Is low-rate distributed denial of service a great threat to the Internet?

Ming Chen | Jing Chen | Xianglin Wei | Bing Chen

1 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
2 The 63rd Research Institute, National University of Defense Technology, Nanjing, China

Correspondence
Xianglin Wei, The 63rd Research Institute, National University of Defense Technology, Nanjing 210007, China.
Email: wei_xianglin@163.com

Abstract
Low-rate Distributed Denial of Service (LDDoS) attacks, in which the attackers send packets to a victim at a sufficiently low rate to avoid being detected, are considered to be a subtype of DDoS attacks and a potential threat to Internet security. However, an overwhelming attack paradigm on the Internet has rarely been reported due to the harsh requirements for launching LDDoS attacks; therefore, most existing LDDoS attacks are constructed and evaluated through theoretical deduction and/or simulation tests. In this backdrop, the authors aim to figure out what the conditions for launching a successful LDDoS attack are, and how harmful an attack could be. They first analyse the characteristics of LDDoS attacks, and derive the conditions and parameters for initiating LDDoS attacks using a queuing model. Based on the analysis results, an LDDoS algorithm is presented. Then, an LDDoS validation prototype is built on a Network Function Virtualization network to validate the derived parameters and conditions. Finally, a series of experiments are conducted on the testbed, and the results show that a successful LDDoS attack could be achieved based on the derived algorithm; however, its attack effect only lasts for a short time compared with its DDoS counterparts.

1 | INTRODUCTION

Denial of Service (DoS) is one of the major cybersecurity threats facing the Internet and enterprise networks in the Internet of Things (IoT) era, where everything is interconnected to facilitate smart applications. A DoS attacker can degrade and even corrupt the network and/or the upper-layer service by sending a large number of packets to the victims, including bottleneck links [1], edge computing servers [2, 3], or cloud data centres [4–6]. Furthermore, Distributed Denial of Service (DDoS) attacks could further increase the impacts of DoS attacks through launching attacks from many different distributed geographical or logical locations. Therefore, launching DDoS attacks would consume a huge amount of resources since the attacker has to control and utilise a large number of devices. In contrast, it is not difficult to find the existence of a DDoS attack due to its inherent characteristics, including scattered attack sources, huge amount of attack traffic, and severe network damage [1].

To elude detection, Low-rate Distributed Denial of Service (LDDoS) attacks are devised to exploit the vulnerability of Transmission Control Protocol (TCP)'s Retransmission Time Out (RTO) scheme [1, 7, 8]. An attacker that controls multiple puppet machines can send a number of periodic short-pulse traffic over short periods or continuously emit attack packets at a constant low rate to a victim (e.g. a bottleneck link) to throttle legitimate TCP flows. The attack flows aggregate at the victim, and this would enforce the legitimate TCP connections bypassing the victim entering the unstable state of RTO or fast recovery repeatedly, and thus degrade or weaken service capabilities of the network for processing legitimate user's normal flows. LDDoS can easily fail traditional detection and defence methods targeting DDoS attacks, since the average rate of each attack flow is indistinguishable from legitimate traffic flows. Therefore, LDDoS attacks have attracted much attention from both academia and industry in recent years. However, few LDDoS attack events are reported on the Internet, although LDDoS is considered as an efficient attack paradigm in simulations. To reveal the reasons behind this paradox, the following issues deserve more research. First, the conditions of a successful LDDoS attack in given network settings are not clearly stated in existing works [5]. Second, a
quantitative analysis of LDDoS is still absent, although a few conceptual analysis results are available. This may greatly hinder the in-depth analysis of LDDoS attacks. Therefore, it is reasonable to doubt whether LDDoS is a great threat to the Internet, and how harmful it could be.

Herein, the authors firstly establish a quantitative analysis model for LDDoS attacks using a queuing model, and derive the conditions of launching a successful LDDoS attack; then, a testbed based on Network Function Virtualization (NFV) [9] networks is constructed exploiting their flexibility and customisability to validate the LDDoS model and analytic results; finally, a series of experiments are conducted on the testbed to show the effectiveness of the derived conditions and parameters. The contributions herein are threefold: first, the two conditions for launching a successful LDDoS attack are derived through establishing a queuing model; second, an algorithm for an attacker to initiate an LDDoS attack is presented based on the derived conditions; and third, the feasibility of launching attacks utilising the presented algorithm is verified on a constructed NFV testbed with diverse parameter settings.

Section 2 summarises related work. Section 3 establishes a queuing model for analysing LDDoS attacks and derives the requirements for a successful LDDoS attack. Section 4 introduces the NFV-based testbed for testing LDDoS attacks, and validates the conclusions derived by the model through analysing experimental results. Section 5 summarises the full text and discusses future work.

