and update the trust value of flows to determine the corresponding defense strategy. Our defense framework can work well with a low communication and computation overhead, and avoid the introduction of specialized hardware or software, which can be deployed in off-shelf switch products.
3. We verify the performance of the proposed framework in a testbed and evaluate the defense effect and time cost of our system. Simulation results show that our method can quickly detect LFA and provide effective mitigation strategies. Specifically, our method significantly reduces the time costs required to determine the rerouting strategy by 79.8%, while decreasing the average transmission delay for legitimate flows, compared to the existing method.

Part of this work was presented in IEEE ICCT 2023 [13]. Compared with our previous work, we introduce trust evaluation to our framework and our system can distinguish malicious traffic and reroute the redundant flows based on trust value of flows. The rest parts of this paper are organized as follows. In Section II, we introduce the related works. Section III illustrates our proposed LFA defense system model in detail. We display the experiments and discuss the results in Section IV. Finally, we conclude our work in Section V.

II. RELATED WORKS

Recently, researchers have pursued a variety of approaches to defend against LFAs. According to the purposes of these methods, we categorize them as LFA prevention methods, LFA detection methods and LFA mitigation methods. In an LFA defense system, the three methods can complement each other.

2.1 LFA Prevention Methods

Several researches have been developed to deal with LFAs proactively and prevent the risk of LFAs [17–23]. Hirayama *et al.* [17] monitored the number of traceroute packet within the network to identify the

potential risk of LFAs. This method has been proved to be effective, but can cause frequent false alarm cases. The authors of [18] and [19] designed routing systems to hide the bottleneck of the network by concealing invisible links and presenting a deceptive linkmap respectively. Venkateswaran *et al.* [20] integrated obfuscation and active moving target defense to implement cyber deception. By elevating the entropy of information accessible through the system's service interfaces, they hinder attackers from acquiring details about internal control structures, thereby reducing the effectiveness of launching LFAs with constrained bot resources. Liaskos *et al.* [22] provided a metric to evaluate the correlation between network topology and LFA detection efficiency, which can conduct the network operator to design the topology of the network.

2.2 LFA Detection Methods

To detect LFAs, the network administrators usually focus on the congested links or monitor malicious traffic in the network. Xue *et al.* [24] developed a novel framework named Linkscope. Linkscope employs non-cooperative measurement techniques to capture the abnormal performance degradation and infer the target links in LFAs and this method can circumvent the mediation problem of autonomous systems (ASes). Several studies have focused on detecting LFAs by taking advantage of SDN. The authors of Woodpecker [25] deployed a heuristic algorithm to select a set of switches and upgrade them into SDN-enabled switches. Woodpecker can locate and determine the causes of congestion by utilizing global congestion information, enabled by the global perspective of SDN. Ravi *et al.* [26] proposed a congestion location algorithm in hybrid SDN environment, which can detect and locate the congested links.

Another type of detection method is to detect malicious traffic in the network, and machine learning (ML) based methods are state-of-the-art [27–34]. The multilayer perception (MLP) [27], convolutional neural network (CNN) [28], long short-term memory (LSTM) module [29, 30], and Transformer models [31, 32] have been employed for malicious traffic detection. However, since malicious traffic in LFAs is difficult to distinguish, LFA detection methods utilize more complicated designs to ensure high detection accuracy. Rasool *et al.* [27] extracted 14 features from

traffic flows and applied an MLP network to classify malicious flows. However, the selected features are based on a single flow and do not consider the cooperation of bots. Rezazad *et al.* [33] and Xie *et al.* [34] utilized random forest (RF) to classify malicious traffic. The method is effective to distinguish attack traffic from background traffic based on enough features but would fail if the attackers select the decoy servers distributedly in the network. Several recent researches have introduced blockchain to LFA defense systems. Cao *et al.* [35] deployed LFA detection module on border routers to record traceroute information, and Jiang *et al.* [36] applied blockchain to record and share the network features in real time. Then both methods utilize machine learning based methods, the recurrent neural network (RNN) and LSTM module respectively, to capture the attack patterns and identify suspicious sources.

2.3 LFA Mitigation Methods

To mitigate LFAs, traffic engineering-based methods have been widely applied [25, 37–39]. In Woodpecker [25], the authors formulate an optimization problem to balance the traffic load on bottleneck links and other links. Wang *et al.* [39] proposed a Deep Q-learning (DQL) based traffic engineering algorithm to obtain rerouting paths and can effectively reduce the memory space and search time. Mitigation methods based on rerouting do not require the identification of malicious flows, but they cannot eliminate the attack traffic in the network. The authors of [40] set a rate limiting window for suspicious flow to prevent attackers from injecting too much traffic. Similarly, Yang *et al.* [41] proposed a mitigation method based on flow rate control to punish aggressive flows by dropping more of their packets. These methods can achieve high drop probabilities for attack flows but can also influence the flows from legitimate devices.

Game theory provides a novel perspective on defending against LFAs. Chen *et al.* [42] modeled the attack mechanisms and defense strategies in LFAs with a Bayesian game and obtained the Bayesian Nash equilibrium. The model is proved effective and enables the defenders to develop precise and accurate defense strategies. Wang *et al.* [43] formulated a game-theoretic interaction between attackers and defenders to infer the target area and identify the optimal routing

scheme to mitigate LFAs. Ma *et al.* [44] characterized the behavior of attackers and allocated the monitoring resource with a Stackelberg game model to enable the network response effectively and accurately.

Several researches consider the mitigation of LFAs in large-scale networks. The frameworks in [45] and [46] are effective in handling LFAs, but they require the global coordination of ASes. Xie *et al.* [47] formulated the bandwidth pricing problem to facilitate the domain-level resource coordination. The method can provide incentives for Internet service providers (ISPs) to cooperate and achieve efficient network resource utilization and LFA resilience. Additionally, in large-scale networks, the limited resource of programmable switches can be the bottleneck of the defense system. Zhou *et al.* [48] developed a distribution protocol to schedule the flow states on different switches to reduce the memory bottleneck. The limitation is it may lose some information once a switch fails.

