IET Information Security

*Research Article*

IET Journals
The Institution of Engineering and Technology

# Detecting LDoS attack bursts based on queue distribution

ISSN 1751-8709
Received on 24th March 2018
Revised 24th December 2018
Accepted on 22nd January 2019
E-First on 20th February 2019
doi: 10.1049/iet-ifs.2018.5097
www.ietdl.org

*Meng Yue[1] ✉, Zhijun Wu[1], Jingjie Wang[1]*

*[1]School of Electronics, Information & Automation, Civil Aviation University of China, Tianjin, People's Republic of China*

✉ *E-mail: myue_23@163.com*

**Abstract:** Low-rate denial of service (LDoS) attacks exploit the congestion control mechanism to degrade the network quality of service. As a classic active queue management algorithm, random early detection (RED) algorithm is widely used to avoid network congestion. However, RED is vulnerable to LDoS attacks. LDoS attacks with well-configured attack parameters force RED queue to fluctuate severely, thereby throttling transmission control protocol (TCP) senders' sending rate. A feedback control model is proposed to describe the process of the congestion control, by which the congestion window and queue behaviours are analysed combined. After that, a two-dimensional queue distribution model composed of the instantaneous queue and the average queue is designed to extract the attack feature. Moreover then, a combination of a simple distance-based approach and an adaptive threshold algorithm is proposed to detect every LDoS attack burst. Test results of network simulator (NS)-2 simulation and test-bed experiments indicate that the proposed detection strategy can almost completely detect LDoS attack bursts and is especially robust to legitimate short bursts.

## 1 Introduction

Low-rate denial of service (LDoS) attack was first proposed in 2003 [1]. During the 15 year period, such attack has been developed into many variants such as reduction of quality (RoQ) [2], fraudulent resource consumption (FRC) [3, 4], slow attack [5], stealthy DoS [6], and tail [7]. In general, LDoS attack has three properties: (i) it exploits the vulnerability of a specific protocol or system in networks to inflict significant degradation in some aspects of the service such as resource utilisation, system stability, or service quality. (ii) The cost is low. A single attack source can launch attacks, and its attack traffic is much smaller than the flooding DoS (FDoS) attack. (iii) The average attack rate is very low (even lower than legitimate traffic), so it has strong concealment, which makes the detection more difficult. The present paper mainly focuses on researching the original TCP-oriented LDoS attack (it is commonly modelled by a series of periodic square bursts) in random early detection (RED) router scenario.

In this scenario, LDoS would hinder the RED from stabilising its queue, and hence resulting in a noisy feedback signal to the TCP congestion control, which in turn would lead to high jitters due to oscillations, as well as inefficiencies due to queue drainage, i.e. the decline of the throughput. Owing to the popularity of TCP + RED in current networks, the countermeasure against such attack is worthy tackling.

RED itself and its variants [8–12] have been proven to be not robust enough to counter short burst LDoS attacks [2, 13]. To defend against LDoS attacks, many strategies have been proposed. In the beginning, some efforts were devoted to mitigating the damage without detecting it but through modifying existing protocols or increasing extra resources. Kuzmanovic and Knightly [14] proposed the retransmission timeout (RTO) randomisation to defend the RTO-matched LDoS attacks. However, they argued that LDoS attacks could still filter out portions of TCP traffic. On another hand, this method is difficult to popularise, as it requires to modify the TCP protocol. Sarat and Terzis [15] indicated that a relatively small increase in the buffer size was sufficient to render the LDoS attack ineffective. As the buffer size increases, the attackers need to transmit at a much higher rate to fill the router buffer, at which point they are on longer LDoS attacks, and can be detected by RED with preferential propping. The limitation of this work is that the over-buffered router will increase the queuing delay of normal packets. Afterwards, researchers attempted to first detect if the LDoS attacks are launched and then filtered the attack traffic. Sun *et al.* [16] indicated that LDoS attacks can be detected by matching its features such as a high rate, short burst, and periodic. On the basis of the above features, they used a deficit round-robin algorithm to allocate bandwidth and protect legitimate flows. However, the false positive rate is relatively high. Legitimate flows thus suffer in the rate-limit packet filtering process. Chen *et al.* [17] extracted the attack features in the frequency domain. They used the normalised cumulative power spectrums density to calculate the distance of distribution curves between TCP traffic and LDoS traffic to determine the existence of attacks. After that, they established black and white lists to cut-off attack flows. However, this method needs extra storage to store the feature tables.

In summary, there are still many problems in detecting and defending LDoS attacks. The present work targets at improving detection performance. We attempt to explore a multi-dimensional feature of LDoS attacks by modelling the queue distribution. Also, we try to identify each LDoS attack burst as soon as possible by estimating the attack period. The primary contribution of this paper can be summarised as follows:

(i) A feedback control model under LDoS attacks is exploited. On the basis of this model, RED queue behaviours are revealed. According to the queue behaviours, we estimate the attack period which is used as the detection window for timely attack detection.
(ii) A queue distribution model composed of the instantaneous queue and the average queue is proposed to extract LDoS attack feature. The proposed queue distribution feature is structured from two dimensions, so it has higher distinguishability from legitimate random bursts without attack features. Finally, a combination of a distance-based approach and an adaptive threshold algorithm is proposed for identifying every LDoS attack burst. The proposed approach outperforms other detection approaches in terms of higher accurate detection rate, lower false negative rate, lower false positive rate, and lower algorithm complexity.

The remaining of this paper is organised as follows. Section 2 presents related works. Section 3 exploits a feedback control model to describe the attack process. Section 4 reveals the queue

**Table 1** Commonly used notations

| Notation | Definition |
|---|---|
| $B$ | router buffer size |
| $C$ | bottleneck link capacity |
| minRTO | minimum RTO |
| $Q$ | average queue length |
| $q$ | instantaneous queue length |
| $Q_{min}$ | minimum queue threshold of the RED algorithm |
| $Q_{max}$ | maximum queue threshold of the RED algorithm |
| $w$ | weight of the RED algorithm |
| $R$ | attack burst rate |
| $L$ | attack burst length |
| $T$ | attack burst period |
| $N$ | heterogeneous-flow number |
| $d$ | equivalent homogeneous delay |
| $n$ | equivalent homogeneous-flow number |
| $d_{th}$ | detection threshold |

behaviours and builds a two-dimensional (2D) queue distribution model to extract attack features. Moreover then, a combination of a distance-based approach and an adaptive threshold algorithm is proposed to detect attack bursts. Section 5 presents test results from NS-2 simulation and test-bed experiments to test the performance of the detection approach. Section 6 summarises our findings and discusses future works.