2 | RELATED WORK

The danger of DDoS attacks on the Internet has led to many kinds of precautions against it. In recent years, DDoS attack technology has evolved greatly: on the one hand, the attack method is increasingly concealed, making it more difficult to detect; on the other hand, newly exposed network protocol design flaws are exploited by DDoS attacks. In 2001, Luo et al. first discovered the existence of low-rate denial of service (LDoS) attacks, called Plusing DoS (PDoS), on the Abilene backbone [10]. Researchers have discovered that this new DoS attack paradigm with pulse characteristics may last for a long time with high concealment. However, this type of attack does not necessarily paralyse the target system, but it can degrade the system performance. Since then, the working principle and detection/defence methods for LDoS attacks have gradually become a new topic in the field of network security.

The formulation and impact of LDoS or LDDoS attacks on the network have been investigated through simulation and testbed experiments. Kuzmanovic et al. first revealed the LDoS attack against the TCP protocol [8]. Guirguis et al. presented another concept, named the Reduction of Quality (RoQ) attack, using congestion control of the TCP protocol and the loopholes of the router queue management mechanism to reduce the performance of the router [11]. Kieu et al. analysed the throughput of a single TCP flow under LDDoS attack [12]. A comprehensive survey of LDDoS attacks could be found in reference [1].

The detection and defence of LDDoS attacks have attracted most attention in this area. Traditionally, practitioners detected DDoS attacks based on their long-term and high-intensity statistical characteristics. In contrast, LDDoS attacks do not have such characteristics and could fail traditional DDoS detection methods easily. Siracusano et al. presented a methodology for the detection of LDDoS attacks based on characteristics of malicious TCP flows [13]. Li et al. considered the LDDoS attacks in the container-based cloud environment [14]. Sahoo et al. presented a DDoS attack detection method in software-defined networking (SDN)-based data centre networks using information distance metrics [15]. Chen et al. introduced power spectrum entropy-based detection and mitigation of LDoS attacks [16]. Ren et al. proposed a method for detecting LDDoS attack traffic using a linear multiple regression model with Simple Network Management Protocol contents [17]. Lin et al. proposed a fair robust random early detection (FRRED) algorithm, a TCP-friendly AQM algorithm to improve the performance in terms of throughput and fairness [18]. Wu et al. designed a cross-correlation-based LDDoS attack time synchronisation and stream aggregation method [19]. Chen proposed the Fourier-Robust RED (FRRED) algorithm to detect LDoS attacks by implementing an active queue management system using Power Spectral Density (PSD) entropy [20]. Zhang et al. presented a detection algorithm that combines the PSD entropy function and support vector machine to distinguish LDoS traffic from normal traffic [21]. A factorisation machine-based LDDoS detection algorithm is put forward in reference [22]. Sahoo et al. presented a learning automata-based DDoS attack defence mechanism in software-defined networks [23]. Liu et al. introduced an intrusion detection and defence hybrid method for LDDoS attacks in an edge environment, which takes advantage of locality-sensitive features extraction and a Deep Convolution Neural Network to auto learn the optimal features of the original data distribution and employs a deep reinforcement learning Q-network as the powerful decision maker to defend attacks [2]. Pérez-Díaz et al. presented an SDN-based architecture to detect LDDoS attacks using machine learning techniques [24]. Liu et al. designed a LDDoS attacks detection method using data compression and behaviour divergence measurement [25]. Agrawal et al. introduced a LDDoS attack defence method based on power spectral density analysis [4].

From the above analysis, it is known that the current research into LDDoS attacks mainly focuses on theoretical analysis; however, there is a lack of analysis of its requirements in the actual Internet environment and the possible harm caused by LDDoS attacks. Table 1 compares this work with existing references.

3 | LDDoS ANALYSIS

This section first introduces LDDoS attacks; then, the conditions of launching LDDoS attacks are quantitatively analysed using a queuing model.
3.1 LDDoS attacks

LDoS is a pulsed attack that exploits TCP's congestion control mechanism to reduce its throughput; in contrast, LDDoS is essentially a distributed implementation of LDoS. As shown in Figure 1, an LDDoS attacker recruits some handlers to look for potential vulnerable victims and then controls a set of bots to send low-rate attack traffic to the victim. The aggregation of this traffic is usually embedded in legitimate network traffic and is difficult to detect using intrusion detection systems (IDSs). The aggregated attack traffic is large enough within a short time period to overwhelm the victim's limited buffer, causing a continuous timeout for the legitimate TCP flows.

LDDoS is mainly launched for two purposes. On the one hand, it could be adopted to throttle legitimate TCP flows. Through exploiting TCP's RTO mechanism to periodically send short attack flows to the bottleneck link in a network, an attacker can force the attacked TCP flows to continuously enter the RTO state and thus decrease their throughput. On the other hand, it can also be utilised to downgrade the performance of an application server, including edge server or cloud server [3, 5]. Specifically, an attacker enforces a service provided by the server to be busy for a certain period. Thereafter, although the network is not congested and the burden of the server is low, some users still feel that the server is always busy and cannot be accessed. The success of the two above-mentioned LDDoS attacks lies in the openness, predictability, and weakness of the protocol design. In other words, an attacker can achieve its attack goal in a smart way without seizing too many resources. LDDoS attacks have several characteristics, including low attack rate and high efficiency. Next, the requirements for launching a successful LDDoS attack by building a queuing model are discussed.

