IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 10, 2021, accepted February 8, 2022, date of publication February 22, 2022, date of current version March 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3153067

Quick Suppression of DDoS Attacks by Frame Priority Control in IoT Backhaul With Construction of Mirai-Based Attacks

RINTARO HARADA[1], NAOTAKA SHIBATA[1], SHIN KANEKO[1], KAZUAKI HONDA[1], JUN TERADA[1], YOTA ISHIDA[2], KUNIO AKASHI[2,3], AND TOSHIYUKI MIYACHI[2]

[1]NTT Access Network Service Systems Laboratories, NTT Corporation, Yokosuka, Kanagawa 239-0847, Japan
[2]National Institute of Information and Communications Technology, Nomi, Ishikawa 923-1211, Japan
[3]Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan

Corresponding author: Rintaro Harada (rintarou.harada.zm@hco.ntt.co.jp)

This work was supported by the Research and Development Contract "Wired-and-Wireless Converged Radio Access Network for Massive IoT Traffic" with the Ministry of Internal Affairs and Communications for radio resource enhancement, Japan, under Grant JPJ000254.

ABSTRACT We propose a novel distributed denial of service (DDoS) attack suppression system that significantly reduces discarding of normal traffic (i.e., the traffic from Internet of Things (IoT) devices that are not infected with a malware) with a small number of equipment by controlling the priority of frames in a network accommodating IoT devices. Experimental results showed that our proposed system prevented the discarding of the normal traffic in a few seconds when attack traffic was generated by a traffic generator. Moreover, we constructed Mirai-based DDoS attack traffic and experimentally demonstrated that the discarding of the normal traffic was prevented in 30 milliseconds in our proposed system. We also confirmed that the attack traffic detected by a DDoS protector that was installed in front of an IoT server was autonomously blocked at the switches that the traffic came through from the IoT devices (i.e., the entrances to a backbone network) by integrating various vendors' products.

INDEX TERMS Attack suppression, distributed denial of service (DDoS) attack, Internet of Things (IoT), priority control.

I. INTRODUCTION

Internet of Things (IoT) devices have become increasingly widespread and will be used for various applications in the following 3GPP Release 16/17-based 5G services [1]. IoT devices communicate with an IoT server through IoT-gateways (GWs). The IoT-GWs are located in various areas and accommodated in a backbone network (we call it an "IoT backhaul"). To effectively accommodate numerous IoT devices, the IoT backhaul should be constructed with switches in order to utilize the effects of statistical multiplexing [2].

Security vulnerabilities are inevitable in IoT devices because they must be low-cost. In particular, they tend to have few resources to implement sufficient security measures. These devices are easily infected by malware that manipulate them for distributed denial of service (DDoS) attacks. In a DDoS attack, a malicious user commands a lot of infected IoT devices to simultaneously send IoT server attack traffic such as a UDP flood or an ICMP destination unreachable flood. For example, in the malware named "Mirai" [3], the infected IoT devices rapidly grow in number because of its self-propagation characteristic (i.e., an IoT device infected with Mirai infects other IoT devices in the same network with Mirai), which leads to attack traffic with a very high rate. When the attack traffic arrives at the IoT server, the resources of the IoT server are exhausted and IoT services cannot be provided. Moreover, the attack traffic causes the maximum capacity in the IoT backhaul to be exceeded, which means that some traffic from non-infected IoT devices (i.e., normal traffic) is discarded. This discarding causes retransmissions of normal traffic, which is a critical and urgent problem for IoT devices with a limited battery life.

There are a variety of related works that detect and suppress attack traffic. In [4], DDoS protectors detect and block attack traffic that causes the maximum capacity to be exceeded in the IoT backhaul at the entrances to the IoT backhaul from the IoT-GWs. This system costs a lot because many

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

DDoS protectors are needed. Another system in which DDoS protectors are collocated and the traffic that can be attack traffic (i.e., suspicious traffic) is diverted from the routes on which normal traffic is transmitted to the IoT server can detect and suppress the attack traffic [5], [6]. The estimation of whether the traffic is attack or not is performed by a network controller (NWC) that remotely monitors the overall network. However, this system also costs a lot because it requires many routes dedicated to the suspicious traffic. Therefore, a cost-effective system with a small number of equipment is required. Moreover, the DDoS protectors take a long time to detect the attack traffic because they need to investigate multiple parameters in the traffic. When the capacity in the IoT backhaul keeps being exceeded due to the attack traffic until the DDoS protector detects the attack traffic, the normal traffic is discarded for a long time and the battery life of the IoT devices is reduced because of the retransmission. Therefore, a quick attack suppression system is also required.

In this paper, we propose a novel attack suppression system that significantly reduces the discarding of normal traffic due to the capacity excess in the IoT backhaul by quickly controlling the priority of frames without adding any routes. In this system, the NWC estimates attack traffic more quickly than the conventional NWC and controls the priority of frames so that only suspicious traffic has the minimum priority. Moreover, the switches block attack traffic detected by one DDoS protector installed in front of the IoT server in order to prevent the maximum capacity in the IoT backhaul from being exceeded. The novelty of our system is that it can significantly reduce the discarding of normal traffic by quickly estimating suspicious traffic based only on its rate before the DDoS protector detects attack traffic. In addition to that, our system suppresses suspicious traffic only by controlling the priority of frames until the DDoS protector detects attack traffic, so it does not need any additional switches and fibers.