## 2 Related works

Kuzmanovic first modelled the TCP-oriented LDoS attack and denoted it as shrew [1]. Then, such attack is named as RoQ by Guirguis [2] and PDoS by Luo [18]. Although these LDoS attacks are denoted by different names, they have the same pattern (i.e. periodic short bursts). Subsequently, LDoS attacks were extended to other platforms such as cloud computing, application, and software defined networks (SDNs). FRC [3, 4] is a slow-and-low attack over a longer duration of time that exploits the pay-as-you-go pricing model in cloud computing. By fraudulently employing the cloud resources (e.g. bandwidth and web content), the attacker incurs huge economic damage to the cloud consumers. Low-rate application DoS attacks (e.g. slowreq [19], slow next [20], and tail attacks [7]) exploit specific application vulnerabilities to damage the quality of service. These DoS attacks send specially crafted messages to exploit the specific behaviour of applications (e.g. how requests are processed) instead of generating a huge flood of traffic to overwhelm the network. Low-rate flow table overflow attack [21] and slow ternary content-addressable memory exhaustion attack (slow ternary content-addressable memory (TCAM) [22] are two types of SDN-oriented LDoS attacks. Although their specific implementations are slightly different, both of them take advantage of the same vulnerability of the timeout mechanism in openflow. The attacker gradually and periodically installs malicious rules to overflow the flow table. In this case, no rule is uninstalled leaving the flow table always full, and legitimate rules are not allowed to install.

So far, a huge amount of researches exist on LDoS detection. Researchers mainly focus on the application of the network traffic features to distinguish between the attack traffic and legitimate traffic. Commonly, some mathematical algorithms such as signal processing, machine learning, and information metric are used to build the detection model. Luo and Chang [18] found that LDoS attacks incurred abnormal fluctuation of incoming traffic rate and decline of outgoing TCP acknowledgement (ACK)s. On the basis of these features, they further employed a discrete wavelet transform and cumulative sum to detect the changing points. Aiello *et al.* [23] deeply studied slow DoS attacks. They defined various slow DoS attacks in detail and completely categorised them [5]. Moreover, they made great contributions to defend against these attacks [24]. They tracked the number of packets received by the web server in a given period of time and then used the combination of the Fourier transform and the mutual information to identify the frequency-spectra feature of slow DoS attacks. Wu *et al.* [25] indicated that the aggregate network traffic was multifractal. On the basis of the algorithm of multifractal detrended fluctuation analysis, they proposed a threshold detection approach through the Hölder exponent. This approach can determine the start and the end of LDoS attack. Yue *et al.* [26] proposed a new identification approach based on wavelet transform and combined neural network to classify normal network traffic and LDoS attack traffic. Wavelet energy spectrum coefficients extracted from the sampled traffic were used for multifractal analysis of traffic over different time scales. The combined neural network was designed to classify multi-scale spectrum coefficients that presented different multifractal characteristics belonging to normal network traffic and LDoS attack traffic.

The advantage of the traffic-based detection approach is to simplify the analysis of the entire aggregate of traffic only [24]. However, these approaches commonly have three major shortcomings: (i) most of them are only able to determine if LDoS attack traffic exists in networks by a long sample period but unable to identify each attack burst immediately [17, 23]. (ii) A few of them are robust for normal traffic bursts commonly existed in networks. A legitimate random burst is easily identified as an attack burst [15, 27]. (iii) These existing approaches generally have a high complexity, which consumes computing and storage resources [25].

All the above research outcomes motivate us to explore a new model to expose the router queue behaviours under LDoS attack, and then design a new detection approach based on the queue distribution.

## 3 Modelling

In this section, we briefly review the original LDoS model, and then establish a feedback control model to describe the process of congestion control. The definitions of commonly used notations in this paper are listed in Table 1.

### 3.1 Attack model

As shown in Fig. 1*a*, the LDoS attack is originally modelled by a series of square bursts with three parameters: burst width $L$, burst rate $R$, and burst period $T$ [1, 28, 29]. A successful LDoS attack will have burst rate $R$ large enough to induce packet loss, burst length $L$ long enough to induce queue severe fluctuation but short enough to avoid detection, and $T$ is chosen so that when the queue attempts to be stable, it is faced with another fluctuation. In the premise of satisfying the above requirements, the periodic square burst model is commonly used because (i) it is easy for implementation by the fixed square pattern. (ii) As shown in Figs. 1*b* and *c*, it is easy to synchronise or aggregate the bursts to form the distributed LDoS attacks [30, 31].

### 3.2 Feedback control model

Previous studies have noted that the RED algorithm is vulnerable to LDoS attacks [1, 2]. The basic idea behind the RED algorithm is to avoid congestion depending on the average queue length which is calculated by exponentially weighted moving average (EWMA) method [8]

$$Q(n) = (1 - w) \times Q(n - 1) + w \times q(n) \tag{1}$$

In (1), $Q$ is the average queue length, $q$ is the instantaneous queue length, and $w$ is the weight. RED monitors the average queue length, and randomly drops (or marks when used in conjunction with explicit congestion notification) packets when congestion is approaching.

In practise, RED recalculates the average queue length only when a new packet is arriving. Considering a condition of a single TCP connection through a RED router, a case such as below would be challenging for RED: assuming the congestion happens, the calculated value of the average queue length would be large. If the average queue length is still calculated by (1) when new packets

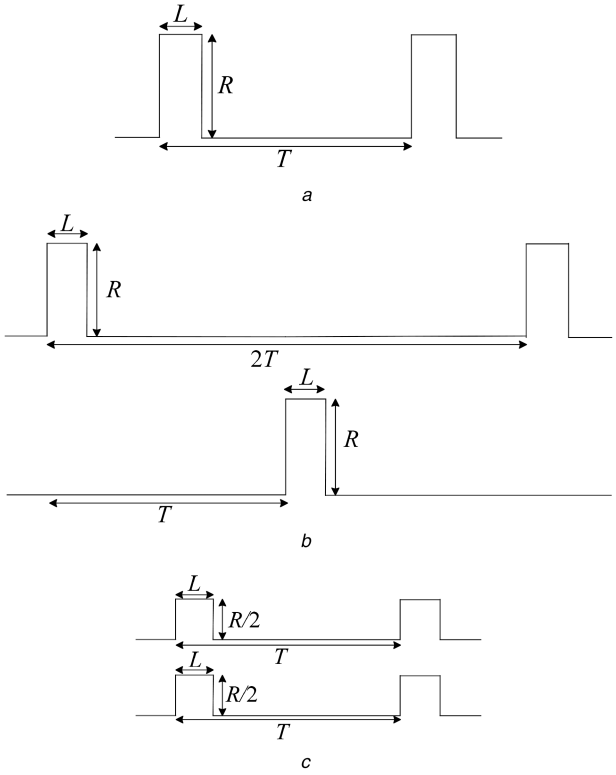



**Fig. 1** *Modelling the LDoS attack*
***(a)*** Model of a single attack source, ***(b)*** Model of synchronisation, ***(c)*** Model of aggregation

