Light-Weight DDoS Mitigation at Network Edge with Limited Resources

1st Ryo Yaegashi
3rd Yu Nakayama
Department of Computer and Information Sciences
Tokyo University of Agriculture and Technology
Tokyo, Japan
{ryo.y, yu.nakayama}@ieee.org

2nd Daisuke Hisano
Graduate School of Engineering
Osaka University
Osaka, Japan
hisano@ieee.org

Abstract—The Internet of Things (IoT) has been growing rapidly in recent years. With the appearance of 5G, it is expected to become even more indispensable to people's lives. In accordance with the increase of Distributed Denial-of-Service (DDoS) attacks from IoT devices, DDoS defense has become a hot research topic. DDoS detection mechanisms executed on routers and SDN environments have been intensely studied. However, these methods have the disadvantage of requiring the cost and performance of the devices. In addition, there is no existing DDoS mitigation algorithm on the network edge that can be performed with the low-cost and low performance equipments. Therefore, this paper proposes a light-weight DDoS mitigation scheme at the network edge using limited resources of inexpensive devices such as home gateways. The goal of the proposed scheme is to simply detect and mitigate flooding attacks. It utilizes unused queue resources to detect malicious flows by random shuffling of queue allocation and discard the packets of the detected flows. The performance of the proposed scheme was confirmed via theoretical analysis and computer simulation. The simulation results match the theoretical results and the proposed algorithm can efficiently detect malicious flows using limited resources.

Index Terms—Internet of Things, Communication system security, Computer network management, Queueing analysis

I. INTRODUCTION

The Internet of Things (IoT) has been rapidly growing over the past decade to change our lives and society. Various and enormous smart embedded devices such as sensors are deployed everywhere and communicate with edge and/or cloud servers via the Internet. This paradigm will be dominant in the era of 5G and beyond to support global services in a wide variety of areas including industry, education, logistics, government and so on.

Over the past years, Distributed Denial-of-Service (DDoS) attacks have been increasing in both frequency and volume [1]. DDoS is a network attack for disrupting legitimate services by flooding traffic to targeted servers. IoT devices are used by malicious attackers for a powerful amplifying platform of large-scale cyber attacks [2]. A recent prominent example of such attacks is the Mirai botnet, which was identified in 2016 [4]. Such botnets spread among IoT devices including consumer household devices by means of brute force [5]. IoT botnets have continuously emerged by malwares such as Mirai variants [6]. DDoS defense against attacks from IoT devices is a significant research topic.

Many DDoS defense strategies have been proposed and developed. Among them, DDoS mitigation at the network edge is a promising solution for reducing the amount of malicious traffic in the Internet. The typical parameters used for monitoring incoming traffic to detect and identify malicious traffic are source IP address, traffic increasing degree, and similarity among the traffic [7]. Also, packet-level machine learning technique for anomaly detection was proposed to distinguish normal and attack traffic [8]. An edge computing approach was proposed to gather information about incoming traffic and communicate the collected information with nearby detection services [9]. A major drawback of these learning-based DDoS detection mechanisms is the cost for data collection and learning. The processing load is also a significant issue for resource-limited devices at the network edge such as home gateways and inexpensive layer-2 switches.

To address this problem, this paper proposes a light-weight DDoS mitigation mechanism at the network edge using limited resources of inexpensive devices. The goal of the proposed scheme is to simply detect and mitigate flooding attacks such as UDP flood. It detects malicious flows by random shuffling of queue allocation, and the packets of detected flows are discarded without queuing in the same way as typical active queue management schemes. It enables early mitigation of DDoS attacks at the network edge to prevent them from flowing into the network. The proposed shuffling algorithm is simple and easy to execute in low-cost equipment without any additional hardware, because it leverages queue resources which are generally not fully utilized. The contribution of this work is to propose a simple and
light-weight DDoS mitigation mechanism which can be employed together with existing DDoS defense schemes.

This paper is organized as follows. Related work and the contribution of this paper are introduced in section II. The proposed DDoS mitigation scheme is explained in section III. The performance of the proposed scheme is theoretically analyzed in section IV. The feasibility of the proposed idea is evaluated through computer simulations in section V. The conclusion of this paper is provided in section VI.