III. SYSTEM MODEL

In this section, we present our framework for defending against LFAs in SDN. Figure 1 illustrates the architecture of our LFA defense system, comprising a traffic aggregation module, an LFA detection module, a trust evaluation module and an LFA mitigation module. The traffic aggregation module collects the traffic statistics in the data plane periodically and acquires flow features such as the five-tuple and the amount of traffic. In addition, the module aggregates the flows virtually to obtain the amount of traffic on each link within the network. Subsequently, the LFA detection module updates the historical traffic distribution of the network, and utilizes relative entropy analysis to monitor the risk of LFAs. Meanwhile, the trust evaluation module updates the trust value of source and destination addresses based on the communication behaviors and network status. Once the LFA detection module alarms the risk of LFAs, the mitigation module drops the flow entries of the upstream switches of the congested link, develops a defense strategy to reroute or drop traffic to mitigate congestion, and distribute flow rules to the data plane. The traffic aggregation module can be implemented based on OpenFlow protocols, while the detailed implementation of other three modules will be presented in the following subsections.

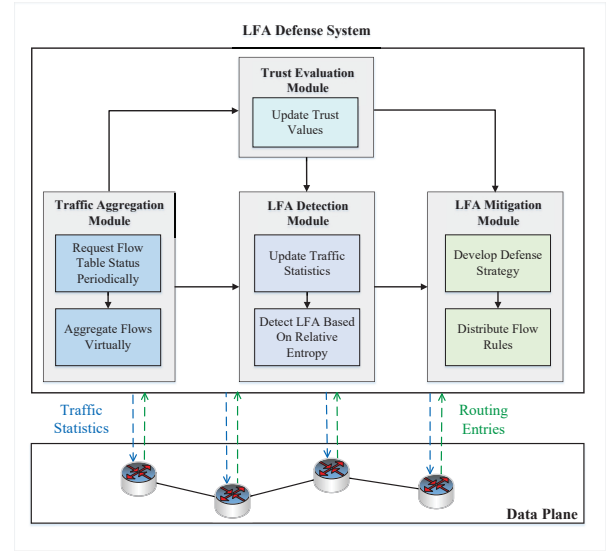


Figure 1. An overview of our system model.

3.1 Relative Entropy-Based LFA Detection

We apply a relative entropy-based method to detect LFAs. In the network, persistent links with large-scale traffic and limited bandwidth can be the bottleneck of the network. Attackers are able to locate the links by probing the network with tracing tools such as traceroute. Let $W = (\mathcal{N}, \mathcal{L})$ denote the whole network with a set of nodes $\mathcal{N}$ and links $\mathcal{L}$. We denote the set of persistent links to the target area as $\mathcal{L}_t = \{l_1, l_2, \dots, l_k\}$, and let λ_i be the scale of traffic on link l_i . To launch an LFA, attackers select a subset of $\mathcal{L}_t$ as target links.

In each time period, the traffic aggregation module updates the current traffic load on each link, and normalizes the distribution of traffic as

$$\theta_i = \frac{\lambda_i}{\sum_{l_i \in \mathcal{L}_t} \lambda_i}. \quad (1)$$

In attack-free scenarios, the controller monitors the network continuously and updates the traffic distribution σ_i for link l_i in the network by averaging the historical statistics. The relative entropy of the traffic distribution is calculated as

$$H_r = - \sum_{l_i \in \mathcal{L}_t} \sigma_i \log \frac{\theta_i}{\sigma_i}. \quad (2)$$

To determine the detection threshold, we employ relative entropy analysis to the network previously and record the distribution of the entropy value. The

threshold t is set to ensure the system has an acceptable false alarm rate, e.g. 1%. We can formulate our decision of LFA detection as

$$H_r \leq t \Rightarrow \text{attack-free}; \quad (3a)$$

$$H_r > t \Rightarrow \text{LFA exists}. \quad (3b)$$

Once the LFA detection module alarms an LFA risk, it targets the links with high link utilization rates to locate the congestion. Meanwhile, it invokes the LFA mitigation module to formulate LFA defense strategies.

3.2 Trust Evaluation Based on Communication Behaviors

In LFAs, the malicious traffic is of low speed and protocol confirming, which makes it difficult to distinguish it from the legitimate traffic. To identify the malicious and legitimate traffic with high accuracy, we deploy trust evaluation for the source and destination addresses based on four features, including the reuse of IP addresses, traffic symmetry, the entropy of flow destinations, and the priority of network service. The explanations are listed in detail as follows.

- (1) **Reuse of IP addresses.** This feature indicates the frequency of the IP addresses involved in LFAs. The controller updates this feature when LFA detection module alarms the risks of LFAs in the network. Source addresses sending flows and destination addresses receiving flows through the congested links in the current time slot are recorded, regardless of whether the traffic is malicious or legitimate.
- (2) **Traffic symmetry.** This feature refers to the proportion of inbound and outbound traffic volume of the addresses. In each time slot, the controller updates this feature by averaging the total number of sent and received packets in each flow from a particular address, which can be obtained from the traffic aggregation module.
- (3) **Entropy of flow distributions.** This feature refers to the entropy of the distribution of flows generated or received from the evaluated IP address. It is also updated based on the traffic aggregation module in each time slot and reflects the

communication preference of the node.

- (4) **Priority of network service.** This feature refers to the importance and priority of communication content. It can be obtained by sampling the flow packets with deep packet inspection (DPI).

Let $X = (x_1, x_2, x_3, x_4)$ denote the feature vector of the source or destination address, and let $S_n \in [0, 1]$ denote the trust value in the n th time slot. In LFAs, since the malicious traffic is difficult to distinguish based on its communication behavior in a single time slot, we set the first feature, the reuse of IP addresses, as the leading feature in the trust evaluation process. Therefore, we apply different data processing methods to these features.