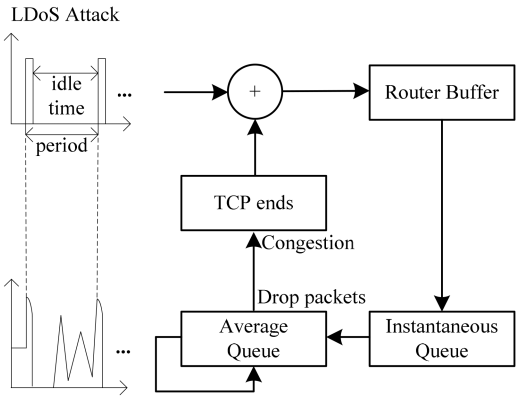



**Fig. 2** *Feedback control model*

arrive after congestion, the decrease rate of the average queue length will be very slow. This case will lead to a short-term high dropping rate. On the other hand, the instantaneous queue will be empty, because no new packets arrive due to the congestion control of the TCP end. Ideally, packets arriving at the router queue should be buffered if the instantaneous queue is empty. Therefore, when the instantaneous queue is empty, RED calculates the average queue length as if $m$ packets have arrived at the router with a queue length of zero [8]

$$\begin{cases} m = (\text{time} - q\_\text{time})/t_a \\ Q(n) = (1-w)^m \times Q(n-1) \end{cases} \tag{2}$$

where time is the current time, q_time is the start of the queue idle time, and $t_a$ is a typical transmission time for a small packet [8]. Equation (2) makes the average queue length decrease rapidly for the case that the instantaneous queue is empty.

The target of the LDoS attack is to congest the router queue, consequently forcing the TCP sender to decrease its congestion window (cwnd) [1, 2]. The attack process can be described by a feedback control model shown in Fig. 2.

As Fig. 2 shows, a high rate LDoS attack burst forces average queue length to increase rapidly, and causes a large number of legitimate TCP packets to drop. Moreover then, RED mechanism feeds back a congestion signal to TCP senders, and TCP senders decrease cwnds by multiplicative decrease mechanism to slow sending rate accordingly or even enter a timeout during the idle time between two attack bursts. In this case, the instantaneous queue length decreases rapidly or even becomes empty, which would cause a decrease of the average queue length. In response, RED will gradually decrease packet dropping rate, and TCP ends will gradually recover from timeout to retransmit packets. TCP sender's cwnd will perform slow start and additive increase (AI) [32] to keep filling the router queue. As soon as the average queue length returns to normal status, next attack burst will be launched to cause the same congestion progress as described above.

## 4 Detection based on the queue distribution model

Some existing LDoS attack detection approaches are easily impacted by legitimate random bursts in the network. The main reason is that they commonly use single-dimensional attack feature to identify attack traffic, so they are not robust to legitimate short bursts (they always false positively report legitimate bursts as LDoS attack bursts). In this section, we build a 2D queue distribution model through the queue behaviours. By doing so, we expect to improve the feature resolution. Moreover, we use a simple distance-based approach to detect every LDoS attack burst based on the proposed queue distribution model.

### *4.1 Queue behaviours under LDoS attacks*

On the basis of the feedback control model, we reveal queue behaviours and deduce the attack period. Considering a single TCP flow and an LDoS attack flow travelling through a RED bottleneck link with a link capacity $C$ (given in packets per second), we assume that the TCP sender's window size is not limited by the receiver's advertised flow control window. The router buffer size is set as a bandwidth–delay product. RED queue behaviours during an attack period can be shown in Fig. 3.

The upper part of Fig. 3 presents the variation of TCP sender's cwnd with time. The middle part of Fig. 3 shows average queue length as a function of time. The lower part of Fig. 3 shows the instantaneous queue length as a function of time. $B$ denotes the router buffer size, $Q_{\min}$ denotes the RED's minimum threshold, and $Q_{\max}$ denotes the RED's maximum threshold. We divide an attack period $T$ into four sub-periods ($T_1$–$T_4$).

$T_1$: $T_1$ equals to the attack burst width $L$. During $T_1$, the router buffer is immediately stuffed by an attack burst with rate $R$ and width $L$, so the instantaneous queue length equals to the buffer size $B$ and the average queue length will grow to $Q_{\max}$ or even more as given by (1). Meanwhile, the TCP connection enters a timeout as the link is stuffed. The TCP sender will not send any packets until the RTO timer overflows. We assume that the instantaneous queue length increases from $Q_{\min}$ to $Q_1$ during $T_1$. The number of attack packets arriving at the queue during $T_1$ can be expressed as $k = L \times R/(8 \times \text{Attack\_pktsize})$, where Attack_pktsize denotes the attack packet size. According to (1), the instantaneous queue length at the end of $T_1$ can be deduced as

$$Q_1 = B - (1-w)^k \times (B - Q_{\min}) \tag{3}$$

$T_2$: $T_2$ equals to $\text{minRTO} - L$. During $T_2$, no packets arrive at the queue, so the average queue length will not update (it keeps its previous value). The instantaneous queue becomes empty immediately because the previous buffered packets are drained rapidly.

In addition, the instantaneous queue keeps empty until the first retransmitted TCP packet arrives at the queue, and then the average queue length will degrade to $Q_2$ as given by (2). $Q_2$ can be denoted as

$$Q_2 = (1 - w)^m \times Q_1 \tag{4}$$

where $m$ is given as $m = (\text{minRTO} - L)/t_a$.

$T_3$: The TCP sender implements slow start and exponentially increases its cwnd until the cwnd reaches the slow start threshold. Subsequently, the congestion avoid algorithm governs the transmission of new data, and the cwnd follows a linear increment with round trip time (RTT). During $T_3$, TCP's sending rate is less than the link capacity $C$, so the instantaneous queue is still empty. According to (1), we can deduce the average queue length $Q_3$ at the end of $T_3$ as follows:

$$Q_3 = (1 - w)^u \times Q_2 \tag{5}$$

where $u$ denotes the number of packets arriving at the queue.

Equation (5) is given with a single TCP flow. Next, we extend it to homogeneous networks with $N$ identical-delay flows and further extend it to the most common scenario of heterogeneous networks with $N$ different-delay flows. TCP flows with identical delay have synchronous behaviours (e.g. they simultaneously enter slow start status to increase cwnd, filling the buffer in the same way). Therefore, $N$ identical-delay flows can be treated as a TCP flow with $N$ times potency. Fred *et al.* [33] proposed an algorithm to convert a heterogeneous-flows scenario into an equivalent homogeneous-flows scenario. We assume a heterogeneous-flows scenario contains $N$ TCP flows with different one-way delays ($d_1$, $d_2$, …, $d_N$). We let $d$ denote the equivalent homogeneous delay, $N_{dj}$ denotes the equivalent flow number of the $j$th flow and $n$ denotes the total equivalent flow number. We can derive the equivalent $n$ homogeneous-flows scenario by

$$d = N / \sum_{i=0}^{N} 1/d_i \tag{6}$$

$$N_{dj} = d/d_j (j = 1, 2, \ldots, N) \tag{7}$$

$$n = \sum_{i=1}^{N} N_{di} \tag{8}$$

Consequently, the maximum value of cwnd and the slow start threshold for each TCP flow should be $(C \times 2d + q_{\max})/n$ and $(C \times 2d + q_{\max})/2n$, respectively, where $q_{\max}$ denotes the instantaneous queue length when the average length returns to $Q_{\min}$, and $2d$ denotes the RTT.