II. RELATED WORK

There have been many research efforts on DDoS detection and mitigation. A router-based defense mechanism called Pushback was developed in [10], where functionality is added to each router to detect attack from congestion signature. Perimeter-based defense mechanisms proposed in [11] enabled Internet service providers (ISPs) to provide anti-DDoS services to their customers. In these mechanisms, edge routers cooperatively identify flooding sources to establish rate-limit filters to block attack traffic. A collaborative defense mechanism was also proposed in [12] to enable routers to distinguish attack flows from legitimate flows.

DDoS mitigation has been a significant issue for software defined network (SDN) because of the centralized control architecture [13]. A distributed collaborative framework was presented in [14] to allow customers to request DDoS mitigation service from ISPs. In this framework, ISPs change the label of the anomalous traffic upon request and redirect them to security middleboxes, while attack detection and analysis modules are deployed at customer side. In addition, a detection method by using the entropy change of the destination IP address was also proposed in [15]. For IoT environments, an enhanced distributed low-rate attack mitigating (eDLAM) mechanism was proposed [16] based on a game model to analyze the attack benefit between attacker and defender.

However, there have been few research on simple DDoS mitigation mechanisms that can easily be executed in low-cost devices at the network edge. Complicated operations including pattern-matching and cooperative approaches are unsuitable for such resource-limited devices. Therefore, the contribution of this paper is to propose a light-weight DDoS mitigation mechanism for the IoT era. Note that the proposed scheme can be employed together with existing DDoS defense mechanisms mentioned above. The detail of the proposed idea is introduced in the next section.

III. PROPOSED DDoS MITIGATION

A. Concept

The goal of the proposed light-weight DDoS mitigation is to simply detect and mitigate flooding attacks such as UDP flood and HTTP flood at the network edge. The main idea is utilizing queuing functions to enable DDoS mitigation using limited resources of low-cost nodes such as switches and gateways. The nodes at the network edge detect malicious flows by random shuffling of queue allocation, and the packets of detected flows are discarded without queuing.

The proposed concept is depicted in Fig. 1. It is assumed that a node has several available queues for mitigation sequence. An escape queue is configured for flow evacuation. First, incoming flows are assigned to available queues as shown in Fig. 1a. If the enqueue rate for a queue exceeds a certain threshold (L), the flows (f1 and f2) are randomly reallocated to available queues. Otherwise, allocated flows (f3, f4, and f5) are detected as legitimate flows to be assigned to the escape queue as shown in Fig. 1b. After the reallocation process, the malicious flow can be identified and discarded to mitigate DDoS attacks, as illustrated in Fig. 1c.

B. Node architecture

The assumed node architecture is depicted in Fig. 2. The requirement is to provide several queues, a switching function, and a scheduler, which is a general architecture for network nodes such as Ethernet switches. For instance, many Ethernet switches provide eight queues for ensuring QoS among priority classes of traffic, whereas many of them are unused because generally two or three priority classes are used to identify real-time and best effort traffic. Therefore, a queue is configured as an escape queue from the unused queues. The other queues are defined as available queues for the proposed mitigation process.

C. Mitigation algorithm

1) Variable definition: The variables used in the proposed algorithm are summarized in Table 1. The escape queue is described as Q_escape. Let Q denote the set of available queues, and q ∈ Q is the queue identifier. The current queue size is described as l_q. The number of assigned flows to qth queue is n_q. Also, y_q is a boolean flag for qth queue. Let F denote the set of incoming flows, and f ∈ F is the flow identifier. The queue to which fth flow is assigned is denoted as x_f. The threshold for increase in the queue size is defined as L. Finally, τ is the cycle length.

2) Algorithm: The proposed DDoS mitigation algorithm is described in Algorithm 1. It is cyclically executed with the time interval τ. The proposed algorithm consists of two steps.

In the first step, the increase in the queue size is evaluated for each available queue. If the increase (l_q − l^{last}_q) exceeds the threshold L, the flag y_q is set as TRUE. Otherwise, the flag is set as FALSE.

In the second step, incoming flows are reallocated to available queues. If the flag y_q for the queue to which
(a) Flow allocation

(b) Reallocation

(c) DDoS mitigation

![Fig. 1: Concept of proposed DDoS mitigation.](image)

Fig. 1: Concept of proposed DDoS mitigation.