The rest of this paper is organized as follows: Section II describes conventional systems for suppressing attack traffic. Section III presents our system. Section IV reports experimental results in two environments: in the first, we evaluate the basic performance of our system against attack traffic generated by a traffic generator. In the second, we construct a Mirai-based attack traffic and examine the feasibility of our system. Section V concludes this paper with a brief summary.

II. RELATED WORKS

IoT devices transmit the data to the IoT-GWs using wireless connections and the data are then transferred to the IoT server via the IoT backhaul, which consists of switches. This section describes related works for detecting and suppressing attack traffic.

The attack traffic is detected and suppressed at a DDoS protector. The DDoS protector should precisely and quickly detect the attack traffic to alleviate the impact on normal traffic. Machine learning such as a convolutional neural network (CNN) has been studied to precisely detect attack traffic [7]–[10]. In machine learning, as the DDoS protector learns more patterns of the attack traffic and the normal traffic, it enables to precisely judge whether incoming traffic is the attack traffic or not. For further precise and quick detections, training data of the attack traffic are also utilized in machine learning [11]. The training data allow the DDoS protector to distinguish the attack traffic from the normal traffic precisely and quickly because the training data can reduce time in which the DDoS protector learns the patterns of the attack traffic. These methods with machine learning can be used in any system with the DDoS protectors.

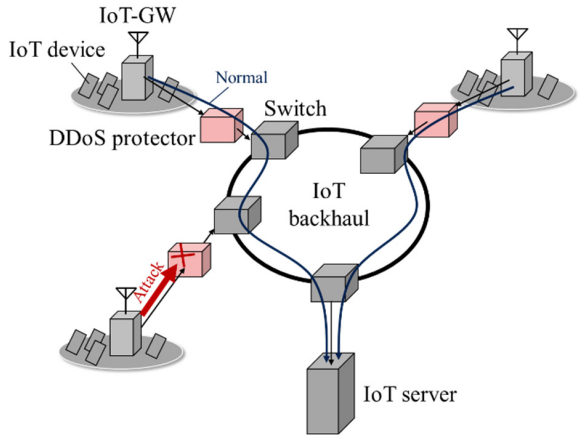
The place where the DDoS protector is deployed is also important. The system described in [4] deploys DDoS protectors in front of each switch that the traffic comes through from the IoT-GWs, as shown in Fig. 1 (a). In this case, after the DDoS protectors detect and block attack traffic, any attack traffic does not enter the IoT backhaul. Therefore, the excess of the maximum capacity due to the inflow of the attack traffic is eliminated. This system costs a lot because it requires the installation of as many DDoS protectors as switches through which traffic comes from the IoT-GWs.

A protection system adding a NWC with a collocated DDoS protector as shown in Fig. 1 (b) is a promising solution [5], [6]. The system reduces the number of DDoS protectors that have to be installed. The NWC monitors each switch and estimates whether attack traffic comes through the IoT backhaul or not. The suspicious traffic, which the NWC estimates as attack traffic, is then diverted to the DDoS protector. If the DDoS protector judges the traffic as the attack traffic, the NWC orders the switch to block the traffic and the inflow of the traffic to the IoT backhaul is prohibited. If the DDoS protector judges the traffic as the normal traffic, the route is reverted and the traffic is transferred to the IoT server as before. If the suspected traffic is actually the normal traffic, the IoT services are temporarily interrupted because the traffic is diverted and does not reach the IoT server. In order to keep providing the IoT services, the diverted traffic should come back to the original routes after passing the DDoS protector and reach the IoT server.

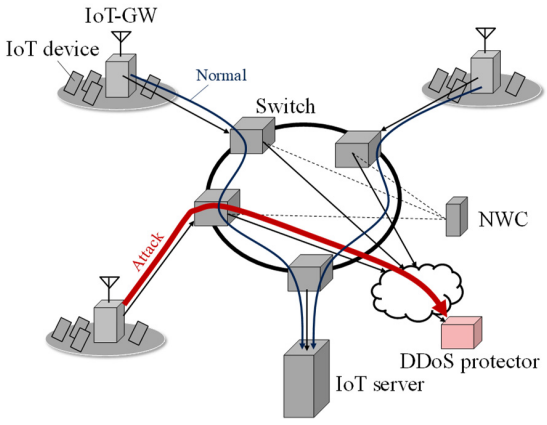
For realizing such a protection system, a dedicated network to the suspicious traffic is required in order to prevent the suspicious traffic from causing the maximum capacity in the IoT backhaul to be exceeded during the transmission to the DDoS protector. The addition of the dedicated network to the existing network costs a lot and takes much time. Moreover, the suspected traffic is transferred via the same route as the normal traffic in front of the IoT server until the DDoS protector precisely detects the attack traffic. If the total rate of suspected and normal traffic exceeds the capacity, some normal traffic is discarded for a long time. Therefore, we propose an attack suppression system that eliminates the dedicated network and reduces the discarding of the normal traffic before the DDoS protector detects the attack traffic.

III. PROPOSED ATTACK SUPPRESSION SYSTEM

Fig. 2 shows the topology of the network with our proposed attack suppression system. A DDoS protector is deployed in



(a) Block attack traffic at entrances to a network



(b) Divert attack traffic at entrances to a network

FIGURE 1. Conventional systems.

front of the IoT server in order to detect any attack traffic to the IoT server. Our system needs only one DDoS protector, unlike the system in section II. An NWC is connected to each switch and the DDoS protector. To reduce discarding of normal traffic due to the capacity excess in the IoT backhaul before the DDoS protector finishes detecting attack traffic, the system includes a quick attack suppression (QAS). The rate of traffic dramatically increases in an attack and the QAS immediately and autonomously executes traffic suppression as soon as the rate of traffic becomes larger than a certain threshold rate.