Accordingly, $T_3$ can be divided into two phases: the first phase is that the cwnd is exponentially increased from 1 to $(C \times 2d + q_{\max})/2n$. This phase lasts $\log_2[(C \times 2d + q_{\max})/2] \times d$ seconds. The second phase is that the cwnd is additively increased from $(C \times 2d + q_{\max})/2n + 1$ to $(C \times 2d)/n$, and this phase lasts $[(C \times 2d)/n - (C \times 2d + q_{\max})/2n]$ seconds.

Therefore, $T_3$ can be deduced as

$$T_3 = \log_2\left(\frac{C \times 2d + q_{\max}}{2n}\right) \times d + \left(\frac{C \times 2d}{n} - \frac{C \times 2d + q_{\max}}{2n}\right) \times d \tag{9}$$

$T_4$: The cwnd keeps the additive growth from $(C \times 2d)/n$ to $(C \times 2d + q_{\max})/n$. As soon as the cwnd exceeds the link capacity, the instantaneous queue will be filled constantly by extra packets. The average queue returns to $Q_{\min}$ slowly. As soon as the average queue length reaches $Q_{\min}$, the next attack burst will be launched. Note that there are no packets lost because the average queue length is less than $Q_{\min}$.

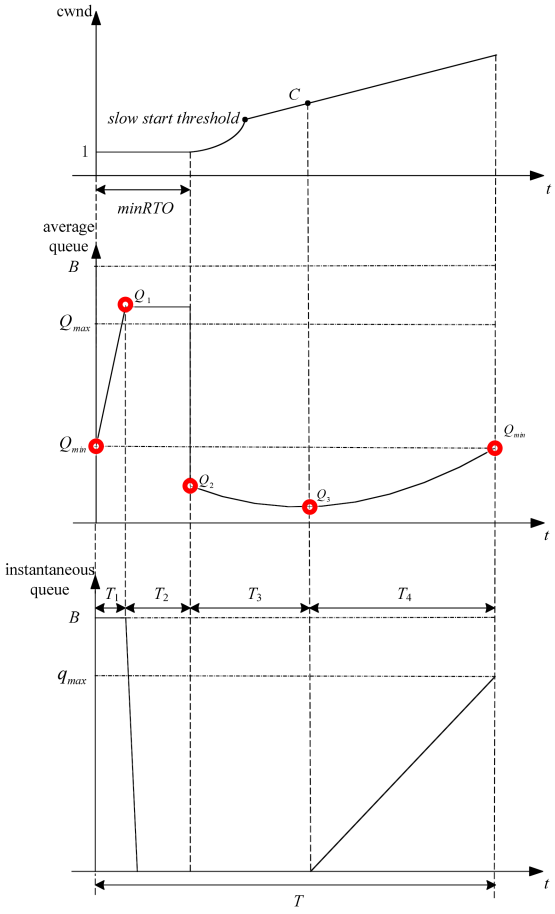



**Fig. 3** *RED queue behaviours during an attack period*

During $T_4$, TCP follows the AI mechanism. For the $i$th TCP connection, its cwnd is increased by $1/\text{cwnd}_i$ every time an ACK is received. We assume the equivalent cwnd size of $n$ homogeneous-flows is $\sum_{i=1}^{n} \text{cwnd}_i/n$, which equals the number of packets occupying the bottleneck link and the router buffer. It can be expressed as $\sum_{i=1}^{n} \text{cwnd}_i/n = C \times 2d + q(n-1)$. Moreover, when the $n$th packet arrives at the queue, the instantaneous queue length can be denoted as

$$q(n) = q(n-1) + \frac{1}{C \times 2d + q(n-1)} \tag{10}$$

Take (10) into (2), the average queue length can be denoted as (11) for $n$ equivalent flows

$$Q(n) = (1 - w) \times Q(n-1) + w \times \left(q(n-1) + \frac{n}{C \times 2d + q(n-1)}\right) \tag{11}$$

In addition, $T_4$ lasts $[(C \times 2d + q_{\max})/n - (C \times 2d)/n]$ equivalent homogeneous RTTs. Hence, $T_4$ can be deduced as

$$T4 = \left(\frac{C \times 2d + q_{\max}}{n} - \frac{C \times 2d}{n}\right) \times d + \sum_{q=1}^{q_{\max}/n} \frac{q \times n}{C} \tag{12}$$

where $\sum_{q=1}^{q_{\max}/n} q \times n/C$ denotes the time induced by queuing.

Since the initial value of the instantaneous queue length and the final value of the average queue length are known, the variable $q_{\max}$ can be solved by iterating (10) and (11).

As analysed above, the attack period can be denoted as

$$T = \min \text{RTO} + T_3 + T_4 \tag{13}$$

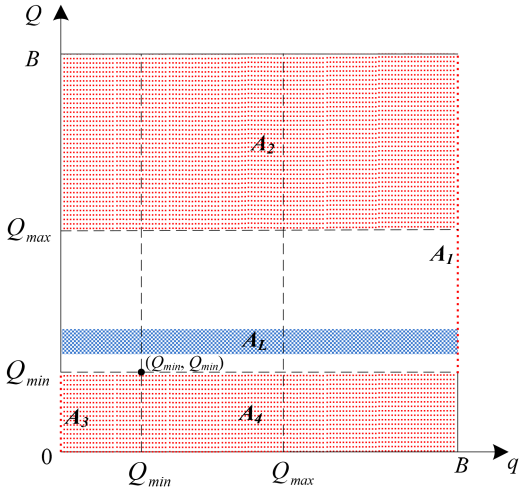



**Fig. 4** *2D queue distribution model*

To detect every attack burst as soon as possible, we set a detection window equalling to the attack period $T$. By doing so, we attempt to identify an attack burst in a detection window.

### *4.2 2D queue distribution model*

According to the queue behaviours analysed above, the instantaneous queue and the average queue inevitably exhibit abnormal features under LDoS attacks. To characterise the attack features, we build a 2D queue distribution model by combining the instantaneous queue length and the average queue length. Fig. 4 shows the 2D queue distribution model.