For the three features x_2, x_3, x_4 , we consider the statistics in the whole network, and normalize the features based on the behavior of all the addresses. Let μ_i and σ_i^2 denote the mean and variance of feature x_i respectively. We normalize the feature as

$$x'_i = \frac{x_i - \mu_i}{\sigma_i}, \quad i = 2, 3, 4. \quad (4)$$

We use the weighted average of three features to reflect the behavior of the address in the current time slot, which can be expressed as

$$x' = \sum_{i=2}^4 a_i x_i. \quad (5)$$

For feature x_1 , we do not normalize it based on the behavior of other addresses. Instead, we set a piecewise function $f(x)$ to denote the penalty of the trust value if an address is involved in an LFA. The function is expressed as

$$f(x_1) = \begin{cases} 0 & 0 \leq x_1 < m \\ ae^{x_1} + b & m \leq x_1 < n \\ 1 & x_1 \geq n \end{cases} \quad (6)$$

The parameter m is set to decrease the false positive rate since legitimate traffic can also pass through the target links, and the trust evaluation module can tolerate the first m times when the address may be involved in the LFAs. With the increasing frequency of IP addresses involved in LFAs, the penalty function increases when evaluating the suspicious addresses. The

selection of parameters m and n is analyzed in Section IV. To ensure the continuity of the piecewise function, the parameters can be calculated as

$$a = \frac{1}{e^n - e^m}, b = -\frac{e^m}{e^n - e^m}. \quad (7)$$

Based on the four features, the trust evaluation module updates the trust value in each time slot. In the n th time slot, if the LFA detection module alarms the risk and the IP address is involved in the attack, the trust value is updated as

$$S_n = S_{n-1} - f(x_1)\sigma(x'), \quad (8)$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ represents the Sigmoid function. Otherwise, the trust value is updated as

$$S_n = S_{n-1} + c. \quad (9)$$

In attack-free status, the trust value is set to increase linearly, which is much lower than the decreasing speed when LFA risk exists, since attackers can conceal the bots as inactive nodes or legitimate users. When the LFA detection module alarms the risk of an LFA, the LFA mitigation module formulates a strategy to reroute or drop the traffic based on the trust value of the source and destination addresses.

3.3 Optimized LFA Mitigation

Once the system alarms an LFA event, the mitigation module deploys two strategies: blocking the suspicious traffic at the border switches of network and rerouting the traffic to mitigate the congestion. The mitigation strategy for a particular flow is based on the trust value of the flow.

Let F_j denote a traffic flow in the network with source address src_j and destination address dst_j . Let S_j and D_j denote the corresponding trust values for src_j and dst_j respectively. We set the trust value of F_j as

$$T_j = S_j D_j. \quad (10)$$

Assume that link l_i is the target link of the link flooding attack. We define a mapping and represent it as $I(l_i, src_j, dst_j)$. If the flow from src_j to dst_j passes through link l_i , the value is determined as

Algorithm 1. *Identify the rerouting paths in the network.*

Input: Network $W = (\mathcal{N}, \mathcal{L} \setminus \{l_{A,B}\})$, source A , sink B , λ_i , B_i and α_i of link l_i , $f_{A,B}$

Output: Rerouting path set P , traffic constraint F

```

1:  $P = \emptyset, F = \emptyset, f = 0$ 
2: while  $f < f_{A,B}$  do
3:   find an augment path  $P_j$  using breadth-first search (BFS)
4:    $f_j = \min_i (B_i \alpha_i - \lambda_i)$ , for  $l_i$  in path  $P_j$ 
5:    $P = P \cup \{P_j\}, F = F \cup \{f_j\}$ 
6:    $f = f + f_j$ 
7: end while
8: return  $P, F$ 
```

$I(l_i, src_j, dst_j) = 1$. Otherwise, $I(l_i, src_j, dst_j) = 0$.

To deal with the suspicious flows, we first set a threshold θ for blocking suspicious flows with low trust values. If a switch detects $I(l_i, src_j, dst_j) = 1$ and $T_j < \theta$, it drops the flow F_j at the border switches of the network, which can alleviate the link utilization in the network and the method is more lightweight to enable the network to react more quickly when LFA risks exist.

To derive the rerouting scheme of other flows with higher trust values, we consider the range of switches to distribute rerouting flow entries and measure our routing scheme with the average time delay of legitimate flows. First, to identify the available rerouting paths in the network, we deployed an algorithm based on the Edmonds-Karp algorithm [49]. Let B_i be the bandwidth of link l_i and α_i be the maximum link utilization rate of link l_i to avoid congestion. Let link $l_{A,B}$ be the target link and the forwarding direction of the attack traffic is from switch A to switch B , and to mitigate the congestion, the amount of traffic to reroute from $l_{A,B}$ is $f_{A,B}$. The method is shown as Algorithm 1. Notice that the time complexity of the Edmonds-Karp Algorithm is $\mathcal{O}(mn^2)$, where m and n denote the number of nodes and links in the network. In LFA scenarios, the bottleneck of the network is among the target link set $\mathcal{L}_t$, therefore the complexity of our method is $\mathcal{O}(mn^2)$, where n denotes the size of target link set, which is much lower than the number of links in the network. Additionally, if there are multiple congested links in the network, we can apply the algorithm repeatedly for each link.