3.2 Condition analysis

In contrast with the high-intensity traffic in DDoS attacks, the rate of each LDDoS attack flow is low. However, to ensure the success of an LDDoS attack, two requirements should be fulfilled: on the one hand, the aggregated attack traffic can overwhelm a bottleneck link in the network; on the other hand,
each attack flow should be a pulse in nature to elude detection. Here, the focus here is on those LDDoS attacks targeting TCP flows. To ensure that the scattered low-rate traffic causes congestion at the victim and forms an LDDoS attack, the premises are: (1) convergence: low-rate traffic aggregates at the victim's buffer to cause a congestion at a short period; (2) synchronisation: the arrival time of converged attack flows is exactly the same as the period of the target TCP flows' RTO, while the rate of each attack flow can be very low in other periods.

It is assumed that the drop tail queue management algorithm is adopted to manage the output router's buffer of the bottleneck link with a pre-defined maximum queue length. Newly arrived packets will be added into the buffer as long as it is not full. Otherwise, the received data packets will be discarded. This tail-drop operation is independent of the types of arrived packets, for example TCP or User Datagram Protocol (UDP). Therefore, if the accumulated attack packets can make the output buffer full, other TCP or UDP packets that share the bottleneck link will be dropped. This will further cause the sender to reduce the sending buffer of impacted TCP flows before receiving any acknowledgement, and thus can decrease the throughput of the target TCP flows. If the buffer exhibits a congestion state whenever a TCP retransmission packet arrives at the bottleneck link, the victim TCP flow would stay in the suppressed state.

Figure 2 further depicts the change pattern of a victim's output buffer, and the output buffer size is $B$. Before the LDDoS attack, the buffer size is in a low and stable state (period $\tau$) before the predetermined arrival time of attack flows $s_i$. Low-rate attack flow $f_i$ arrive at the time $t_i$. When all attack flows arrive at the bottleneck and form a large attack traffic at period $\alpha$, the victim's output buffer will quickly become full at time $s_2$. This persistent low-rate traffic will make the queue saturated for a short period, that is the period $\xi$ from $s_2$ to $s_3$. Then, the buffered packets are delivered to the output port since the attack flows stop arriving, that is the period $\beta$ from $s_3$ to $s_4$. In this way, an attack period $T$ that is composed of time periods $\tau$, $\alpha$, $\xi$, and $\beta$ is formed, that is $T = \tau + \alpha + \xi + \beta$. Obviously, if $T$ can be synchronised with TRTO, that is the target TCP's RTO, and the saturation continues to coincide with the period in which the target TCP flow's retransmission packets arrive at the buffer, with the aggregated maximum pulse of all attack flows exceeds $B$, an attack towards the victim buffer and thus the TCP packets flow over the same buffer can be formed. Then, the arrival time of the $i$-th attack flow $f_i$ at the $k$-th attack period will be $t_i + k \times T$.

Therefore, the premise of forming an LDDoS attack can be formulated as:

$$
\begin{align*}
T & = T_{RTO} = \tau + \alpha + \xi + \beta \\
|t_i - s_i| & < \delta \\
M & = \sum_{i=1}^{n} f_i \times |s_2 - s_1| \geq B
\end{align*}
$$

(1)

For ease of description, Table 2 lists the main symbols adopted herein.
Based on the buffer queue shown in Figure 3, a $M^m/M/1/B$ queuing model can be established, which is described by using a first-in first-out queue at each output port, and allows up to $m$ data packets to reach the attacked bottleneck link in each time step, while at most one data packet can leave. Therefore, after the buffer queue is attacked by the LDDoS in a stable state, the attack traffic can reach the attacked bottleneck link periodically to keep the buffer queue full and cause congestion for the target TCP flow, forming an effective attack.

The assumptions adopted here are threefold: (1) all the attackers’ clocks are synchronised to ensure that all attack flows aggregate at the victim; (2) the controller of the LDDoS attack could estimate or measure the round-trip time (RTT) of a packet; and (3) the buffer size at the victim could be estimated.
A quad-tuple LDDoS \((T, L, b_i, n)\) can be used to represent an LDDoS attack, in which \(T\) is the attack period of the attack traffic, \(L\) represents the low-rate attack traffic width, \(b_i\) indicates the rate of attack traffic sent by each attacker, and \(n\) is the number of attack sources. Then, the total attack traffic is \(M = \sum_{i=1}^{n} b_i\), and \((M \geq B)\), and \(B\) is the size of the victim's output buffer.

Sending a one-time attack traffic cannot affect significantly the target TCP flows. In order to launch a powerful LDDoS attack, it is necessary to repeat the pulse attack within a certain period of time, so that the target TCP flows can be blocked for a long time. The characteristics of a single attack flow can be described by the attacker \(\text{Attacker}_i\) \((i = 1, 2, \ldots)\) on the left side of Figure 3. Each attack flow can be set with three parameters: rate \(b_i\), burst length \(t_b\), and burst period \(T\). \(b_i\) indicates the rate at which the attack data are sent, and the rate of converged traffic needs to be greater than the buffer size of the bottleneck link, so that the retransmission effort of the target TCP flows keeps failing due to continuous time out, causing the flow rate to drop sharply. The burst length \(t_b\) indicates the duration of the attack flow, which depends on the attacker's bandwidth, RTT, and transmission rate. It is difficult to form an attack if \(t_b\) is too small; in contrast, it is easy to be detected if it is too large. The burst period \(T\) regulates the interval between two attack bursts, and it should be synchronised with the RTO of target TCP flows.