Fig. 3 shows the time chart of our system. Each switch monitors the rate of incoming traffic at a constant interval. The traffic monitoring is performed for each IoT-GW, which are distinguished by VLAN ID (VID). When the average monitored rate during the monitoring interval becomes larger than the threshold rate, the switches tell the NWC which VIDs exceed the threshold rate. When the NWC receives the VID information, it sends commands for the switches to minimize the priorities of the traffic. If there is any traffic whose priority is the lowest and whose rate does not exceed the threshold rate, the NWC commands the switches to temporarily raise the priority of the traffic by one level to prevent discarding the traffic before the above-mentioned minimization.

After the QAS is executed, the DDoS protector detects the attack traffic. The DDoS protector sends the address information of the attack traffic to the NWC. When the NWC receives it, it commands each switch to block the attack traffic. In this way, the attack traffic is blocked at the switches that the attack traffic comes through from the IoT-GWs, and the excess of the capacity in the IoT backhaul is prevented. After the attack traffic gets blocked, the NWC commands the switches to revert the priorities of the remaining traffic (i.e., the normal traffic) to the original value. This process is autonomously conducted immediately after the DDoS protector detects it.

We give additional explanations about the attack estimation and suppression in the QAS. In most cases, the self-propagation characteristic of Mirai immediately expands the infection within an IoT-GW, so we do not consider the mixture of infected and non-infected devices within an IoT-GW (i.e., within a VID). Therefore, the attack estimation and suppression performed for each VID in our system is considered to be suitable. Moreover, the NWC in the section II has to carefully estimate the traffic by investigating multiple parameters of the traffic because the suspected traffic diverted to the dedicated network does not reach the IoT server and the IoT services are temporarily interrupted. On the other hand, in our system, both the suspected traffic and the non-suspected traffic reaches the IoT server even after the QAS are executed. Therefore, the NWC can quickly judge which traffic should be suspected by only one parameter. The transmission rate is the most appropriate parameter for the quick judgement since it changes soon after the attack traffic occurs. Especially in IoT networks, the rate of DDoS attack traffic instantaneously increases when all of the infected IoT devices simultaneously start sending attack traffic. Although low-rate attack traffic is not estimated as attack traffic by the QAS, it is accurately detected by the DDoS protector afterwards. Of course some low-rate attack traffic can waste the resources in the IoT server until it gets suppressed, but it does not overflow the IoT backhaul. Thus, it does not have to be suppressed quickly in the QAS. Even if the QAS unintentionally estimates high-rate normal traffic as attack traffic, the QAS just minimizes the priority of the normal traffic and the normal traffic can keep reaching the IoT server. We set the threshold value to the average rate of normal traffic plus a margin. The threshold rate in our system makes it possible to quickly judge whether an attack has been launched simply from the rate of traffic. Note that there are no limits in the number of VIDs and the number of priorities in our system. In this way, our system achieves the novelty described in the section I.

Instead of the priority control, shaping or policing can suppress the suspected traffic and reduce the normal traffic from being discarded. In this case, only a partial suspected traffic reaches the IoT server, and the DDoS protector cannot precisely judge whether the traffic is attack or not since the rate of the suspicious traffic already gets suppressed. On the other hand, the QAS in our system enables the DDoS protector to detect the attack traffic because the QAS hardly discards the suspected traffic.

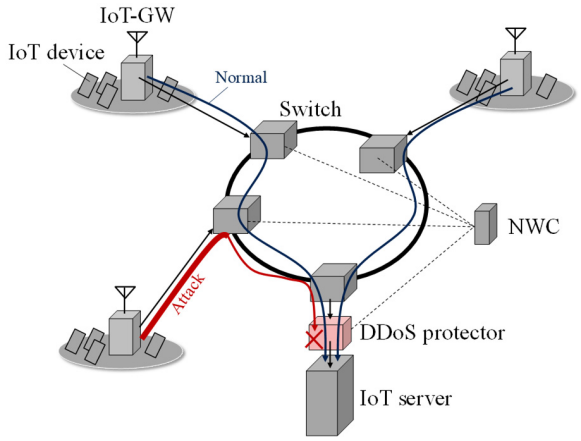


FIGURE 2. Topology of IoT network with our proposed system.

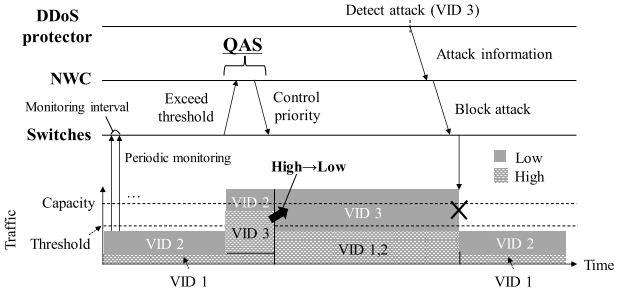


FIGURE 3. Time chart of our proposed system.