After deploying the previous algorithm, we obtain a set $P = \{P_1, P_2, \dots, P_k\}$ for available rerouting paths and each path P_i begins at A and terminates at B . For flow F_j passing through link $l_{A,B}$, we denote the current routing path as P_0 . Let a vector h_j indicate the rerouting scheme for F_j . If $h_j = 0_{1 \times n}$, we maintain the current routing path P_0 . If the i th element of h_j is set as 1, we apply path P_i to reroute the traffic and the alternative path can be expressed as $(P_0 \cup P_i) \setminus (P_0 \cap P_i) \setminus \{l_{A,B}\}$. Let a vector $C_{1 \times n}$ with elements C_i indicate the time delay for routing the flow from path P_i in attack-free scenarios. Noting that the attackers may dynamically change their target links, we attempt to ensure the stability of legitimate flows in the rerouting stages. We take into account the trust value of each flow and legitimate flows with high trust values are more likely to retain their optimal routing paths. We formulate an optimization problem to minimize the weighted time delay as

$$\begin{aligned} \min \quad & \sum_{F_j} I(l_i, src_j, dst_j) T_j C h_j^T \\ s. \quad & t. \quad \sum_{F_j} \lambda_{F_j} h_j^T \leq F \\ & \lambda_i / B_i \leq \alpha_i, \quad \forall l_i \in \mathcal{L}_t \end{aligned} \quad (11)$$

Notice that to obtain the optimal solution of the centralized optimization problem, the controller needs to determine the rerouting scheme for a large number of flows, which increases the complexity and solution time of the problem. Therefore, we attempt to apply a distributed rerouting scheme in which the controller first identifies the proportion of traffic to be rerouted in each available path by normalizing the output F in Algorithm 1. Then, the border switches formulate optimization problems separately similar to problem 11 and the constraints of the optimization problem change to $\sum_{F_j} \lambda_{F_j} h_j^T \leq pF$ where p denotes the proportion of traffic entering the network from the particular border switch.

By deploying the distributed optimization problem, the scale of the problem can be significantly reduced. The LFA mitigation module then solves the problems and distributes flow rules to the data plane. We will present the evaluation and comparison of the mitigation methods in Section IV.

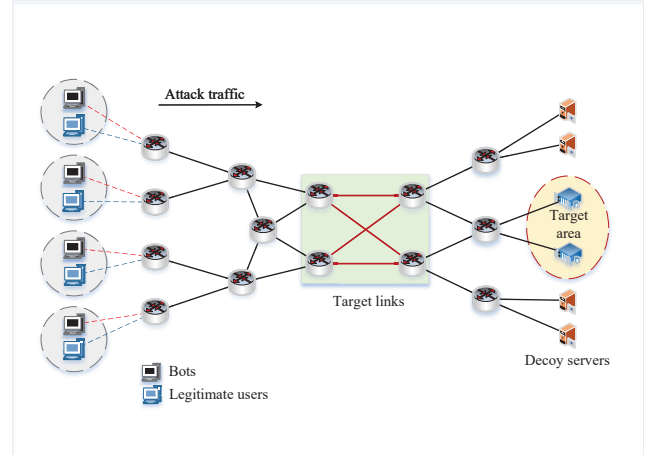


Figure 2. The experimental network topology.

IV. PERFORMANCE EVALUATION

4.1 Experimental Settings

We carry out a series of experiments in a testbed to evaluate the performance of our proposed LFA defense system. Figure 2 shows the network topology used in our experiments to simulate LFAs, which consists of 14 OVSes and 18 links and each link has a bandwidth of 100 Mbps. Four links in the network are selected as available target links for LFA attackers and are marked in red in the figure. The number of legitimate users and the maximum number of bots are both set as 100, and they are distributed across four areas connected by four border switches in the network. The number of destination addresses in the target area and the decoy servers are set as 50.

To simulate legitimate traffic, we apply the behaviors of the benign traffic in the BoT-IoT dataset [50], which is created by the Cyber Range Lab of the Australian Center for Cyber Security (ACCS) using the tshark tool. For each flow generated from the legitimate users, the probability of the destination address is in the target area is set as 75%, and the traffic has no preference for the routing path. To simulate malicious flows, since there are no public datasets for LFAs so far [51], we devise the traffic flow based on the attack mechanism in [9]. A link flooding attack usually involves several rounds, and the duration of each round is no longer than 180 seconds. In our experiments, the attacker randomly selects 1 or 2 links as the target in each round, and the proportion of traffic load on the target link is set to be 75%, 100%, and 125% times of

the link bandwidth. The bots and legitimate users are connected to the border switches of the network and the volume of attack traffic is changed by coordinating different numbers of bots.

Furthermore, in LFAs, the attacker may devise the behavior of bots to evade the supervision of the network [12]. In our experiments, we assume that the attacker has prior knowledge of the network topology and current routing schemes. In each round of attack, the attacker deploys the behavior for each bot, and the available choices include sending traffic to decoy servers through the selected target links, sending traffic to random destination in the network, and remaining inactive. The probability distribution for each option is related to the intensity of the attack. In the following experiments, the probabilities are set as 60%, 20%, and 20% respectively.

4.2 LFA Detection Performance

We employed our relative entropy-based LFA detection method in the testbed. The traffic aggregation module requests the network status with a period of 25 seconds. Figure 3 shows the average relative entropy of the traffic distribution during LFAs in 50 independent experiments. The fluctuation in relative entropy correlates with the severity of LFA. As the maximum link traffic of the targeted links increases, there is a noticeable rise in relative entropy. By setting a suitable detection threshold, our method can achieve a detection accuracy exceeding 95%. However, due to the sensitivity of entropy-based methods to the selection of thresholds, direct comparison of detection accuracy may lack persuasiveness. Hence, the main advantage of our method is more lightweight compared to ML-based detection methods at similar detection rates.

Additionally, the malicious traffic in LFAs may not reach target links simultaneously, resulting in a sustained rise in traffic on target links throughout the attacks. Our module has the capability to alarm potential LFA risks before the volume of traffic reaches its peak, enabling the proactive deployment of mitigation strategies.

4.3 Trust Evaluation Performance

To evaluate the behavior of legitimate users and bots, we set the parameters as $m = 3$, $n = 6$ and $c = 0.1$ in the trust evaluation module. Figure 4 shows the trust

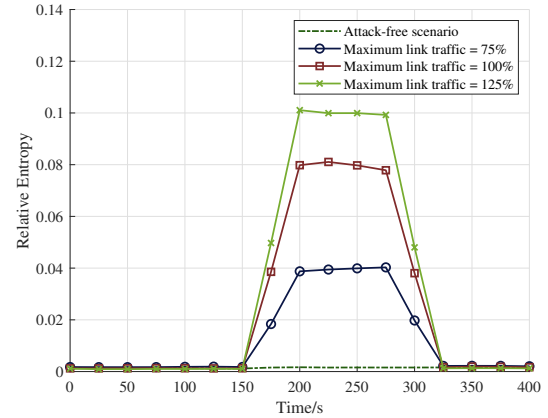


Figure 3. Relative entropy of the network in LFA detection.