Attack flows constituting an LDDoS attack need to be synchronised to ensure that they arrive at the victim's output buffer within a short time period. Synchronisation here requires: (1) the time of all attack flows reaching the attacked buffer satisfies \(|t_i - s| < \delta\), where \(t_i\) is the first arrival time when the \(i\)-th attack flow \(f_i\), and \(s\) is a predetermined time; and (2) the burst period \(T\) is the same as the RTO of the attacked TCP flows. In Section 4, how to enable all attack flows to achieve the above two synchronisation requirements in a testbed built on a distributed NFV network is validated.

### 3.4 Algorithm description

As mentioned above, all attackers have to collaborate with one another to launch an LDDoS attack at the victim. Therefore,
for an attacker, called the controller, that controls a botnet with a number of attackers, it needs to coordinate their attacking behaviours to ensure that all attacking pulses arrive at the victim in the desired way. In this section, an algorithm for the controller is designed in accordance with the derived LDDoS conditions.

Algorithm 1 illustrates the controller’s behaviour for launching an LDDoS attack, in which it controls n attack sources. In step 1, the controller infers the victim’s buffer size B; and it measures the RTT at the network using Ping in step 2; afterwards, it calculates the mean square error of its RTT estimation in step 3. Then, a TCP flow’s RTO is derived in step 4, and the attack period T is calculated based on RTO in step 5. After selecting n attackers from the botnet in step 6, the controller could determine each attacker’s attack traffic volume based on the number of attackers and the victim’s buffer size in step 7. Steps 8 and 9 determine and dispatch each attacker’s parameters. Finally, step 10 synchronises all selected attackers’ clocks. For each selected attacker, it follows the controller’s command, and sends packets to the dedicated victim using the dispatched parameters by the controller.

Algorithm 1 Attack behaviour for an LDDoS controller

Require:
\[ \alpha, \beta, \delta \]

Ensure: Selected attackers and their respective attack parameters

1: Infer the buffer size at the victim link;
2: Measure the RTT using Ping;
3: Calculate the mean square error of RTT estimation;
4: Calculate a TCP flow’s RTO;
5: Calculate the attack period T according to RTO;
6: Select n attackers from its botnet;
7: Determine each attacker’s attack traffic volume based on the number of attackers and the victim’s buffer size;
8: Determine the start time of each attacker and its attack period;
9: Dispatch the attack parameters to each selected attacker;
10: Synchronise the selected attackers’ clock.

4 | VALIDATION AND RESULTS ANALYSIS

In order to verify the correctness of the analytical results derived in Section 3, a testbed is built based on a NFV network due to its flexibility and easy-to-deploy features. Then, the experimental results are analysed, and the complexity of fulfilling the conditions for a successful LDDoS attack is discussed.

4.1 | Testbed construction

NFV is an architecture concept that uses virtualisation technologies to virtualise entire classes of network functions into building blocks that may connect, or chain together, to create transmission services. It has many advantages over traditional network paradigms, including increased elasticity in terms of scaling up and scaling down the network, increased service agility to support new faster service rollouts, and flexibility in parameter settings. It is suitable for constructing testbeds here for validating the above-mentioned analysis results.

The authors use the topology generator Brite [26] to generate a typical network topology consisting of 31 nodes that conforms to the power law distribution, and generate an NFV network prototype NET1 based on the Linux Container (LXC). The constructed network environment, called NET1, is shown in Figure 4. In Figure 4, a total of 31 nodes are included; n1–n23 are routers, and PC1–PC6, and Server are hosts. The IP addresses of each interface are labelled in the figure.

In order to make the network environment more realistic, D-ITG [27] is utilised to generate distributed background traffic with Poisson distribution, and Quagga [28] is adopted to set parameters, for example bridge construction and routing protocol to obtain an NFV experimental environment. Through network configuration and traffic experiments, NET1 aims to emulate the behaviours of the Internet, and has the characteristics of easy-to-control, flexibility, easy-to-observe etc., and is an ideal environment for LDDoS research.

4.1.1 | The configuration of NET1

There are six border routers n1–n6 and 18 internal ones; the bandwidth of each link is set to be 100 Mbps. Open Shortest Path First is adopted as the routing protocol. A Poisson flow is injected into this network every 5 s as the background flow, and each flow is composed of UDP packets with an intensity of 1 Mbps.