IV. EVALUATIONS

We performed two evaluations using commercial switches and a commercial DDoS protector to investigate the applicability of our system to the existing network. We chose the commercial DDoS protector which implemented an attack traffic detection algorithm based on machine learning and send detected attack information outside of it. The DDoS protector uses multiple parameters to detect attack traffic. We cannot know the details of the parameters because they are in black boxes, but what kinds of parameters the DDoS protector uses is out of our proposal. Although the switches and the DDoS protector came from different vendors, we achieved a coordinated autonomous operation of the proposed system by using application programming interfaces (APIs), which were embedded in our developed NWC. For a simple configuration, two traffic priorities were used: priority 2 was higher than priority 1. Both attack traffic and normal traffic had priority 2 when they were input to the IoT backhaul in order to confirm that the priority of attack traffic got minimized and the discarding of normal traffic got reduced by QAS.

A. EVALUATIONS WITH ATTACK TRAFFIC GENERATED BY A TRAFFIC GENERATOR

To evaluate the basic performance of the QAS, we generated the attack traffic and the normal traffic by a traffic generator. Fig. 4 shows the experimental setup. The switches, the NWC, the DDoS protector, and the IoT server were connected by a 1 Gigabit Ethernet. The traffic generator and switch #1 were connected by two 1 Gigabit Ethernet cables. The traffic generator and switch #2 were connected by a 1 Gigabit Ethernet cable. The IoT backhaul was composed of three switches. The attack traffic was input to the switch #1. The normal traffic was input to the switch #2.

1) THE LENGTH OF TIME IN WHICH NORMAL TRAFFIC WAS DISCARDED WITH RESPECT TO VARIOUS RATES OF ATTACK TRAFFIC

First, we evaluated the length of the time in which the normal traffic was discarded after the attack traffic started to be transmitted through the IoT backhaul ("discarding time") with respect to various rates of attack traffic. The attack traffic had two or four VIDs, each of which was the same rate. When the attack traffic had two VIDs, each 1 Gigabit Ethernet cable between the traffic generator and the switch #1 transmitted the attack traffic with a VID. When the attack traffic had four VIDs, each 1 Gigabit Ethernet cable between the traffic generator and the switch #1 transmitted the attack traffic with two VIDs. Each frame in the attack traffic was 1,000 Byte. The normal traffic had two VIDs, each of which was 0.3 Gbps (i.e., the total rate was 0.6 Gbps). Therefore, the available bandwidth of 0.6 Gbps was needed in order to prevent the discarding of normal traffic. Each frame in the normal traffic was 1,500 Byte. Each switch monitored the rate of traffic at every VID at intervals of 10 milliseconds.

Fig. 5 shows the discarding time with respect to the attack traffic on the horizontal axis (denoted as R_{att}). Overall, the discarding time increased as R_{att} increased. The discarding time was the same when R_{att} was higher than 1.6 Gbps in the case of the attack traffic with two VIDs, and higher than 1.8 Gbps in the case of the attack traffic with four VIDs. Also, the discarding time increased as the number of VIDs in the attack traffic increased at the same R_{att}.

The above results were caused by the following reasons. Generally speaking, when the maximum capacity in the IoT backhaul is exceeded after the attack occurs, some of the normal traffic is discarded. In order to reduce the discarding, the NWC in our system minimized the priority of the attack traffic in the QAS (if there was some normal traffic with the lowest priority, the NWC raised its priority before that). The NWC executed these priority controls VID by VID. The discarding of the normal traffic was prevented when enough bandwidth became available for the normal traffic as the result of a priority minimization of the attack traffic. Therefore, the discarding time depended on the number of VIDs whose priority should be minimized, which increased as the rate of attack traffic and the number of VIDs in the attack traffic increased. For example, in the case of the attack traffic with two VIDs, when R_{att} was 0.7 Gbps, the maximum capacity of 1 Gbps in the IoT backhaul was exceeded by the attack traffic of 0.7 Gbps and the normal traffic of 0.6 Gbps, both of which had priority 2. After the priority of one VID in the attack traffic got minimized, the traffic with priority 2

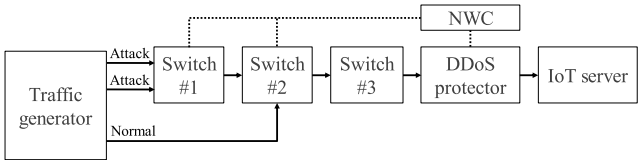


FIGURE 4. Experimental setup in section IV.A.

was the attack traffic of 0.35 Gbps and the normal traffic of 0.6 Gbps, which was less than the capacity in the IoT backhaul. Therefore, the discarding of the normal traffic was prevented. The discarding time was about 1.14 s. On the other hand, when R_{att} was 1.6, 1.8, or 2 Gbps, the priority of two VIDs in the attack traffic should be minimized in order to prevent the discarding of the normal traffic. That added about 0.5 s to the discarding time, in which the NWC minimized the priority of one VID. In the case of the attack traffic with four VIDs, when R_{att} was 1.4 or 1.6 Gbps, the discarding time was about 2.16 s. When R_{att} was 1.8 or 2 Gbps, the discarding time was about 2.66 s. As in the case of the attack traffic with two VIDs, the discarding time was changed by the number of VIDs whose priority got minimized before the discarding of the normal traffic was prevented.