In Fig. 4, $X$-axis denotes the instantaneous queue length and $Y$-axis denotes the average queue length. The areas $A_1$, $A_2$, $A_3$, and $A_4$ in Fig. 4 represent the queue distribution under LDoS attacks. These four distribution areas, respectively, correspond to the four sub-periods in Fig. 3. If we sample the instantaneous queue length and the average queue length during $T_1$, these samples will distribute in $A_1$ which is a line in fact with ($q = B$, $Q_{\min} < Q < B$). Similarly, $A_2$ ($q = 0$, $0 < Q < Q_{\min}$) corresponds to $T_2$, $A_3$ ($q = 0$, $0 < Q < Q_{\min}$) corresponds to $T_3$, and $A_4$ ($Q_{\min} < q < B$, $0 < Q < Q_{\min}$) corresponds to $T_4$. In addition, the area $A_L$ presents the queue distribution in normal conditions, where the average queue length keeps a little greater than $Q_{\min}$ and the instantaneous queue length fluctuates around the average queue length. Especially, if the legitimate short bursts appear in networks, the instantaneous queue length will increase significantly, whereas the increment of the average queue length will be slow as given by (1). These behaviours reflect the effectiveness of RED in controlling the queue's stability.

Fig. 4 implies that the queue distribution in normal conditions will centralise on the centre ($Q_{\min}$, $Q_{\min}$), especially in the direction of $q$. In contrast, LDoS attacks cause the points of queue distribution to deviate from the centre ($Q_{\min}$, $Q_{\min}$). Even if a legitimate burst appears, such burst is commonly random, so it will not induce the queue distribution to achieve the abnormal range, whereas an LDoS attack burst with well-configured attack parameters inevitably causes the router queue to fluctuate severely.

### *4.3 Distance-based adaptive threshold detection*

On the basis of the feature of the queue distribution, we use a distance-based approach to detect every LDoS attack burst. We let $d_{\text{AED}}$ denote the average Euclidean distance (AED) between the sample points and the centre ($Q_{\min}$, $Q_{\min}$). Here, $d_{\text{AED}}$ is defined as

$$d_{\text{AED}} = \frac{\sum_{i=1}^{N_{\text{p}}} \sqrt{[w \times (q_i - Q_{\min})]^2 + [(1-w) \times (Q_i - Q_{\min})]^2}}{N_{\text{p}}} \quad (14)$$

where $N_{\text{p}}$ denotes the number of the sample points during a detection window. To detect every attack burst as soon as possible, the detection window can be set as the attack period as given by (13). Moreover, $w$ is the weight of the RED algorithm. The direction of the average queue is given a greater weight because RED only allows the average queue to fluctuate slightly in normal conditions even if short bursts appear. In this way, the devotion for $d_{\text{AED}}$ from the average queue is extended, so it can identify an LDoS attack burst without the impact of legitimate bursts.

Using the calculated AEDs, we can set a threshold $d_{\text{th}}$ to identify the LDoS attack burst. It is considered an LDoS attack burst existed if $d_{\text{AED}}$ is greater than $d_{\text{th}}$. Otherwise, there is no LDoS attack burst.

The threshold value is a crucial parameter that directly affects the detection rate, the false negative rate, and the false positive rate. In real networks, using a fixed threshold value might be less practical since it is impossible to tune threshold values for various types of network traffic. If the network traffic is steady but the detector has a high threshold, the false negative rate will be increased. Meanwhile, if the network traffic is unsteady but the threshold value is low, the detector will work very sensitively [34]. Here, we design an adaptive algorithm based on the EWMA algorithm to dynamically tune the threshold. The EWMA has an ability to absorb transient bursts, so it can reduce the impact of the normal burst on detection performance [35, 36].

We define a sliding window with a length of $\beta T$ and step of $T$, where $\beta$ is a positive integer. We let $d_{\text{AED}}(i)$ denote the AED in the $i$th detection window $T$ and $d_{\text{th}}(i)$ denote the threshold value. Moreover then, we use the EWMA to implement an adaptive threshold algorithm. The threshold is given as

$$d_{\text{th}}(i) = \mu(i-1) + 3\sigma(i-1) \quad (15)$$

where $\mu(i-1)$ is the mean of AEDs in the prior sliding window, which is updated as EWMA algorithm

$$\mu(i) = (1-w) \times \mu(i-1) + w \times d_{\text{AED}}(i) \quad (16)$$

where $w$ is the weight of the RED and $\sigma(i)$ is the standard deviation with $\mu(i)$. Here, $3\sigma$ error level could give us a high confidence interval [37], which is good enough even in high-precision detection scenarios [30]. Finally, we can make a decision as the following rule:

If $d_{\text{AED}}(i) < d_{\text{th}}(i)$, then no attack burst.
If $d_{\text{AED}}(i) \geq d_{\text{th}}(i)$, then an attack burst exists.

In addition, $d_{\text{th}}(i)$ is updated only under normal conditions. If an attack burst is identified, $d_{\text{th}}(i)$ will not be updated. This rule can avoid forming an over-high threshold.

## 5 Experiments and results analyses

In this section, we conduct experiments through both NS-2 simulation and test bed to evaluate the effectiveness of our proposed detection approach.

### *5.1 NS-2 simulation experiments*

First, we conduct NS-2 experiments to verify our proposed model. The NS-2 simulation topology is shown in Fig. 5. A bottleneck RED link of 10 Mbps capacity is shared by 15 TCP connections and a single constant bit rate (CBR) connection representing the attack traffic. The rate for each access link is 100 Mbps. The min*RTO* is set to its default value 1 s [1, 31]. The RTT ranges from 20 to 430 ms. The bottleneck buffer size is given by the bandwidth–delay product. The RED queue minimum and maximum thresholds are set to 50 and 150, respectively. The RED queue weight is 0.001. The simulation period is 60 s and the attack starts at 30 s. Attack burst parameters are set as burst length $L$ = 0.3 s, burst rate $R$ = 10 Mbps, and burst period $T$ = 4.5 s as given by (13).