To make the test process concise, part of the NET1 network that involves the LDDoS attack experiment is redrawn in Figure 5. Ten senders, that is User1 to User6, connect to router n5; 10 receivers are Receiver1 to Receiver10; n UDP attackers are Attacker1 to Attacker10; one victim server connects to router n25; and users and receivers use TCP as the transportation layer protocol. The link between n9 and n25 is the bottleneck link, and its bandwidth and delay are 10 Mbps and 20 ms, respectively. The length of each output queue is 100 packets. The bandwidth and delay of remaining links are 10 Mbps and 20 ms, respectively.

The experiment lasts for 80 s. In the first 10 s, only legitimate and background TCP flows are transmitted in the network; the attack starts at the 10th second, and lasts until the end of the experiment. The authors set up the LDDoS sender
on each attacker to adjust the attack effect of the LDDoS by controlling the parameters of quad-tuple LDDoS \((T, L, b, n)\) as defined in Section 3.3. Wireshark is installed on Receiver\(_1\), Receiver\(_2\), and the attacked server, to measure and analyse the attack effect. Since all LXC\(_s\) are running on the same host server, their clocks are synchronised.

### 4.2 Experiments and results analysis

Based on the quad-tuple LDDoS \((T, L, b, n)\) defined in Section 3, a series of verification experiments are performed in the NFV network to evaluate the influence of parameters on the influence of LDDoS attacks.

#### 4.2.1 Experiment 1: Synchronisation of attack period \(T\) with RTO

One of the two important conditions for forming an LDDoS derived in Section 3 is that the attack period \(T\) is synchronised with the target TCP flows’ RTO. In the experiments, an LDDoS attack is formed by a large amount of periodic low-rate traffic, where the attack period \(T\) is an important factor.

---

**FIGURE 4** The topology of the constructed network NET1

**FIGURE 5** The topology of the Low-rate Distributed Denial of Service attack experiment
affecting the attack’s influence. This interval needs to be controlled during the experiments so that data packets sent by a TCP flow entering the timeout retransmission state are discarded when they arrive at the bottleneck router.

A TCP flow’s RTO is related to the RTT of the data packet, which is easy to measure using Ping. In Linux systems, the RTO calculation process is as follows: the estimated RTT of a packet is recorded as $R_p$, the mean square error of RTT estimation is recorded as $R_m$, the default minimum of RTO is $RTO_{MIN}$, and the first RTT measurement is recorded as $R_m$, then it can be formulated as:

$$R_i = R_m$$  \hspace{1cm} (2)

$$R_D = \frac{R_m}{2}$$  \hspace{1cm} (3)

$$T_{RTO} = R_i + \max\left\{ G, 4 \times \frac{R_D}{2} \right\}$$  \hspace{1cm} (4)

where $G$ denotes the clock granularity (typically $G \leq 100 \text{ ms}$). Assume the subsequent RTT measurement is $R'$, and the measurement host needs to recalculate the $T_{RTO}$ according to the updated $R_D$ and $R_m$, the update can be formulated as:

$$R_D = (1 - \alpha) \times R_D + \alpha \times |R_i - R'|$$  \hspace{1cm} (5)

$$R_i = (1 - \beta) \times R_i + \beta \times R', \beta = 0.125$$  \hspace{1cm} (6)

Then, $T_{RTO}$ can be calculated as:

$$T_{RTO} = R_i + \max\{RTO_{MIN}, 4 \times R_D\}$$  \hspace{1cm} (7)

Assume that $\min T_{RTO} > R_i + 4 \times R_D$; then, the LDDoS period $T$ must satisfy:

$$\min T_{RTO} < T \leq R_i + RTO_{MIN} + 4 \times R_D \hspace{1cm} (8)$$

The experiments are conducted on Linux systems, and this can mitigate the impacts of dramatically changing RTT on RTO and make the RTO trends smoother. The NET1 experimental environment is an intranet environment, and the actual RTT value is very small. In Equation (7), $T_{RTO} = R_i + RTO_{MIN}$, therefore the $T_{RTO}$ of the LXC can be changed and set by $RTO_{MIN}$. In the following experiments, the $T_{RTO}$ of an attacker is recorded as $T_{RTO} = 1 \text{ s}$.

Generally speaking, as the duration of the attack flow is extended, the attack influence is gradually weakened. If the duration of the LDDoS attack traffic is set to 100-250 ms, the actual network can neither identify the attack traffic nor avoid being filtered by multiple router buffers. A comparison is conducted of the attack traffic width in this range, showing that the aggregation of the attack flows and the attack effect are best when $L = 200 \text{ ms}$. Therefore, the low-rate attack traffic width $L$ is set to be 200 ms.

In Experiment 1, the LDDoS quad-tuple is set to be LDDoS = ($t$, 0.200, 2 Mbps, 5), that is the period $T = t$, and $t$ are set to be 0.8, 1, 1.5, and 2 s, respectively; $L = 200 \text{ ms}$, $b_I = 2 \text{ Mbps}$, and $n = 5$. The analysis results shows that since all attack flows are at the same distance from the attacker, and the same attack flow is sent simultaneously by multiple threads, thus the time that the attack flows reach the attacked buffer can satisfy the condition of $|t_i - s_1| < \delta$ in Section 3.3, $1 \leq i \leq 5$. The attack pulse is composed of multiple low-rate flows, Figure 6 shows the aggregation of attack flows with the same characteristics. It can be seen that the highest attack pulse is about 9 Mbps, which does not reach the expected 10 Mbps. This is due to the fact that the actual aggregation of the attack flows cannot be strictly controlled in the actual network, and there is a

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure6.png}
\caption{Aggregated attack flows}
\end{figure}
deviation within the normal range of the aggregation. The measurement results show that the link output buffer queue uses the management method in Section 3, and most legitimate TCP packets will be dropped when an LDDoS attack exists.