![Fig. 2: Node architecture.](image)

![TABLE I: Variables](image)

| Variable   | Definition                      |
|------------|---------------------------------|
| $Q_{\text{escape}}$ | Escape queue                   |
| $Q$        | Set of available queues        |
| $q$        | Queue identifier ($q \in Q$)   |
| $l_q$      | Size of $q$th queue            |
| $n_q$      | Number of flows assigned to $q$th queue |
| $y_q$      | Boolean flag for $q$th queue   |
| $F$        | Set of incoming flows          |
| $f$        | Flow identifier ($f \in F$)    |
| $x_f$      | Assigned queue for $f$th flow  |
| $L$        | Threshold for queue size       |
| $\tau$     | Cycle length                   |

$\text{Algorithm 1 DDoS mitigation algorithm}$

1: # Step 1: Queue size evaluation
2: for $q \in Q$ do
3: if $l_q - l_{\text{last}}^q > L$ then
4: $y_q \leftarrow \text{TRUE}$
5: else
6: $y_q \leftarrow \text{FALSE}$
7: end if
8: $l_{\text{last}}^q \leftarrow l_q$
9: end for
10: # Step 2: Queue reallocation
11: for $f \in F$ do
12: if $y_{x_f} = \text{TRUE}$ then
13: if $n_{x_f} = 1$ then
14: $x_f \leftarrow \text{Drop}$
15: else
16: $x_f \leftarrow \text{Random}(q \in Q)$
17: end if
18: else
19: $x_f \leftarrow Q_{\text{escape}}$
20: end if
21: end for

$\text{d, f, g, h}$th flow is assigned ($x_f$) is $\text{TRUE}$, the flow is reallocated. If $n_{x_f} = 1$, $f$th flow is detected as a malicious flow, and the node starts to drop it. Otherwise, this flow is randomly assigned to available queues $q \in Q$. If $y_{x_f}$ is $\text{FALSE}$, $f$th flow is detected as a legitimate flow. Then, it is assigned to the escape queue $Q_{\text{escape}}$.

With this algorithm legitimate flows are allocated to the escape queue, while malicious flows are detected and dropped to mitigate DDoS attacks.

IV. THEORETICAL ANALYSIS

The performance of the proposed scheme is theoretically analyzed in this section.

A. Accuracy for DDoS detection

With the proposed algorithm, a malicious flow can be identified if only that flow is enqueued into an available queue. Thus, the DDoS detection accuracy in a cycle is equal to the probability that only a malicious flow is assigned to an available queue. Let $c$ denote the cycle identifier for executing Algorithm 1.
Here we define \( p_{q,c} \) as the probability that only a flow is assigned to \( q \)th queue at \( c \)th cycle. Since the allocation probability for each queue is uniform, \( p_{q,c} \) is described as
\[
p_{q,c} = \frac{1}{N_Q} \left( \frac{N_Q - 1}{N_Q} \right)^{N_{F,c} - 1},
\]
where \( N_Q = |Q| \) is the number of available queues and \( N_{F,c} \) denotes the number of flows that satisfy \( x_f = q \in Q \), i.e., non-escaped flows, at \( c \)th cycle. Let \( p_c \) denote the probability that a flow is assigned to any of available queues and no other flow is assigned to the same queue at \( c \)th cycle, which is calculated as
\[
p_c = \sum_{q \in Q} p_{q,c}
= \left( \frac{N_Q - 1}{N_Q} \right)^{N_{F,c} - 1}.
\]

Let \( U_c \) and \( E_c \) denote the expected number of undetected and detected malicious flows at \( c \)th cycle, respectively. The relationship between these variables is
\[
U_{c+1} = \text{Max}(0, U_c - E_c).
\]
Note that \( U_0 = \rho N_{F,0} \) where \( \rho \) denotes the ratio of malicious flows to all incoming flows. \( E_c \) is described as
\[
E_c = U_c p_c.
\]

### B. Escape probability

Assuming that the threshold \( L \) is sufficiently large, all the flows that are not enqueued to the same queue with malicious flows can be assigned to the escape queue. The increase in the number of enqueued flows to the escape queue improves the DDoS detection accuracy as formulated in (4).