In order to validate the experimental results, we theoretically calculated the number of VIDs whose priority should be minimized for preventing the normal traffic from being discarded, which is expressed by N_{pre}, as shown in Table 1. N_{pre} should satisfy (1):

$$C \geq R_{\mathrm{nor}} + R_{\mathrm{att}} \frac{N_{\mathrm{att}} - N_{\mathrm{pre}}}{N_{\mathrm{att}}} \tag{1}$$

where, C [Gbps] is the maximum capacity in the IoT backhaul, R_{nor} [Gbps] is the rate of the normal traffic, R_{att} [Gbps] is the rate of the attack traffic, and N_{att} is the number of VIDs in the attack traffic. Equation (2) was acquired by solving (1) for N_{pre}.

$$N_{\mathrm{pre}} = \left\lceil N_{\mathrm{att}} \left(1 - \frac{C - R_{\mathrm{nor}}}{R_{\mathrm{att}}} \right) \right\rceil \tag{2}$$

The experimental results in Fig. 5 were in agreement with the theoretical results in Table 1. In the case of the attack traffic with two VIDs, N_{pre} increased when R_{att} exceeded 0.8 Gbps according to Table 1 (a). This coincided with the experimental results in which the discarding time with R_{att} of 0.7 Gbps was smaller than that with R_{att} of 1.6 Gbps or larger. Similarly, in the case of the attack traffic with four VIDs, N_{pre} increased when R_{att} exceeded 1.6 Gbps according to Table 1 (b). This coincided with the experimental results in which the discarding times with R_{att} of 1.4 or 1.6 Gbps was smaller than those with R_{att} of 1.8 or 2 Gbps.

2) THE LENGTH OF TIME IN WHICH NORMAL TRAFFIC WAS DISCARDED WITH RESPECT TO VARIOUS INTERVALS OF TRAFFIC MONITORING

Second, we evaluated the discarding time with respect to various intervals of traffic monitoring (“monitoring intervals”).

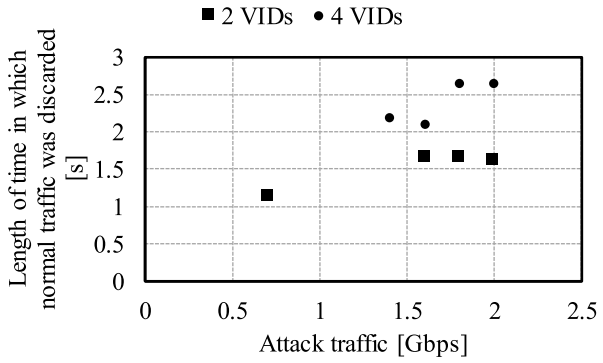


FIGURE 5. The length of time in which normal traffic was discarded with respect to various rates of attack traffic.

TABLE 1. The number of VIDs of attack traffic whose priority should be minimized.

(a) When attack traffic has two VIDs

Range of Ratt [Gbps]	Npre
$0 < R_{\mathrm{att}} \leq 0.4$	0
$0.4 < R_{\mathrm{att}} \leq 0.8$	1
$0.8 < R_{\mathrm{att}}$	2

(b) When attack traffic has four VIDs

Range of Ratt [Gbps]	Npre
$0 < R_{\mathrm{att}} \leq 0.4$	0
$0.4 < R_{\mathrm{att}} \leq 0.533$	1
$0.533 < R_{\mathrm{att}} \leq 0.8$	2
$0.8 < R_{\mathrm{att}} \leq 1.6$	3
$1.6 < R_{\mathrm{att}}$	4

The monitoring interval is the interval in which the switches monitor the rate of incoming traffic as shown in Fig. 3. The discarding time decreases as the monitoring interval decreases because the switches detect the excesses of traffic rate more quickly, but the decrease in the monitoring interval makes it difficult to distinguish the attack traffic and the normal traffic with burst inputs. Especially in the IoT backhaul, micro-burst traffic [12] occurs due to the simultaneous data transmissions from multiple IoT devices, and the instantaneous increase in the transmission rate of the normal traffic is observed. When the monitoring interval is long enough, the micro-burst traffic can be distinguished from the attack traffic because the average rate of the micro-burst traffic during the monitoring interval is smaller than the threshold rate.

In the experimental setup, the attack traffic had two VIDs, each of which was 0.8 Gbps (i.e., the total rate was 1.6 Gbps). Each frame in the attack traffic was 1,000 Byte. The normal traffic also had two VIDs, each of which was 0.05 Gbps (i.e., the total rate was 0.1 Gbps). Each frame in the normal traffic was 1,500 Byte. Fig. 6 shows the discarding time with respect to the various monitoring intervals. The discarding time increased as the monitoring interval increased. As the switches we used in this experiment told the NWC which VIDs had rates of traffic that exceeded the threshold rate after 64 intervals from the time the switches detected the excesses, the discarding time increased by about 64 times the increase in the monitoring interval.

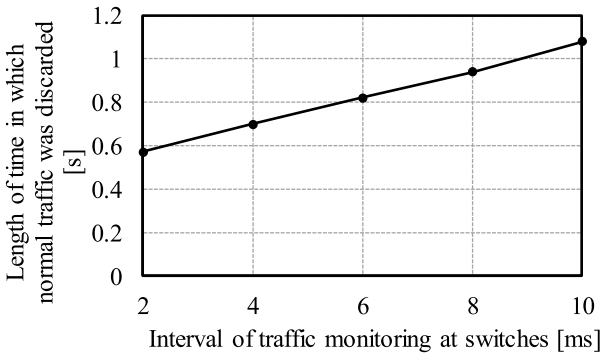


FIGURE 6. The length of time in which normal traffic was discarded with respect to various intervals of traffic monitoring.