The change in the average cumulative throughput of legitimate user TCP flows before and after the attack is used as a reference, and the attack effect of LDDoS is observed. The measurement and analysis results are shown in Figure 7.

As can be seen from Figure 7, the LDDoS attacks could impact legitimate TCP flow's throughput. When the attack period is the same as the RTO of the target TCP flow, the attack's influence is maximised, that is when $T = 1 \text{s}$, the throughput of the bottleneck link can be reduced by more than half. Because its attack effect is periodic, when the attack flows converge within a period $T$, it will decrease the throughput of legitimate users rapidly. However, the throughput will rise quickly due to the short burst time, and the average throughput cannot be reduced to zero. When the attack period $T$ moves closer to $1 \text{s}$, the attack effect becomes more obvious. As the attack time increases, the network is in a continuous suppressed state. In this condition, the throughput for legitimate users also approaches a relatively stable value. Experimental results validate that the attack's impact is remarkable and stable only when the attack period is consistent with the target TCP flows' RTO.

### 4.2.2 | Experiment 2: Synchronous and asynchronous attack

Another key factor of forming an LDDoS attack is synchronisation, even if the times of all the attack traffic scattered around the network reach the attacked victim buffer satisfy $|t_i - s_j| < \delta$. Five attackers are set up in Figure 5 to perform synchronous and asynchronous attacks. In a synchronous scenario, all attackers have the same characteristic and distance to the victim, and send the same attack traffic at the same time.

In contrast, in an asynchronous scenario, all attackers also have the same distance to the victim, but each attacker starts sending attack traffic at different times, that is the attack traffic sent by each attacker arrives independently. By observing the impact of synchronous and asynchronous attacks on the network, the aggregation conditions of the attack flows mentioned in Section 3 can be studied.

In Experiment 2, the LDDoS quad-tuple is set to LDDoS $= (1 \text{s}, 200 \text{ ms}, 2 \text{ Mbps}, 5)$. With the same parameters, the synchronisation and asynchronous attack scenarios are set and compared with the normal scenario, where no LDDoS attack is launched. In the case of a synchronised attack, five attackers can simultaneously superimpose their respective attack traffic. In the case of an asynchronous attack, five attackers start to send attack traffic at the interval of 10 ms, that is $|t_i - s_j| = \delta \times (i - 1)$, $\delta = 10 \text{ ms}$. Although the same numbers of attack packets are sent, it cannot converge into a sufficiently large attack traffic in the case of an asynchronous attack. The results are shown in Figure 8.

As can be seen from Figure 8, the synchronous attack has a better attack effect than the asynchronous attack. This is because the synchronous attack flow can arrive at the same time and aggregate into the expected attack traffic peak, forming an LDDoS attack in the RTO period. Under the condition of an asynchronous attack, since the attack flows are not synchronised, it is difficult for them to aggregate into a large enough attack traffic peak in a certain period of time, and the effect of the LDDoS attack is relatively weak. In particular, if each attacker runs on a different host server and all attackers’ clocks are not synchronised, it will be very difficult to deploy an effective LDDoS attack. In other words, the attackers must first solve the clock synchronisation problem before launching an LDDoS attack, such as using Network Time Protocol (NTP). However, it is a non-trivial task to configure NTP on the puppet machines on the Internet.

### 4.2.3 | Experiment 3: Impact of attack parameters

When the attack flow rate $b_i$ of each attacker is too small, the flow is difficult to detect, but more attackers are needed to launch an attack. When the value of $b_i$ is too large, in contrast, each attack flow could be easily detected by the system and will be blocked by the IDS. In Experiment 3, it is assumed that the attack flow is detected with a threshold of 5 Mbps and LDDoS $= (1 \text{s}, 200 \text{ ms}, b_i \text{ Mbps}, 5)$, in which $b_i$ varies from
0 to 3 Mbps, and all attackers are synchronised. The experiment results are shown in Figure 9.

As can be seen from Figure 9, as the single attack flow rate $b_1$ increases, the impact of LDDoS on TCP flows increases. When $b_1$ is less than 2 Mbps, the attack effect is already obvious. This is because as the aggregated attack traffic peaks increase, the dropped TCP data packets will increase, so the aggregate throughput of legitimate TCP receivers is greatly reduced. When $b_1$ is greater than 2 Mbps, the throughput is reduced by more than half, and the impact of the attack effect on throughput is also gradual. This is because the aggregated total traffic exceeds the bottleneck link capacity during the attack pulse. In other words, further increasing the attack rate will not remarkably increase the attack's influence.