The expected number of assigned flows to a queue at \( c \)th cycle is simply calculated as
\[
n_{q,c}^e = \frac{N_{F,c}}{N_Q}.
\]
The probability that there is no malicious flow among them is
\[
\gamma_{q,c} = \left( 1 - \frac{U_c}{N_{F,c}} \right)^{n_{q,c}^e}.
\]

Therefore, the expected value of escaped flows at \( c \)th cycle is
\[
V_c = \sum_{q \in Q} n_{q,c}^e \gamma_{q,c}
= N_{F,c} \gamma_{q,c}.
\]

### C. Time variation

From the number of non-escaped flows \( N_{F,c} \), the numbers of detected malicious flows and newly escaped flows are reduced at the next cycle. Thus, \( N_{F,c+1} \) is described as
\[
N_{F,c+1} = \text{Max}(0, N_{F,c} - E_c - V_c).
\]
Since both \( E_c \) and \( V_c \) are the functions of \( N_{F,c} \), the DDoS detection accuracy at each cycle can be computed with these equations.

### V. Computer simulation

#### A. Simulation condition

The performance of the proposed scheme was confirmed with computer simulation. First, we verified the consistency with the theoretical performance formulated in (4) and (8). The number of incoming flows \( |F| \) was set to 16, 32, and 64 to evaluate the detection accuracy in different conditions. The number of available queues was set as \( N_Q = 6 \).

Second, the performance in dynamic conditions was evaluated. The number of available queues was set as \( N_Q = 7 \). The number of incoming flows \( |F| \) was 64 in the initial condition, and additional flows arrive during \( 1 \leq c \leq 15 \). The number of additional flows was set as 12 and 24 per cycle. All adding flows were assumed to be legitimate flows. From (7), the expected value of escaped flows is calculated as \( V_c = 24 \) when \( N_Q = 7 \) and \( |F| = 64 \); the performance limitation of the proposed scheme for this configuration is 24 additional flows per cycle.

Finally, to clarify the marginal performance, the relationship between the numbers of incoming flows and available queues was evaluated. We define the processed flows as the total of escaped flows and detected malicious flows. We set the number of incoming flows \( |F| = 128 \). The number of processed flows in 30 cycles was measured with different number of available queues.

The other parameters commonly used in these simulations were as follows. Threshold for queue size \( L \) was set at 62.5 MB. The upstream link bandwidth was 50 Mbps. The enqueued data size of legitimate flows during each cycle was randomly determined using a normal distribution with a mean 2.0 Mbps and a standard deviation 0.5 Mbps. The data rate of malicious flows was set at 100 Mbps assuming UDP flooding attacks. The ratio of malicious flows in all incoming flows was set at \( \rho = 0.1 \). The cycle length \( \tau \) was 10s, and the total simulation time was 300s. The simulation was iterated 1000 times for each condition. The parameters explained above are summarized in Table III.

#### B. Simulation results

Figs. 3a–3c show the theoretical and simulated values of detected malicious flows in each cycle with \( |F| = 1000 \) times for each condition. The parameters explained above are summarized in Table III.
16, 32, and 64. Figs. 4a–4c show the theoretical and simulated values of non-escaped flows in each cycle. The points represent the average number of detected malicious flows and non-escaped flows, and the errorbars represent the standard deviation. We can see that the theoretical values are settled within the ranges at any cycle. Note that the variation between the theoretical and simulated results is due to the randomness of the simulation. Thus, it was confirmed from these results that the proposed scheme properly works to detect and mitigate DDoS attacks.

Figs. 5a–5b show the theoretical and simulated results for dynamic conditions. The points represent the average number of detected malicious flows and non-escaped flows, and the errorbars represent the standard deviation. The simulation results match the theoretical values. When 12 flows were added per cycle, they can be gradually escaped with the proposed algorithm. Since the performance limit with $Q_N = 7$ is 24 escaped flows per cycle, the proposed scheme holds on during $c \leq 15$ and DDoS detection proceeds after 15th cycle as shown in Fig. 5b. These results imply that the proposed scheme works even in dynamic environments if the number of additional flows is less than the performance limit.

Fig. 6 shows the theoretical and simulated values of processed flows after the 30th cycle with $N_Q = 3, 4, 5, 6$ and 7. The number of processed flows increases in accordance with the increase of available queues. Most of the flows are escaped or dropped when $N_Q = 7$. We can also see that about 85% of the incoming flows are assigned to the escape queue even when the number of available queues is 6. Hence, it is thought that more than 100 flows can be identified in 30 cycles if the number of available queues is greater than 5. It is implied that the proposed scheme provides enough scalability for the network edge focused in this paper.