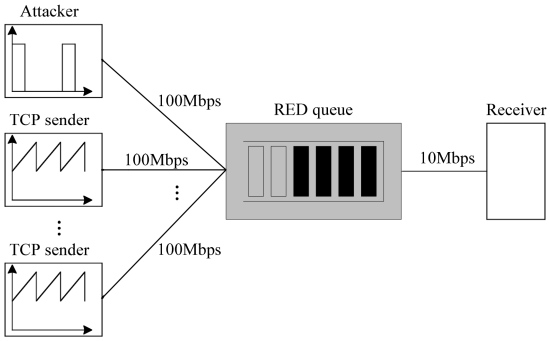
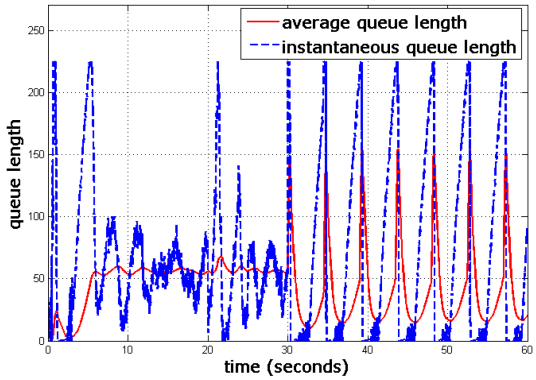
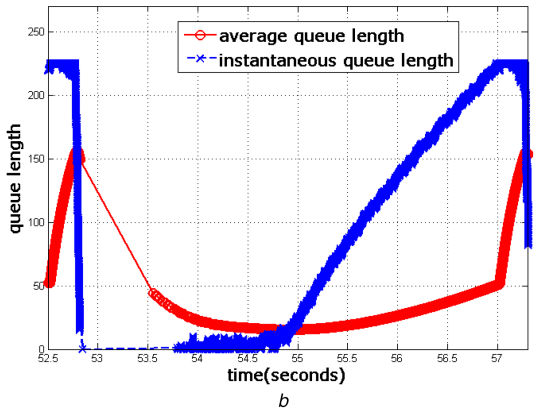



**Fig. 5** *NS-2 network topology*




*a*




*b*

**Fig. 6** *Effect of LDoS attacks on RED queue*
***(a)*** Overview, ***(b)*** Zoom in a sample window

We simulate a random legitimate burst between 20 and 30 s by establishing a new TCP connection between a client and the server (the slow start happens when the connection starts). As the present work mainly focuses on distinguishing every LDoS burst in its attack period, we mainly consider the legitimate short bursts whose behaviours are more similar to LDoS burst (the length of such short burst commonly ranges from tens to hundreds of milliseconds). These short bursts are usually confused with LDoS bursts [15, 27]. For example, Sarat and Terzis [15] have indicated that short TCP bursts would be unduly penalised by previous approaches such as halting anomaly with weighted choKing (HAWK) [27], due to their burst nature. Meanwhile, the consecutive or longer-duration bursts such as flash crowds are more similar to the FDoS. There have been many methods to detect such legitimate bursts [38, 39].

Fig. 6 depicts the effect of LDoS attacks on the RED queue. Fig. 6*a* shows the variation of the queue with time. Fig. 6*b* zooms in an attack period.

In Fig. 6*a*, the RED queue stabilises after a short while, however, fluctuates severely after the attack is launched. In Fig. 6*b*, we take a closer look at the queue behaviours during an attack period (52.5–57 s).

Observe that the experiment results in Fig. 6*b* match well with the theoretical model in Fig. 3. Note that there is a period of idle time during which the average queue length is not recorded. This is

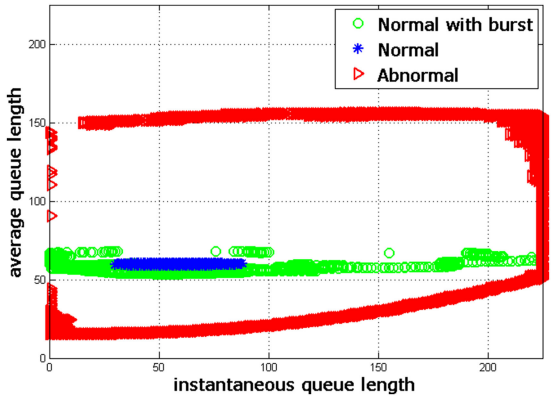



**Fig. 7** *Queue distribution*

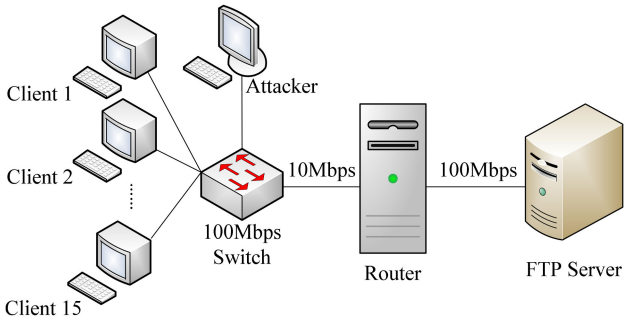



**Fig. 8** *Test-bed network topology*

because NS-2 records the queue length only when a packet is arriving at the queue. Since no packets arrive at the queue during the idle time, the average queue length will keep its previous value. After that, the average queue length directly declines under the minimum RED threshold.

In addition, the legitimate short burst in Fig. 6*a* causes the instantaneous queue to fluctuate severely, and the average queue controlled by RED to fluctuate slightly. On the other hand, the LDoS attack burst with well-configured attack parameters causes both the instantaneous queue and the average queue to fluctuate severely.

Furthermore, we concern the queue distributions in three cases: (i) select a period of normal traffic (10–14.5 s). (ii) Select a period of normal traffic mixed with a legitimate burst (20–24.5 s). (iii) Select a period of normal traffic mixed with an LDoS attack burst (52.5–57 s). Fig. 7 depicts the queue distributions. Observe that the test results match well with the theoretical model in Fig. 4.

### *5.2 Test-bed experiments*

In this section, we conduct test-bed experiments to evaluate the performance of our approach. Fig. 8 depicts the test-bed topology.

Two different IP fields are connected through a RED router. The router is a double-network-card personal computer with the operation system Linux RedHat version 2.6.39. Iproute and tc are used to configure the RED algorithm and the one-way propagation delays between clients and the server. Table 2 shows the parameters in detail, by which, RED can effectively control the queue's stability.

In addition to the settings above, 15 TCP clients and an LDoS attack source are linked to a 100 Mbps switch that connects the router through a 10 Mbps bottleneck link. We use a user datagram protocol-based attack tool developed by Rice University to launch LDoS attack bursts [1]. The attack parameters are set as $L = 300$ ms, $R = 10$ Mbps, and $T = 4.5$ s. The detection window is 4.5 s as calculated by (13). The parameter $\beta$ in the adaptive threshold algorithm is 10.