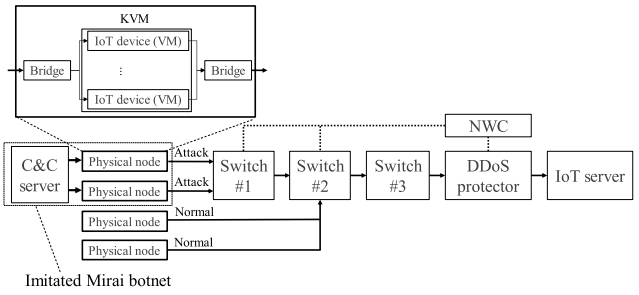


FIGURE 7. Experimental setup in section IV.B.

In addition, we considered the case where the normal traffic was a burst traffic. We assumed that each of 100 IoT devices sent a 1,500-Byte frame in a burst. Under this assumption, the length of a burst was 1.2 ms. In order to distinguish the normal burst traffic from the attack traffic, we set the monitoring interval to 6 ms, which was five times the length of a burst; the estimated rate was reduced to 0.2 Gbps while the instantaneous rate was 1 Gbps in 1.2 ms. According to Fig. 6, the discarding time was about 0.8 ms when the monitoring interval was 6 ms. In this way, the QAS could distinguish the normal burst traffic from the attack traffic as well as prevent the normal micro-burst traffic from being discarded within about a second.

B. EVALUATIONS WITH THE CONSTRUCTION OF MIRAI-BASED ATTACK TRAFFIC

To evaluate our system against real DDoS attacks, we constructed an imitation environment infected with Mirai in StarBED [13], which is a large-scale experimental environment in Japan operated by the National Institute of Information and Communications Technology (NICT). Mirai enables a malicious user to command infected IoT devices through a command and control server (C&C server) to send traffic. The network composed of the infected IoT devices and the C&C server is called a botnet. To conduct DDoS attacks, the malicious user commands all the infected IoT devices in the botnet through the C&C server to simultaneously send attack traffic with a high rate to the IoT server. The experimental setup that we constructed in StarBED is shown in Fig. 7. The traffic generator in Fig. 4 was replaced by physical nodes, on which 450 IoT devices were virtually built using a kernel-based virtual machine (KVM). Some IoT devices were imitated IoT devices in the botnet and sent attack traffic to the IoT server on the basis of commands from the C&C server. Other IoT devices sent normal traffic. The attack traffic had two VIDs. When the attack traffic was input to switch #1, it with each VID was transmitted via different 1 Gigabit Ethernet cable. The normal traffic also had two VIDs and input to switch #2. The attack traffic was ICMP destination unreachable flood. Each switch monitored the rate of traffic at every VID at intervals of 10 milliseconds. We isolated our experimental network logically from other experimental networks, and physically from the StarBED control network that connects with all experimental networks, so that we succeeded to prevent attack traffic from going out to other networks during our experiments.

1) EVALUATION RESULTS OF THE QAS

We evaluated the QAS in this section. 250 IoT devices sent the attack traffic whose rate was 1 Gbps (0.5 Gbps in each VID). 200 IoT devices sent the normal traffic, whose rate was 0.2 Gbps (0.1 Gbps in each VID). Fig. 8 and 9 show the screenshots in which Wireshark [14] captured the traffic input from the IoT devices to the IoT backhaul. The attack traffic in Fig. 8 and the normal traffic in Fig. 9 were sent to the same physical port in the IoT server, though they had different destination IP addresses. We confirmed that the attack traffic in Fig. 8 was a Mirai-based DDoS attack in that it was sent at shorter intervals than the normal traffic in Fig. 9 and all frames in it was ICMP destination unreachable.

Fig. 10 shows the experimental results of the QAS. When the DDoS protectors were installed in front of each switch (as described in section II), the normal traffic continued to be discarded. We confirmed that the discarding continued for about 30 s until the DDoS protector detected and blocked the attack traffic. In contrast, the QAS in our system prevented the normal traffic from being discarded within 30 ms. Our system prevented the discarding of normal traffic against the imitated Mirai-based attack traffic more quickly than against the attack traffic generated by the traffic generator in section IV.A. The Mirai-based attack traffic in this section gradually increased its rate because each IoT device in the botnet began to send traffic after it finished the ARP resolution with the IoT server, whereas the attack traffic generated by the traffic generator in section IV.A instantaneously increased its rate. In section IV.A, the normal traffic was discarded immediately after the rate of attack traffic exceeded the threshold rate. On the other hand, in this section, the gradual increase in the rate of the attack traffic provided some more time interval between the time when the rate of the attack traffic exceeded the threshold rate and the time when the normal traffic began

Time	Source	Destination	Protocol	Info
0.000000000	172.20.55.122	172.20.53.1	ICMP	Destination unreachable (Network unreachable)
0.000000000	172.20.55.15	172.20.53.1	ICMP	Destination unreachable (Network unreachable)
0.000002048	172.20.56.106	172.20.53.1	ICMP	Destination unreachable (Network unreachable)
0.000002048	172.20.55.85	172.20.53.1	ICMP	Destination unreachable (Network unreachable)
0.000002048	172.20.56.103	172.20.53.1	ICMP	Destination unreachable (Network unreachable)

FIGURE 8. Input attack traffic.