### VI. Conclusion

In recent years, the number of DDoS attacks has been increasing. DDoS defense mechanisms against attacks from IoT devices have become an important issue, and various methods have been proposed and developed. However, there have been no DDoS mitigation schemes for low-cost and low-performance devices at the network edge such as home gateways. To address this problem, in this paper we proposed a lightweight DDoS mitigation mechanism at the network edge using the limited resources of inexpensive devices. This mechanism detects

---

**TABLE II: Simulation parameters**

| Parameter   | Value         |
|-------------|---------------|
| Threshold $L$ | 62.5 MB       |
| Link speed  | 50 Mbps       |
| Ratio of DDoS $\rho$ | 0.1       |
| Cycle length $\tau$ | 10 s       |
| Simulation time | 300 s     |
| Number of iterations | 1000   |
malicious flows by random shuffling of queue allocation, and the packets of detected flows are discarded without queuing. It enables simple detection and mitigation of flooding attacks such as UDP flood. It was confirmed through computer simulation that the proposed algorithm identified more than 100 flows by using 7 queues for queue assignment. With the proposed scheme, DDoS attacks can be simply mitigated even in low-cost devices with limited resources.

ACKNOWLEDGMENT
A part of this work This work was supported by JST, ACT-I, Grant Number JPMJPR18UL, Japan.

REFERENCES
[1] A. Wang, W. Chang, S. Chen, and A. Mohaisen, “Delving into internet DDoS attacks by botnets: characterization and analysis,” IEEE/ACM Transactions on Networking, vol. 26, no. 6, pp. 2843–2855, 2018.
[2] D. Peraković, M. Periša, and I. Cvitić, “Analysis of the IoT impact on volume of DDoS attacks,” XXXVI Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju–Postel, vol. 2015, pp. 295–304, 2015.
[3] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, “Botnet in DDoS attacks: trends and challenges,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2242–2270, 2015.
[4] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other botnets,” Computer, vol. 50, no. 7, pp. 80–84, 2017.
[5] M. Lyu, D. Sherratt, A. Sivanathan, H. H. Gharakhelli, A. Rafford, and V. Sivaraman, “Quantifying the reflective DDoS attack capability of household IoT devices,” in Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2017, pp. 46–51.
[6] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, “Analysis of DDoS-capable IoT malwares,” in 2017 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, 2017, pp. 807–816.
[7] C. Zhang and R. Green, “Communication security in Internet of Thing: preventive measure and avoid DDoS attack over IoT network,” in Proceedings of the 18th Symposium on Communications & Networking, 2015, pp. 8–15.
[8] R. Doshi, N. Apthorpe, and N. Fearnley, “Machine learning DDoS detection for consumer Internet of Things devices,” in 2018 IEEE Security and Privacy Workshops (SPW). IEEE, 2018, pp. 29–35.
[9] K. Bhardwaj, J. C. Miranda, and A. Garvillova, “Towards IoT-DDoS prevention using edge computing,” in {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18), 2018.
[10] J. Ioannidis and S. M. Bellovin, “Implementing pushback: Router-based defense against DDoS attacks,” in Proceedings of NDSS, 2002.
[11] S. Chen and Q. Song, “Perimeter-based defense against high bandwidth DDoS attacks,” IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 6, pp. 526–537, 2005.
[12] S. B. Lee, M. S. Kang, and V. D. Gilgor, “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.
[13] K. Kalkan, G. Gur, and F. Alagöz, “Defense mechanisms against DDoS attacks in SDN environment,” IEEE Communications Magazine, vol. 55, no. 9, pp. 175–179, 2017.
[14] R. Sahay, G. Blanc, Z. Zhang, and H. Debar, “Towards autonomic DDoS mitigation using software defined networking,” in Network and Distributed System Security (NDSS) Symposium, 2015.
[15] S. M. Mousavi and M. St-Hilaire, “Early detection of DDoS attacks against SDN controllers,” in 2015 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2015, pp. 77–81.
[16] G. Liu, W. Quan, N. Cheng, H. Zhang, and S. Yu, “Efficient DDoS attacks mitigation for stateful forwarding in Internet of Things,” Journal of Network and Computer Applications, vol. 130, pp. 1–13, 2019.