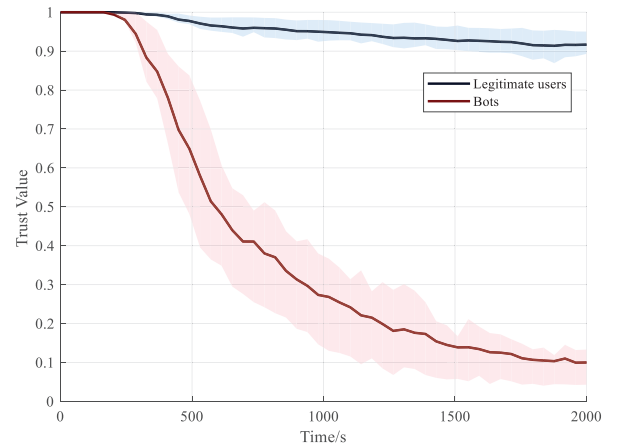


Figure 4. Average trust values of legitimate users and bots.

values of legitimate users and bots in 50 experiments. The line marked in red indicates the average trust value of the bots and the light red boundaries in the figure indicate the quartile points of the trust value distribution. Similarly, the average trust value of legitimate users is marked in blue. Experimental results show that during the attack process, the trust values of bots decrease apparently while the trust values of legitimate users maintain at a relatively high level, therefore the trust evaluation module is able to distinguish users and bots with high precision.

The trust evaluation module is also effective to distinguish the behavior of target servers and decoy servers. Figure 5 shows the trust values of servers in target area and decoy servers in 50 experiments. The line marked in red indicates the average trust value of decoy servers and the line marked in blue indicates the average trust value of servers in the target area. Since

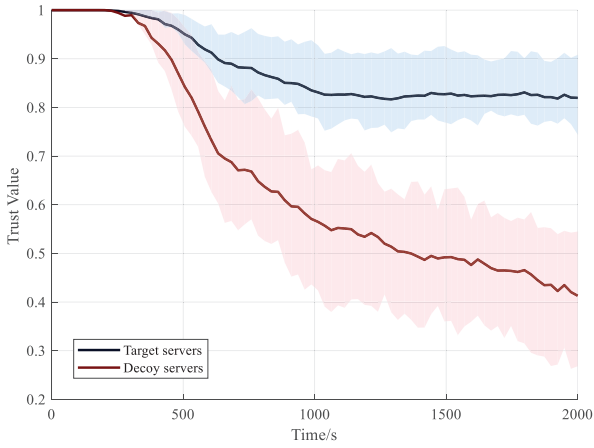


Figure 5. Average trust values of target servers and decoy servers.

in our experimental settings, the bots may also generate malicious traffic to servers in the target area, we set the parameters as $m = 4$, $n = 8$ and $c = 0.1$ to alleviate misjudgment of target servers. Figure 4 and Figure 5 show that it takes more time to distinguish the behavior of target servers and decoy servers and the accuracy is relatively lower since the trust values in Figure 5 have a broader range.

4.4 LFA Mitigation Performance

The LFA mitigation module mainly contains two stages, blocking the suspicious traffic at border switches and deploying rerouting schemes. We first analyze the performance of the classification of suspicious traffic. We label malicious traffic as positive samples and legitimate traffic as negative samples in the experiment. By setting different thresholds θ , the samples can be classified as True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). The true positive rate (TPR) and false positive rate (FPR) can be represented as

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}. \quad (12)$$

To illustrate the performance of our system with different selections of threshold θ , we apply the receiver operating characteristic (ROC) curve in the experiments. We also compare our method with support vector machine (SVM) [19]. The comparison of the ROC curves of SVM and our method is shown in Figure 6. The area under curve (AUC) of SVM method is 0.9689

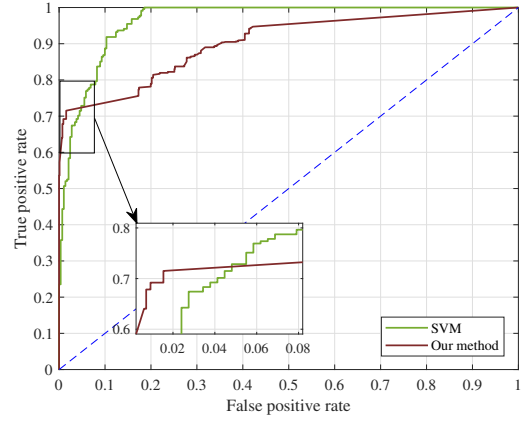


Figure 6. ROC curves of SVM and our method.

and the AUC of our method is 0.9148, which is relatively lower but still acceptable. Our method outperforms the SVM method for the following two reasons. First, SVM is a supervised learning method, therefore the model needs to be trained proactively. However, the label for traffic flows is difficult to obtain and may not be accurate, and the model may fail when the attacker changes the attack strategy. In contrast, our method is an unsupervised learning method and does not need to label the traffic flow. Second, the defense method should have an acceptable false positive rate and do not affect the legitimate traffic. Figure 6 shows that the true positive rate of our method is higher than SVM when the false positive rate is lower than 2%. Since the selection of the threshold must ensure that the system has a low false positive rate, our method exhibits a better performance. In our experiments, the precision, recall, and F1-score of our method are 68.46%, 98.56%, and 80.79% respectively. In comparison, the precision, recall, and F1-score of SVM method under the same FPR are 59.36%, 98.34%, and 74.03% respectively. Due to the consideration of complexity, we do not simply focus on achieving a high precision rate compared with the state-of-the-art ML-based methods. Our method aims to decrease the influence on legitimate traffic, and is able to mitigate the congestion in the rerouting stage.