With the increase in the number of attackers, the difficulty of aggregating multiple attack flows increases and the peak flows of aggregation can be affected. The following experiments were set to study the effect of this parameter on the attack effect. Let LDDoS = (1 s, 200 ms, 2 Mbps, $n$), in which the number of attackers $n$ varies from 2 to 10. The results are shown in Figure 10.

As can be seen from Figure 10, the impact of LDDoS attacks on TCP throughput decreases from fast to flattened, with the decreasing rate reducing as $n$ increases. This is due to the fact that aggregated total traffic exceeds the bottleneck link capacity after $n = 5$. The attack resources required to form an LDDoS attack depend on the filtered traffic threshold and the buffer size. The lower the threshold and the higher the buffer size, the greater the required number of attackers. Therefore, the consumption of attack resources when deploying attackers needs to be carefully considered.

4.2.4 Experiment 4: Comparison of LDDoS and DDoS

The authors compared LDDoS = (1 s, 200 ms, 2 Mbps, 5) with the corresponding DDoS in the same experimental environment built by TFN2K [19].

Figure 11 shows the curve of legitimate TCP flows' throughput in both DDoS and LDDoS attack scenarios. Under the DDoS attack, the throughput of TCP drops rapidly to zero. Under the LDDoS attack, the total TCP flows' throughput still fluctuates around 4 Mbps. Obviously, LDDoS attacks on TCP connections are not as effective as DDoS, since LDDoS attacks are mainly determined by factors such as attack length $L$, attack period $T$, total attack traffic peak $M$, and attack synchronisation. However, DDoS can easily keep the bottleneck links congested by brute force. Therefore, in terms of attack effectiveness, LDDoS is significantly less durable and obvious than DDoS attacks. Although the concealment of LDDoS is better, its construction and implementation requirements are higher.
5 | DISCUSSION

Through experiments and results analysis, the following three conclusions can be drawn.

The conditions of launching an effective LDDoS attack are very strict, as derived in Section 3 and validated in this section, which shows the difficulty of launching an LDDoS attack on the Internet. First, each attacker needs clock synchronisation to ensure their parallel arrival at the victim's output buffer, which is very difficult to implement considering a large number of attackers' scattered locations. Second, synchronisation of the attack period $T$ with target TCP flow's RTO requires the attacker to know the RTO in advance. Actually, measuring and calculating a TCP flow's RTO on the Internet is a non-trivial task since it requires the attacker to monitor the network environment dynamically.

Compared with DDoS, LDDoS's influence on the network performance is lighter and less durable. Setting the parameters of the LDDoS quad-tuples (attack period $T$, attack length $L$, total attack traffic peak $M$, number of attackers $n$ etc.), appropriately can cause periodic pulse attacks and degrade legitimate flows' throughput, but cannot achieve a sustainable attack effect.

LDDoS has high concealment and the low-rate attack flows are embedded in legitimate flows, thus traditional detection methods cannot accurately detect the existence of the attack flows. It can usually cause attacks with limited impact on low-speed bottleneck links, and it is very hard to launch LDDoS attacks towards high-speed backbone networks on the Internet.

The above characteristics of LDDoS attacks can explain why no large-scale LDDoS event on the Internet has been reported so far.

6 | CONCLUSION

The feasibility and influences of launching LDDoS attacks on a testbed built on a NFV network according to current LDDoS theory has been verified herein. A few analysis results are derived about how to launch an LDDoS attack and several analytic results are obtained based on a queuing model. Then, a series of experiments are conducted to investigate the derived LDDoS characteristics, and the following conclusions are made: first, launching an effective LDDoS attack has strict conditions, which are difficult to be implemented on the Internet; second, LDDoS attacks' effects are lighter and less durable compared with the DDoS counterparts; third, LDDoS attacks have a concealed feature, and their attack flows can be easily embedded in legitimate flows and are difficult to be detected. So far, the authors believe that LDDoS is difficult to launch and detect on the Internet.

Next, the authors will investigate the impacts of LDDoS attacks on the upper-layer applications, such as edge computing servers etc.