Time	Source	Destination	Protocol
0.000000000	172.20.32.9	172.20.53.3	UDP
0.000015770	172.20.31.3	172.20.53.2	UDP
0.000133480	172.20.31.8	172.20.53.2	UDP
0.000169920	172.20.32.86	172.20.53.3	UDP
0.000251920	172.20.31.98	172.20.53.2	UDP

FIGURE 9. Input normal traffic.

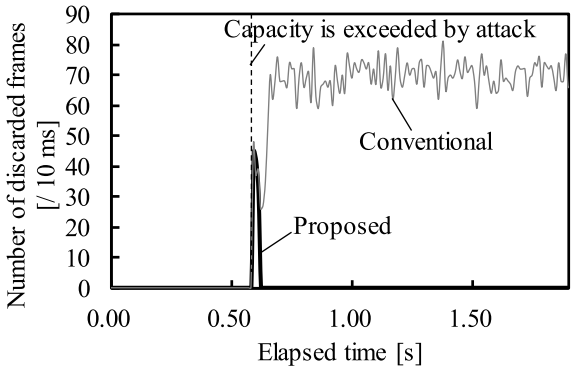


FIGURE 10. Results of quick attack suppression in our system.

to be discarded. Therefore, we confirmed that the discarding time was shorter in the actual attack environment.

2) EVALUATION RESULTS OF ATTACK BLOCKING

In the previous section, we evaluated the QAS. In this section, we evaluated the attack blocking method in our system. We show the rates of the traffic transferred through the IoT backhaul in Fig. 11. 100 IoT devices sent attack traffic, and 10 IoT devices sent normal traffic. First, the normal traffic of 100 Mbps began to be input at 0 s. After about 10 s, the attack traffic began to be input and flooded the IoT backhaul. When the DDoS protectors were installed in front of each switch (as described in section II), the DDoS protector detected and blocked all attack traffic after about 20 s from the time when the attack traffic began flooding the IoT backhaul. After blocking the attack traffic, only normal traffic was transmitted through the IoT backhaul. Our system similarly detected the attack traffic in about 20 s in the DDoS protector installed in front of the IoT server, however, it took about 365 s (from about 30 s to about 395 s on the horizontal axis in Fig. 11) to finish the blocking since it began sending the NWC the source IP addresses in the attack traffic one by one; the rate of traffic in the IoT backhaul was gradually reduced to the rate of the normal traffic. This was caused by the specification in the DDoS protector we used. Although the specification in the DDoS protector elongated the processing time to block all the attack traffic in our system in this experiment, the normal

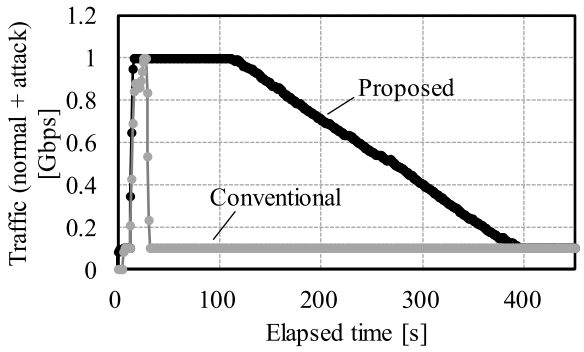


FIGURE 11. Rate of traffic in the IoT backhaul when attack blocking was executed.

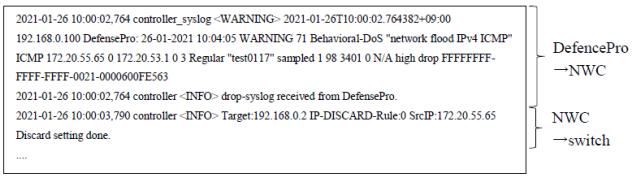


FIGURE 12. Logs in NWC when it received attack information and commanded switches to block attack.

traffic was prevented from being discarded in advance by our QAS as shown in section IV.A. Furthermore, the processing time to block all attack traffic in our system can be shortened by enabling the DDoS protector to send the NWC the source IP addresses in the attack traffic at a shorter interval. Although some attack traffic was discarded due to the priority control in the QAS, our system could detect all the attack traffic in the DDoS protector and block it at the switches.

In order to confirm our attack block mechanism integrating various vendors' products, we obtained an example log in the NWC as shown in Fig. 12. The DDoS protector ("DefencePro [15]") detected the IP address of 172.20.55.65 as an attack and sent a Syslog to the NWC. The NWC then blocked the traffic from this IP address when it attempted to access the switch #1 with the IP address of 192.168.0.2. In this way, all of the IP addresses of attack traffic were autonomously blocked.

V. CONCLUSION

In this paper, we proposed and experimentally evaluated a DDoS attack traffic suppression system that significantly reduces the discarding of normal traffic with a small number of equipment by controlling the priority of frames. In basic evaluations using a traffic generator, we confirmed our system prevented the discarding of the normal traffic in a few seconds. For more reliable evaluations, we constructed Mirai-based DDoS attacks, and experimentally confirmed that our system prevented the normal traffic being discarded in 30 ms. Furthermore, we also found that attack traffic detected by the DDoS protector was autonomously blocked at switches by integrating various vendors' products.

REFERENCES

[1] C. I. Nwakanma, A. P. Anantha, F. B. Islam, J.-M. Lee, and D.-S. Kim, "3GPP release-16 for industrial Internet of Things and mission critical communications," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 403–406.