The second stage of our LFA mitigation module is to reroute the traffic to alleviate the congestion. Figure 7 shows the comparison of link utilization rate after deploying different LFA mitigation methods. In the experiment, the attacker selects Link 1 as the target link and launches an LFA. After blocking the sus-

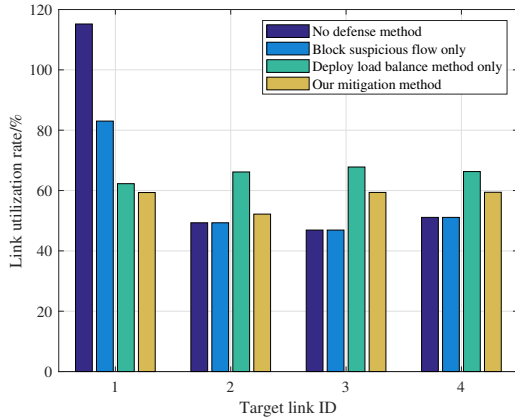


Figure 7. Comparison of link utilization rates after deploying different LFA mitigation methods.

picious traffic, the link utilization rate of the target links decreases effectively, but cannot completely resolve the congestion. By deploying the load balancing method, the average link utilization rate decreases to 60%, which can ensure the operation of the network. Additionally, by blocking the suspicious traffic, the average link utilization rate decreases by 10%, which can not only conserve the bandwidth resources, but also reduce the scale of the optimization problem in the following rerouting stage.

To illustrate the effectiveness of our routing method, we compare the performance of our mitigation strategy with Woodpecker [25]. Assume that the transmission delay of each link in our experimental network is 10ms. Figure 8 shows the performance comparison of average transmission delay for legitimate flows after deploying different LFA rerouting strategies. Results show that by deploying the rerouting strategy derived from the centralized optimization problem 11, the average transmission delay decreases at the rate of 8.60%, 4.98% and 5.56% when the maximum link traffic of target links is set as 75%, 100% and 125% respectively. Additionally, if each border switch formulates an optimization problem and derives the routing strategy for flows from the particular switch, the extra time consumption rates for our distributed routing method are 3.22%, 0.24% and 2.14% in three experiments. Table 1 shows the time costs of determining the rerouting strategy with different methods. Results show that our distributed routing method is effective and can reduce the scale of the optimization problem, which increases the scalability of our mitigation sys-

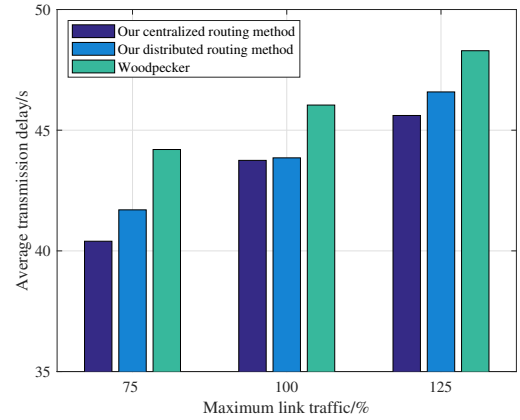


Figure 8. Performance comparison of average transmission delay for legitimate flows.

Table 1. Time cost comparisons of different methods.

Method	Time Cost(ms)
Woodpecker	0.4331
Our centralized routing method	0.3068
Our distributed routing method	0.0873

tem in large scale networks.

After implementing our mitigation strategies, we employ the relative entropy analysis to evaluate the effectiveness of the mitigation phase. Figure 9 illustrates the relative entropy throughout the detection and mitigation process. In all three experiments, there is a noticeable decrease in the relative entropy value, indicating the effectiveness of our LFA defense system in mitigating LFAs.

V. CONCLUSION

In this paper, we have proposed a novel framework to defend against LFAs in SDN environments. Our framework comprises two stages, namely the detection stage and the mitigation stage. For LFA detection, we propose a relative entropy-based method to analyze network traffic distribution. For LFA mitigation, our method evaluates the trust values of UEs and establishes different routing strategies for flows based on their trust values. Our framework blocks the suspicious traffic at border switches of SDN, and can derive the rerouting schemes distributedly at the network entrance. The defense system is lightweight to implement, thereby increases the scalability. Experimental results have shown that our method is effective in de-

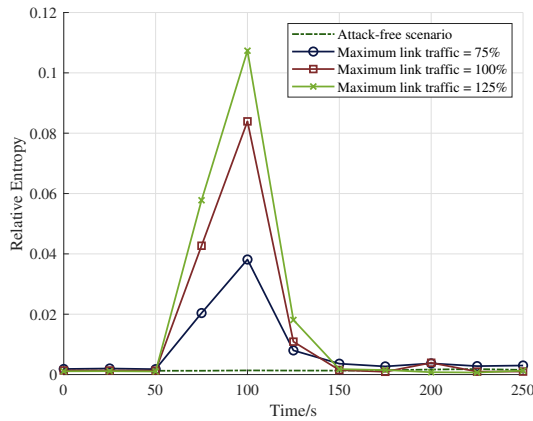


Figure 9. Relative entropy of the network in LFA mitigation.

fending against LFAs, comparing with state-of-the-art methods.

In application, since our method deploys a rerouting strategy for each flow on the target link, the switches are required to have adequate capacity to store newly added rules or have the capability to group and merge the flow entries. Additionally, due to considerations of resource consumption, the rerouting strategies derived from distributed optimization problems are not guaranteed to be the global optimal solution. In future work, we will conduct in-depth research to address these issues.

ACKNOWLEDGEMENT

This work was supported in part by the National Key R&D Program of China under Grant 2018YFA0701601, and in part by the National Natural Science Foundation of China (Grant No. 62201605, 62341110, U22A2002), and in part by Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute.