REFERENCES

1. Wu, Z., et al.: Low-rate DoS attacks, detection, defence, and challenges: a survey. IEEE Access. 8, 43920–43943 (2020)
2. Liu, Z., Yin, X., Hu, Y.: CPSLR-LDDoS detection and defence in edge computing utilising DCNN-Q-learning. IEEE Access. 8, 42120–42130 (2020)
3. Wei, X., et al.: Wireless edge caching based on content similarity in dynamic environments. J. Syst. Architect. 115, 102000. http://www.sciencedirect.com/science/article/pii/S1387621121000163 (2021). https://doi.org/10.1016/j.sysarc.2021.102000
4. Agrawal, N., Tapaswi, S.: Low rate cloud DDoS attack defence method based on power spectral density analysis. Inf. Process. Lett. 138, 44–50 (2018). http://www.sciencedirect.com/science/article/pii/S0020019018301261
5. Cao, Y., et al.: A LDDoS-aware energy-efficient multipathing scheme for mobile cloud computing systems. IEEE Access. 5, 21862–21872 (2017)
6. Bawany, N.Z., Shamsi, J.A.: SEAL: SDN based secure and agile framework for protecting smart city applications from DDoS attacks. J. New. Comput. Appl. 145, 102381 (2019). http://www.sciencedirect.com/science/article/pii/S108480451930195X
7. Li, H., et al.: LAAEM: a method to enhance LDoS attack. IEEE Commun. Lett. 20(4), 708–711 (2016)
8. Kuzmanovic, A., Knightly, E.W.: Low-rate TCP-targeted denial of service attacks. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ser. SIGCOMM 03, pp. 75–86. Association for Computing Machinery, New York (2003). https://doi.org/10.1145/863955.863966
9. Miqumbi, R., et al.: Network function virtualisation: state-of-the-art and research challenges. IEEE Commun. Surv. Tutorials. 18(1), pp. 236–262. (2016)
10. Luo, X., Chang, R.: On a new class of pulsing denial-of-service attacks and the defense. In: Proceedings of the Network & Distributed System Security Symposium (NDSS), San Diego, California, USA, pp. 1–19. (2005)
11. Guirguis, M., Bestavros, A., Matta, I.: Exploiting the transients of adaptation for roq attacks on internet resources. In: Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP 2004), Berlin, Germany, pp. 184–195. (2004)
12. Kieu, M.V., Nguyen, D.T., Nguyen, T.T.: A way to estimate TCP throughput under low-rate DDoS attacks: one TCP flow. In: Proceedings of the 2020 RIVF International Conference on Computing and Communication Technologies, Ho Chi Minh city, Vietnam, pp. 1–8. (2020)
13. Siraeusamo, M., Shiaeles, S., Ghita, B.: Detection of LDDoS attacks based on TCP connection parameters. In: Proceedings of the 2018 Global Information Infrastructure and Networking Symposium (GIS), Thessaloniki, Greece, pp. 1–6. (2018)
14. Li, Z., et al.: Exploring new opportunities to defeat low-rate DDoS attack in container-based cloud environment. IEEE Trans. Parallel Distrib. Syst. 31(3), 695–706 (2020)
15. Sahoo, K.S., et al.: An early detection of low rate DDoS attack to SDN based data centre networks using information distance metrics. Future Gener. Comput. Syst. 89, 685–697 (2018). http://www.sciencedirect.com/science/article/pii/S0167739X18309543
16. Chen, Z., et al.: Power spectrum entropy based detection and mitigation of low-rate DoS attacks. Comput. Network. 136, 80–94 (2018). http://www.sciencedirect.com/science/article/pii/S1389128618301026
17. Ren, J., et al.: Smart NCAP supporting low-rate DDoS detection for ieee 802154-1-5 internet of things. In: Proceedings of the 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), Taiwan, China, pp. 532–535. (2019)
18. Lin, J., et al.: A TCP-friendly AQM algorithm to mitigate low-rate DDoS attacks. Int. J. Autonom. Adapt. Commun. Syst. 9, 149 (2016)

ORCID

Jing Chen https://orcid.org/0000-0003-2672-4587
19. Zhijun, W., et al.: Research on time synchronization and flow aggregation in LDDoS attack based on cross-correlation. In: Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, United Kingdom, pp. 25–32. (2012)

20. Chen, Z., et al.: FRRED: Fourier robust RED algorithm to detect and mitigate LDoS attacks. In: Proceedings of the 2017 Zooming Innovation in Consumer Electronics International Conference (ZINC), Novi Sad, Serbia, pp. 13–17. (2017)

21. Zhang, N., Jaafar, F., Malik, Y.: Low-rate DoS attack detection using PSD based entropy and machine learning. In: Proceedings of the 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, pp. 59–62. (2019)

22. Zhijun, W., et al.: Low-rate DDoS attack detection based on factorization machine in software defined network. IEEE Access 8, 17404–17418 (2020)

23. Sahoo, K.S., et al.: A learning automata-based DDoS attack defence mechanism in software defined networks. In: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, ser. MobiCom 18, pp. 795–797. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3241539.3267764

24. Pérez-Díaz, J.A., et al.: A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. IEEE Access. 8, 155859–155872 (2020)

25. Liu, X., et al.: Low-rate DDoS attacks detection method using data compression and behaviour divergence measurement. Comput. Secur. 100, 102107 (2021). http://www.sciencedirect.com/science/article/pii/S0167404820303801

26. Medina, A., et al.: BRITE: Universal Topology Generation from a User's Perspective. OpenBL, (2001). https://open.bu.edu/handle/2144/1624

27. Avallone, S., et al.: D-ITG distributed Internet traffic generator. In: First International Conference on the Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings., 2004, pp. 316–317. Enschede, Netherlands (2004)

28. Ishiguro, K.: Quagga Software Routing Suite. (2021). http://www.nongnu.org/quagga/. Accessed 01 May 2021

How to cite this article: Chen, M., et al.: Is low-rate distributed denial of service a great threat to the Internet? IET Inf. Secur. 15(5), 351–363 (2021). https://doi.org/10.1049/ise2.12031