[2] N. Shibata, P. Zhu, K. Nishimura, Y. Yoshida, K. Hayashi, and M. Hirota, "First demonstration of autonomous TSN-based beyond-best-effort networking for 5G NR fronthauls and 1,000+ massive IoT traffic," in *Proc. ECOC*, Dec. 2020, pp. 1–4.

[3] 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.

[4] Z. Liu, "CPSS LR-DDoS detection and defense in edge computing utilizing DCNN Q-learning," *IEEE Access*, vol. 8, pp. 42120–42130, 2020.

[5] K. Bhushan and B. B. Gupta, "Detecting DDoS attack using software defined network (SDN) in cloud computing environment," in *Proc. 5th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2018, pp. 872–877.

[6] B. Rashidi and C. Fong, "CoFence: A collaborative DDoS defence using network function virtualization," in *Proc. CNSM*, Nov. 2016, pp. 160–166.

[7] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, and A. Fernandez, "A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 53972–53983, 2020.

[8] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 29–35.

[9] N. Ravi and S. M. Shalinie, "Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3559–3570, Apr. 2020.

[10] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2017, pp. 1–8.

[11] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.

[12] K. Honda, N. Shibata, R. Harada, and S. Kaneko, "Cooperated traffic shaping technique for efficient accommodation of microbursts in IoT backhaul network," *IEICE Commun. Exp.*, vol. 10, no. 6, pp. 307–312, Feb. 2021.

[13] T. Miyachi, T. Nakagawa, K. Chinen, and S. Miwa, "StarBED and SpringOS architectures and their performance," in *Proc. 7th Int. ICST Conf.*, 2011, pp. 43–58.

[14] *Wireshark*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.wireshark.org/

[15] *Radware DefencePro*. Accessed: Aug. 20, 2021. [Online]. Available: https://www.radware.com/products/defensepro/

RINTARO HARADA received the B.E. degree in computer science and engineering and the M.E. degree in computer science and communications engineering from Waseda University, Tokyo, Japan, in 2015 and 2017, respectively. In 2017, he joined NTT Access Network Service Systems Laboratories, NTT Corporation, where he has been engaged in research on optical access networks. He is a member of IEICE. He received the Young Researcher's Award and the Encouraging Award from IEICE, Japan, in 2020 and 2021, respectively.

NAOTAKA SHIBATA received the B.E. degree in electrical and electronic engineering and the M.E. degree in communications and computer engineering from Kyoto University, Kyoto, Japan, in 2007 and 2009, respectively. In 2009, he joined the NTT Access Network Service Systems Laboratories, where he was engaged in research on wireless communication systems. Since 2012, he has been engaged in research on optical-wireless converged networks. He is a member of IEICE. He received the Young Engineer Award from the Institute of Electronics, Information, and Communication Engineers (IEICE), Japan, in 2015.

SHIN KANEKO received the B.E. and M.E. degrees in electronics engineering from The University of Tokyo, Tokyo, Japan, in 2002 and 2004, respectively. In 2004, he joined the NTT Access Network Service Systems Laboratories, Chiba, Japan. His current research interests include next-generation optical access networks and systems. He is a member of IEICE.

KAZUAKI HONDA received the B.S. degree from Kyoto University, Kyoto, Japan, in 2013. In 2013, he joined the NTT Access Network Service Systems Laboratories, NTT Corporation, Kanagawa, Japan, where he has been involved in research on IoT aggregating network and WDM-PON systems. He is a member of IEICE.

JUN TERADA received the B.E. degree in science and engineering and the M.E. degree in computer science from Keio University, Yokohama, Japan, in 1993 and 1995, respectively. In 1995, he joined the NTT LSI Laboratories, where he was involved in the research and development of low-voltage analog circuits, especially A/D and D/A converters. In 1999, he was involved in developing small and low-power wireless systems for sensor networks. In 2006, he was involved in researching high-speed front-end circuits for optical transceivers. He is currently a Senior Research Engineer and a Supervisor with the NTT Access Network Service Systems Laboratories, Yokosuka, Japan, where he is responsible for research and development management for optical and wireless converged access networks. He is a member of IEICE. He has served as a member for the Technical Program Committee of the Symposium on VLSI Circuits. He has been participating in the Asian Solid-State Circuits Conference (A-SSCC), since 2012.

YOTA ISHIDA received the M.E. degree in mechanical engineering from Kanazawa University, Ishikawa, Japan, in 2014. In 2014, he joined LIFULL Company Ltd., where he worked on real estate information services. In 2020, he joined the National Institute of Information and Communications Technology (NICT), Japan, where he worked on network emulation technology.

KUNIO AKASHI received the M.S. and Ph.D. degrees in information science from the Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, in 2010 and 2017, respectively. From 2017 to 2021, he joined the National Institute of Information and Communications Technology (NICT), Japan, where he worked on network emulation technology. In 2021, he joined The University of Tokyo, Japan, as an Assistant Professor. His current research interests include computer networks and network security.

TOSHIYUKI MIYACHI received the M.S. and Ph.D. degrees in information science from the Japan Advanced Institute of Science and Technology, Ishikawa, Japan, in 2002 and 2007, respectively. In 2007, he joined the National Institute of Information and Communications Technology (NICT), Japan, where he has been engaged in research on evaluation of network technologies especially on actual node-based network testbed. He is currently the Director of the Hokuriku StarBED Technology Center, NICT.