In the experiment, we conduct ten groups of tests, and each test lasts 900 s. In each test, we let clients randomly establish a TCP connection with file transfer protocol server to simulate legitimate short bursts. The behaviours of these legitimate bursts (e.g. rate and

**Table 2** Experiment parameters

| Parameter | Value |
|---|---|
| buffer size | 225 |
| RED minimum threshold | 50 |
| RED maximum threshold | 150 |
| RED weight | 0.001 |
| RTT | uniformly distributed within [20, 430 ms] |

**Table 3** Ten groups of test results

| Test group | Attack duration, s | Total of attack bursts | Detected bursts | False negative bursts | False positive bursts |
|---|---|---|---|---|---|
| 1 | 320 | 72 | 70 | 2 | 1 |
| 2 | 535 | 119 | 119 | 0 | 2 |
| 3 | 511 | 114 | 111 | 3 | 2 |
| 4 | 334 | 75 | 75 | 0 | 1 |
| 5 | 306 | 68 | 67 | 1 | 2 |
| 6 | 502 | 112 | 110 | 2 | 0 |
| 7 | 429 | 96 | 95 | 1 | 2 |
| 8 | 304 | 68 | 68 | 0 | 1 |
| 9 | 549 | 122 | 119 | 3 | 3 |
| 10 | 487 | 109 | 108 | 1 | 2 |
| total | 4277 | 955 | 942 | 13 | 16 |

**Table 4** Comparison of different detection approaches

| Names | Items | | | | |
|---|---|---|---|---|---|
| | Detection accuracy | | | Complexity | |
| | $P_D$, % | $P_{FN}$, % | $P_{FP}$, % | Space | Time |
| NCAS | 87.4 | 12.6 | 17.9 | $O(n)$ | $O(n^2)$ |
| multifractal | 91.2 | 8.8 | 14.3 | $O(n \log_2 n)$ | $O(n \log_2 n)$ |
| our approach | 98.6 | 1.4 | 1.7 | $O(n)$ | $O(n)$ |

duration) are random, due to the different network parameters involved such as RTT, queue length, and current extent of congestion. The LDoS attack starts randomly between 150 and 300 s. Also, we randomly choose an integer ranging from 300 to 600 s as the attack duration. Test results are presented in Table 3.

In Table 3, the first column is the sequence number of ten groups of tests. The second column is the duration of the attack in each group. The third column presents the sum of attack bursts. The fourth column presents the number of accurately detected attack bursts. The fifth column presents the number of attack bursts without detection. The sixth column presents the number of normal bursts assumed erroneously to attack bursts. In total, the LDoS attack flow is conducted for 4277 s including 955 attack bursts. Here, 942 attack bursts are reported accurately, 13 attack bursts are reported false negatively, and 16 legitimate bursts are reported false positively. The average detection rate is 98.6%, the false negative rate is 1.4%, and the false positive rate is 1.7%.

Furthermore, our approach is compared with two existing LDoS attack detection approaches the classic normalised cumulative amplitude spectrums (NCASs) approach [17] and the multifractal approach [25]. These approaches are implemented in the same experiment environment. Table 4 reports the comparison results.

Test results illustrate that our approach outperforms other approaches in terms of higher accurate detection rate $P_D$, lower false negative rate $P_{FN}$, and especially, lower false positive rate $P_{FP}$. Legitimate short bursts are the main factor that leads to the high false positive rate in the former two approaches. Contrarily, our approach is robust to them. This is because the extracted attack feature is structured from two dimensions, so it presents higher distinguishability between LDoS attack burst and legitimate burst. In addition, space and time complexities of our approach are lower than the other two approaches. Moreover, unlike other approaches, our approach can detect each attack burst, rather than roughly determine if the attack occurs in a long sample period. Therefore, our approach presents real-time and fine-grained performance.

## 6 Conclusion

In this paper, we build a feedback control model to describe the process of RED congestion control, by which the congestion window and router queue behaviours are analysed combinedly. Moreover then, we propose a 2D queue distribution model to extract attack features. After that, we combined the AED with a simple adaptive threshold algorithm to detect every LDoS attack burst. Experiments in NS-2 simulation platform, as well as test-bed network environment, are conducted to test the detection performance. Test results prove that our detection approach outperforms other existed approaches in terms of three aspects: (i) it is robust to legitimate random bursts, so it presents a low false positive rate. (ii) Its algorithm complexity is low. (iii) It can detect every attack burst in time by estimating an appropriate detection window.

DoS attack defence is an endless game between attacker and defender. The winner is who cost less but own more. In future, research efforts can be devoted to the following aspects: (i) expose new vulnerabilities that can be exploited by LDoS attacks in new scenarios such as application layer LDoS attack, LDoS attack in the cloud, and LDoS attack in information centric networking (ICN). (ii) Develop the existing attack models to enhance concealment and attack potency, which has a close relevancy for the assessment of the extent to which defence mechanisms are capable of mitigating the attack's impact. Attack traffic synchronisation and aggregation are worthy tackling. (iii) Explore new attack detection and mitigation methods. Researchers can consider the combination of the existing attack features and higher-layer features for LDoS detection (e.g. IP 5-tuples, TCP flags, and time to live (TTL)). In addition, SDN technology, resilience mechanism, and game theory are worthy concerning LDoS mitigation.

## 7 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Nos. 61601467, U1533107, and U1433105), the Natural Science Foundation of Tianjin (No. 17JCZDJC30900), and the Fundamental Research Funds for the Central Universities of CAUC (No. 3122018C003).

## 8 References

[1] Kuzmanovic, A., Knightly, E.W.: 'Low-rate TCP-targeted denial of service attacks: the shrew versus the mice and elephants'. ACM SIGCOMM Computer Communication Review, Kalrushe, Germany, 2003, pp. 75–86

[2] Guirguis, M., Bestavros, A., Matta, I.: 'Exploiting the transients of adaptation for RoQ attacks on Internet resources'. Proc. Int. Conf. Network Protocols ICNP, Berlin, Germany, 2004, pp. 184–195

[3] Idziorek, J., Tannian, M., Jacobson, D.: 'Attribution of fraudulent resource consumption in the cloud'. Proc. IEEE Int. Conf. Cloud Computing, Honolulu, HI, USA, 2012, pp. 99–106

[4] Idziorek, J., Tannian, M., Jacobson, D.: 'Detecting fraudulent use of cloud resources'. Proc. ACM Conf. Computer Communications Security, Chicago, Illinois, USA, 2011, pp. 61–72

[5] Cambiaso, E., Papaleo, G., Chiola, G., *et al.*: 'Slow DoS attacks: definition and categorisation', *Int. J. Trust Manage. Comput. Commun.*, 2013, **3**, (20), pp. 300–319

[6] Ficco, M., Rak, M.: 'Stealthy denial of service strategy in cloud computing', *IEEE Trans. Cloud Comput.*, 2015, **3**, (1), pp. 80–94

[7] Shan, H.S., Wang, Q.Y., Pu, C.: 'Tail attacks on web applications'. Proc. ACM Conf. Computer Communications Security, Dallas, TX, USA, 2017, pp. 1725–1739

[8] Floyd, S., Jacobson, V.: 'Random early detection gateways for congestion avoidance', *IEEE ACM Trans. Netw.*, 1993, **1**, (4), pp. 397–413

[9] Wang, C.G., Liu, J.C., Li, B., *et al.*: 'LRED: a robust and responsive AQM algorithm using packet loss ratio measurement', *IEEE Trans. Parallel Distrib. Syst.*, 2007, **18**, (1), pp. 29–43

[10] Cheng, M., Wang, H., Yan, L.: 'Dynamic RED: a modified random early detection', *J. Comput. Inf. Syst.*, 2011, **7**, (14), pp. 5243–5250