References

- [1] W. Feng, Y. Wang, *et al.*, “Structured satellite-uav-terrestrial networks for 6g internet of things,” *IEEE Network*, 2024.
- [2] Y. Wang, W. Feng, *et al.*, “Hybrid satellite-uav-terrestrial networks for 6g ubiquitous coverage: A maritime communications perspective,” *IEEE Journal on Selected Areas in Communications*, 2021, vol. 39, no. 11, pp. 3475–3490.
- [3] C. Liu, W. Feng, *et al.*, “Mec-empowered non-terrestrial network for 6g wide-area time-sensitive internet of things,” *Engineering*, 2022, vol. 8, pp. 96–107.

- [4] X. Chen, W. Feng, *et al.*, “Zero trust architecture for 6g security,” *IEEE Network*, 2023.
- [5] W. Feng, Y. Lin, *et al.*, “Radio map-based cognitive satellite-uav networks towards 6g on-demand coverage,” *IEEE Transactions on Cognitive Communications and Networking*, 2023.
- [6] T. L. Ericsson, “Ericsson mobility report november 2023,” in *Ericsson Mobility Report*. Ericsson, 2023.
- [7] Y. Ma, X. Chen, *et al.*, “Ddos detection for 6g internet of things: Spatial-temporal trust model and new architecture,” *China Communications*, 2022, vol. 19, no. 5, pp. 141–149.
- [8] Y. Guo, Y. Wang, *et al.*, “Intelligent edge network routing architecture with blockchain for the iot,” *China Communications*, 2023, vol. 20, no. 11, pp. 151–163.
- [9] M. Kang, S. Lee, *et al.*, “The crossfire attack,” in *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 127–141.
- [10] M. F. Hyder and T. Fatima, “Towards crossfire distributed denial of service attack protection using intent-based moving target defense over software-defined networking,” *IEEE Access*, 2021, vol. 9, pp. 112 792–112 804.
- [11] X. Chen, L. Xiao, *et al.*, “Ddos defense for iot: A stackelberg game model-enabled collaborative framework,” *IEEE Internet of Things Journal*, 2021, vol. 9, no. 12, pp. 9659–9674.
- [12] A. Rezapour and W. Tzeng, “RI-shield: Mitigating target link-flooding attacks using sdn and deep reinforcement learning routing algorithm,” *IEEE Transactions on Dependable and Secure Computing*, 2021, vol. 19, no. 6, pp. 4052–4067.
- [13] H. Sun, X. Chen, *et al.*, “A low-cost detection and mitigation method for link flooding attacks in sdn,” in *2023 IEEE 23rd International Conference on Communication Technology (ICCT)*, 2023, pp. 1225–1230.
- [14] K. Kalkan, G. Gur, *et al.*, “Defense mechanisms against ddos attacks in sdn environment,” *IEEE Communications Magazine*, 2017, vol. 55, no. 9, pp. 175–179.
- [15] S. ScottHayward, S. Natarajan, *et al.*, “A survey of security in software defined networks,” *IEEE Communications Surveys & Tutorials*, 2015, vol. 18, no. 1, pp. 623–654.
- [16] M. A. Aglan, M. A. Sobh, *et al.*, “Reliability and scalability in sdn networks,” in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, 2018, pp. 549–554.
- [17] T. Hirayama, K. Toyoda, *et al.*, “Fast target link flooding attack detection scheme by analyzing traceroute packets flow,” in *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015, pp. 1–6.
- [18] D. Belabed, M. Bouet, *et al.*, “Centralized defense using smart routing against link-flooding attacks,” in *2018 2nd Cyber Security in Networking Conference (CSNet)*, 2018, pp. 1–8.
- [19] X. Ding, F. Xiao, *et al.*, “Active link obfuscation to thwart link-flooding attacks for internet of things,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 217–224.
- [20] V. Venkateswaran, C. Huang, *et al.*, “Security management in content distribution networks: A delay-variance reduction approach for content mirror site placement,” in *2023 15th International Conference on Communication Systems*