[11] Ott, T.J., Lakshman, T.V., Wong, L.H.: 'SRED: stabilized RED'. Proc. IEEE INFOCOM, New York, USA, 1999, vol. **3**, pp. 1346–1355

[12] Feng, W.C., Shin, K.G., Kandlur, D.D., *et al.*: 'The blue active queue management algorithms', *IEEE/ACM Trans. Netw.*, 2002, **10**, (4), pp. 513–528

[13] Tang, Y.J., Luo, X.P., Hui, Q., *et al.*: 'Modeling the vulnerability of feedback-control based Internet services to low-rate DoS attacks', *IEEE Trans. Inf. Forensics Sec.*, 2014, **9**, (3), pp. 339–353

[14] Kuzmanovic, A., Knightly, E.W.: 'Low-rate TCP-targeted denial of service attacks and counter strategies', *IEEE ACM Trans. Netw.*, 2006, **14**, (4), pp. 683–696

[15] Sarat, S., Terzis, A.: 'On the effect of router buffer sizes on low-rate denial of service attacks'. Proc. Int. Conf. Computing Communications Networks ICCCN, San Diego, CA, USA, 2005, pp. 281–286

[16] Sun, H.B., Lu, J.C.S., Yau, D.K.Y.: 'Defending against low-rate TCP attacks: dynamic detection and protection'. Proc. Int. Conf. Network Protocols ICNP, Berlin, Germany, 2004, pp. 196–205

[17] Chen, Y., Hwang, K., Kwok, Y.K.: 'Collaborative detection and filtering of shrew DdoS attacks using spectral analysis', *J. Parallel Distrib. Comput.*, 2006, **66**, (9), pp. 1137–1151

[18] Luo, X.P., Chang, R.K.C.: 'On a new class of pulsing denial-of-service attacks and the defense'. Int. Conf. Network Distributed System Security Symp., San Diego, CA, USA, 2005, pp. 61–79

[19] Aiello, M., Papaleo, G., Cambiaso, E.: 'Slowreq: a weapon for cyberwarfare operations. Characteristics, limits, performance, remediations', *Adv. Intell. Syst. Comput.*, 2014, **2014**, pp. 537–546

[20] Cambiaso, E., Papaleo, G., Chiola, G., *et al.*: 'Designing and modeling the slow next DoS attack', *Adv. Intell. Syst. Comput.*, 2015, **2015**, pp. 249–259

[21] Cao, J.H., Xu, M.W., Qi, L., *et al.*: 'Disrupting SDN via the data plane: low-rate flow table overflow attack', *Lect. Notes Inst. Comput. Sci. Soc. Inf. Telecommun. Eng.*, 2018, **2018**, pp. 356–376

[22] Tulio, A.P., Yuri, G.D., Iguatemi, E.F.: 'Slow TCAM exhaustion DDoS attack', *IFIP Adv. Inf. Commun. Technol.*, 2017, **2017**, pp. 17–31

[23] Aiello, M., Cambiaso, E., Mongelli, M., *et al.*: 'An on-line intrusion detection approach to identify low-rate DoS attacks'. Proc. Int. Carnahan Conf. Security Technol., Rome, Italy, 2014, pp. 1–6

[24] Mongelli, M., Aiello, M., Cambiaso, E., *et al.*: 'Detection of DoS attacks through Fourier transform and mutual information', *IEEE Int. Conf. Commun.*, 2015, **2015**, pp. 7204–7209

[25] Wu, Z.J., Zhang, L.Y., Yue, M.: 'Low-rate DoS attacks detection based on network multifractal', *IEEE Trans. Dependable Secur. Comput.*, 2016, **13**, (5), pp. 559–567

[26] Yue, M., Liu, L., Wu, Z.J., *et al.*: 'Identifying LDoS attack traffic based on wavelet energy spectrum and combined neural network', *Int. J. Commun. Syst.*, 2018, **31**, (2), pp. 1–16

[27] Kwok, Y.K., Tripathi, R., Chen, Y., *et al.*: 'HAWK: halting anomalies with weighted choking to rescue well-behaved TCP sessions from shrew DDoS attacks', *Lect. Notes Comput. Sci.*, 2005, **2005**, pp. 423–432

[28] Lou, J., Yang, X., Wang, X., *et al.*: 'On a mathematical model for low-rate shrew DDoS', *IEEE Trans. Inf. Forensics Sec.*, 2014, **9**, (7), pp. 1069–1083

[29] Yue, M., Wu, Z.J., Wang, M.X.: 'A new exploration of FB-shrew attack', *IEEE Commun. Lett.*, 2016, **20**, (10), pp. 1987–1990

[30] Chen, Y., Hwang, K., Kwok, Y.K.: 'Filtering of shrew DDoS attacks in frequency domain'. Proc. Conf. Local Computer Networks LCN, Sydney, NSW, Australia, 2005, pp. 786–793

[31] Li, H., Zhu, J.H., Wang, Q.X., *et al.*: 'LAAEM: a method to enhance LDoS attack', *IEEE Commun. Lett.*, 2016, **20**, (4), pp. 708–711

[32] Allman, M., Paxson, V., Blanton, E.: 'TCP congestion control', Internet engineering task force'. RFC 5681, September 2009

[33] Fred, S.B., Bonald, T., Proutiere, A., *et al.*: 'Statistical bandwidth sharing: a study of congestion at flow level', *Comput. Commun. Rev.*, 2001, **2001**, pp. 111–122

[34] No, G., Ra, I.: 'Adaptive DDoS detector design using fast entropy computation method'. Fifth Int. Conf. Innovative Mobile & Internet Services in Ubiquitous Computing, Seoul, South Korea, 2011, pp. 86–93

[35] Siris, V.A., Papagalou, F.: 'Application of anomaly detection algorithms for detecting SYN flooding attacks'. GLOBECOM IEEE Global Telecommunications Conf., Dallas, TX, USA, 2004, pp. 2050–2054

[36] Tang, D., Chen, X.S., Liu, H.Y., *et al.*: 'Adaptive EWMA method based on abnormal network traffic for LDoS attacks', *Math. Probl. Eng.*, 2014, **2014**, pp. 1–12

[37] Devore, J.L., Farnum, N.R.: '*Applied statistics for engineers and scientists*' (Published by Duxbury Press, Pacific Grove, CA, USA, 1999)

[38] Sun, D.G., Yang, K., Shi, Z.X., *et al.*: 'A distinction method of flooding DDoS and flash crowds based on user traffic behavior'. IEEE Trustcom/BigDataSE/ICESS, Sydney, NSW, Australia, 2017, pp. 65–72

[39] Yu, S., Zhou, W.L., Jia, W.J., *et al.*: 'Discriminating DDoS attacks from flash crowds using flow correlation coefficient', *IEEE Trans. Parallel Distrib. Syst.*, 2012, **23**, (6), pp. 1073–1080