- & Networks (COMSNETS), 2023, pp. 505–512.
- [21] J. Kim, J. Nam, *et al.*, “Bottlenet: Hiding network bottlenecks using sdn-based topology deception,” *IEEE Transactions on Information Forensics and Security*, 2021, vol. 16, pp. 3138–3153.
 - [22] C. Liaskos and S. Ioannidis, “Network topology effects on the detectability of crossfire attacks,” *IEEE Transactions on Information Forensics and Security*, 2018, vol. 13, no. 7, pp. 1682–1695.
 - [23] S. Murtuza and K. Asawa, “Early prevention and mitigation of link flooding attacks in software defined networks,” *Journal of Network and Computer Applications*, 2024, vol. 224, p. 103832.
 - [24] L. Xue, X. Ma, *et al.*, “Linkscope: Toward detecting target link flooding attacks,” *IEEE Transactions on Information Forensics and Security*, 2018, vol. 13, no. 10, pp. 2423–2438.
 - [25] L. Wang, Q. Li, *et al.*, “Woodpecker: Detecting and mitigating link-flooding attacks via sdn,” *Computer Networks*, 2018, vol. 147, pp. 1–13.
 - [26] N. Ravi, S. M. Shalinie, *et al.*, “Balance: Link flooding attack detection and mitigation via hybrid-sdn,” *IEEE Transactions on Network and Service Management*, 2020, vol. 17, no. 3, pp. 1715–1729.
 - [27] R. U. Rasool, U. Ashraf, *et al.*, “Cyberpulse: A machine learning based link flooding attack mitigation system for software defined networks,” *IEEE Access*, 2019, vol. 7, pp. 34 885–34 899.
 - [28] E. Aydın and Ş. Bahtiyar, “Ocids: An online cnn-based network intrusion detection system for ddos attacks with iot botnets,” in *2021 14th International Conference on Security of Information and Networks (SIN)*, 2021, vol. 1, pp. 1–8.
 - [29] W. Yu and B. Zhou, “Using long-short-term memory to effectively identify persistent routes under stealthy link flooding attacks in software-defined networks,” in *2023 IEEE Symposium on Computers and Communications (ISCC)*, 2023, pp. 1007–1012.
 - [30] V. Gaur and R. Kumar, “Ddoslstm: Detection of distributed denial of service attacks on iot devices using lstm model,” in *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 2022, pp. 1–7.
 - [31] Y. Luo, X. Chen, *et al.*, “Transformer-based malicious traffic detection for internet of things,” in *ICC 2022-IEEE International Conference on Communications*, 2022, pp. 4187–4192.
 - [32] Y. Luo, H. Sun, *et al.*, “Detecting while accessing: A semi-supervised learning-based approach for malicious traffic detection in internet of things,” *China Communications*, 2023, vol. 20, no. 4, pp. 302–314.
 - [33] M. Rezazad, M. R. Brust, *et al.*, “Detecting target-area link-flooding ddos attacks using traffic analysis and supervised learning,” in *Advances in Information and Communication Networks: Proceedings of the 2018 Future of Information and Communication Conference (FICC)*, 2019, vol. 2, pp. 180–202.
 - [34] J. Xie, L. Wei, *et al.*, “Lfa-cleaner: Mitigating link flooding attacks in software-defined networks,” in *2023 International Conference on Mobile Internet, Cloud Computing and Information Security (MICCIS)*, 2023, pp. 154–161.
 - [35] W. Cao, Y. Huang, *et al.*, “A blockchain based link-flooding attack detection scheme,” in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2021, vol. 4, pp. 1665–1669.
 - [36] X. Jiang, Q. Shi, *et al.*, “Credible link flooding attack detection and mitigation: A blockchain-based approach,” *IEEE Transactions on Network and Service Management*, 2024.
 - [37] C. Liaskos, V. Kotronis, *et al.*, “A novel framework for modeling and mitigating distributed link flooding attacks,” in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.
 - [38] J. Wang, R. Wen, *et al.*, “Detecting and mitigating target link-flooding attacks using sdn,” *IEEE Transactions on Dependable and Secure Computing*, 2018, vol. 16, no. 6, pp. 944–956.
 - [39] J. Wang, Y. Liu, *et al.*, “Relfa: Resist link flooding attacks via renyi entropy and deep reinforcement learning in sdn-iot,” *China Communications*, 2022, vol. 19, no. 7, pp. 157–171.
 - [40] Y. Li, Y. Liu, *et al.*, “In-network collaborative link flooding attack defense with adaptive anomaly analysis,” in *GLOBE-COM 2023-2023 IEEE Global Communications Conference*, 2023, pp. 1717–1722.
 - [41] G. Yang, H. Hosseini, *et al.*, “Modeling and mitigating the coremlt attack,” in *2018 Annual American Control Conference (ACC)*, 2018, pp. 3410–3416.
 - [42] X. Chen, W. Feng, *et al.*, “Defending against link flooding attacks in internet of things: A bayesian game approach,” *IEEE Internet of Things Journal*, 2021, vol. 9, no. 1, pp. 117–128.
 - [43] X. Wang, X. Ma, *et al.*, “On modeling link flooding attacks and defenses,” *IEEE Access*, 2021, vol. 9, pp. 159 198–159 217.
 - [44] X. Ma, B. An, *et al.*, “Randomized security patrolling for link flooding attack detection,” *IEEE Transactions on Dependable and Secure Computing*, 2019, vol. 17, no. 4, pp. 795–812.
 - [45] S. Lee, M. Kang, *et al.*, “Codef: Collaborative defense against large-scale link-flooding attacks,” in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, 2013, pp. 417–428.
 - [46] C. Basescu, R. M. Reischuk, *et al.*, “Sibra: Scalable internet bandwidth reservation architecture,” in *Symposium on Network and Distributed System Security (NDSS 2016)*, 2016.
 - [47] L. Xie, S. Meng, *et al.*, “Differential pricing strategies for bandwidth allocation with lfa resilience: A stackelberg game approach,” *IEEE Transactions on Information Forensics and Security*, 2023.
 - [48] H. Zhou, S. Hong, *et al.*, “Mew: Enabling large-scale and dynamic link-flooding defenses on programmable switches,” in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 3178–3192.
 - [49] J. Edmonds and R. M. Karp, “Theoretical improvements in algorithmic efficiency for network flow problems,” *Journal of the ACM (JACM)*, 1972, vol. 19, no. 2, pp. 248–264.
 - [50] N. Koroniotis, N. Moustafa, *et al.*, “Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset,” *Future Generation Computer Systems*, 2019, vol. 100, pp. 779–796.
 - [51] W. Sun and C. Wang, “An improved mean shift clustering

algorithm for lfa detection,” in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2021, pp. 926–934.

Biographies



Sun Hancun received the B.S. degree from the Department of Electronic Engineering, Tsinghua University in 2022, where he is currently pursuing the Ph.D degree. His research interests include network security and communication.



Chen Xu received the B.S. and M.S. degrees from the School of Mathematical Sciences, Peking University in 2009 and 2013, respectively, and the Ph.D degree from the Department of Electronic Engineering, Tsinghua University in 2022. He has worked as an assistant research fellow for several years. He is currently a Postdoctoral Researcher, Tsinghua University. His research interests include game theory, network management, and security.



Luo Yantian received the B.S. and Ph.D degrees from the Department of Electronic Engineering, Tsinghua University, in 2018 and 2023, respectively. His research interests include machine learning and network security.



Ge Ning received the B.S. and Ph.D degrees from Tsinghua University, Beijing, China, in 1993 and 1997, respectively. From 1998 to 2000, he was with ADC Telecommunications, Dallas, TX, USA, where he researched the development of ATM switch fabric ASIC. Since 2000, he has been with the Department of Electronics Engineering, Tsinghua University, where he is currently a Professor and also serves as the Director of Communication Institute. He has published over 100 papers. His current research interests include communication ASIC design, short range wireless communication, and wireless communications. Dr. Ge is a Senior Member of the China Institute of Communications and the Chinese Institute of Electronics.