Computer Networks 56 (2012) 3417–3431

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Flow level detection and filtering of low-rate DDoS

Changwang Zhang [a,b,*], Zhiping Cai [a], Weifeng Chen [c], Xiapu Luo [d], Jianping Yin [a]

[a] School of Computer Science, National University of Defense Technology, Changsha, China
[b] Department of Computer Science, University College London, Gower Street, London WC1E 6BT, United Kingdom
[c] Department of Math & Computer Science, California University of Pennsylvania, USA
[d] Computing Department, Hong Kong Polytechnic University, China

ARTICLE INFO

Article history:
Received 29 December 2011
Received in revised form 27 May 2012
Accepted 2 July 2012
Available online 16 July 2012

Keywords:
DDoS
Detection
Low-rate DoS
Congestion

ABSTRACT

The recently proposed TCP-targeted Low-rate Distributed Denial-of-Service (LDDoS) attacks send fewer packets to attack legitimate flows by exploiting the vulnerability in TCP's congestion control mechanism. They are difficult to detect while causing severe damage to TCP-based applications. Existing approaches can only detect the presence of an LDDoS attack, but fail to identify LDDoS flows. In this paper, we propose a novel metric – Congestion Participation Rate (CPR) – and a CPR-based approach to detect and filter LDDoS attacks by their intention to congest the network. The major innovation of the CPR-base approach is its ability to identify LDDoS flows. A flow with a CPR higher than a predefined threshold is classified as an LDDoS flow, and consequently all of its packets will be dropped. We analyze the effectiveness of CPR theoretically by quantifying the average CPR difference between normal TCP flows and LDDoS flows and showing that CPR can differentiate them. We conduct ns-2 simulations, test-bed experiments, and Internet traffic trace analysis to validate our analytical results and evaluate the performance of the proposed approach. Experimental results demonstrate that the proposed CPR-based approach is substantially more effective compared to an existing Discrete Fourier Transform (DFT)-based approach – one of the most efficient approaches in detecting LDDoS attacks. We also provide experimental guidance to choose the CPR threshold in practice.

1. Introduction

Distributed Denial-of-Service (DDoS) attacks [1] have been identified as a major threat to today's Internet services. Being a new kind of DDoS attacks, TCP-targeted Low-rate Distributed Denial-of-Service (LDDoS) [2] attacks are more efficient in terms of causing damage to legitimate flows and more difficult to detect when compared to traditional flooding-based DDoS attacks.

Traditional flooding-based DDoS attacks employ a "sledge-hammer" approach of high-rate transmission of packets, which obviously distinguishes themselves from normal data flows in statistical characteristics. Many of the proposed approaches for detecting DDoS attacks have been based on these statistical characteristics [3–7].

LDDoS attacks are quite different from the traditional flooding-based DDoS attacks as they exploit the vulnerabilities in TCP's congestion control mechanism. Instead of sending continuous network traffic, an LDDoS attacker sends periodically pulsing data flows, which may dramatically reduce the average rate of attack flows. LDDoS attacks have already been observed in the Internet2 Abilene backbone [8], thus presenting a new challenge to the security of the Internet.

When facing large scale LDDoS attacks, existing defense approaches can only detect the presence of the LDDoS attack, but cannot determine whether a particular flow is an attack flow or not. In this paper we propose a novel metric "Congestion Participation Rate" (CPR) to identify

[*] Corresponding author at: Dept. of Computer Science, University College London, Gower Street, London WC1E 6BT, United Kingdom. Tel.: +44 0 20 7679 0371; fax: +44 0 20 7387 1397.

E-mail address: changwang.zhang.10@ucl.ac.uk (C. Zhang).

1389-1286 © 2012 Elsevier B.V. Open access under CC BY license.
http://dx.doi.org/10.1016/j.comnet.2012.07.003

LDDoS flows. The CPR-based approach exploits the fact that LDDoS flows actively induce network congestion whereas normal TCP flows actively avoid network congestion. That is, normal TCP flows will tend to send fewer packets during network congestion whereas LDDoS flows would not. The Congestion Participation Rate (CPR) can accurately capture this fundamental difference, and hence allow us to identify LDDoS flows. Our contributions are summarized as follows:

- We propose a novel metric – Congestion Participation Rate (CPR) to identify LDDoS flows by measuring the intention of network flows to congest the network. To the best of our knowledge, it is the first metric that could recognize LDDoS flows by quantifying each flow's intention of congesting the network.
- We propose and implement a CPR-based approach to detect and filter LDDoS attacks. The CPR-based approach is an originality innovation that can effectively identify LDDoS in a per-flow basis in large-scale LDDoS attacks as far as we are concerned.
- We conduct intensive experiments, including both ns-2 simulations and test-bed experiments, to validate our analytical results and evaluate the performance of the CPR-based approach. The experimental results demonstrate that the CPR-based approach is effective for all of the LDDoS attacks considered while the existing Discrete Fourier Transform (DFT)-based approach is only effective for a small set of LDDoS attacks.
- We obtain the trade-off between the detection rate and the false positive rate for the CPR-based approach through a comprehensive set of experiments. This trade-off provides experimental guidance for choosing a CPR threshold in practice.

It is worth noting that the CPR-based approach is designed to distinguish between normal TCP flows and LDDoS flows. Differentiating normal UDP flows and LDDoS flows will be investigated in future work. The LDDoS attacks in this paper, if not declared otherwise, precisely refer to TCP-targeted LDDoS attacks including Shrew attacks [2] and Pulsing DoS (PDoS) attacks [9]. The CPR-base approach is also designed to counter TCP-targeted LDDoS attacks [2,9].

The rest of the paper is organized as follows. Section 2 presents the modeling of LDoS attacks and LDDoS attacks. Section 3 defines the metric of CPR, describes the CPR-based approach and analyzes the boundaries of average CPR for normal and attack flows. Intensive experiments based on ns-2 simulations and real network configurations are presented in Sections 4 and 5 respectively. Section 6 discusses several important issues related to the deployment of the CPR-based approach. Section 7 reviews existing work that is directly related to the proposed approach. Finally we conclude the paper in Section 8.

2. Modeling LDDoS attacks

In this section we model the LDDoS attacks. Our target is to detect and filter LDDoS attack flows. A flow is uniquely determined by a 5-tuple (*Source IP, Source Port, Destination IP, Destination Port, Protocol*). We use four parameters (T_a, T_b, R_b, s) to describe an LDoS attack flow, where $T_a$is the LDoS attack period, T_b is the LDoS attack burst width (or pulsing width), R_b is the LDoS attack burst rate (or pulsing rate), and *s*is the starting time of the attack flow (see Fig. 1).

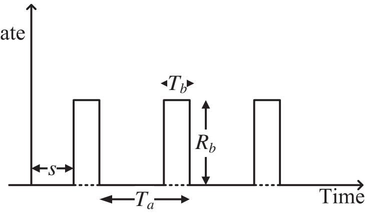


Fig. 1. LDoS attack flow.

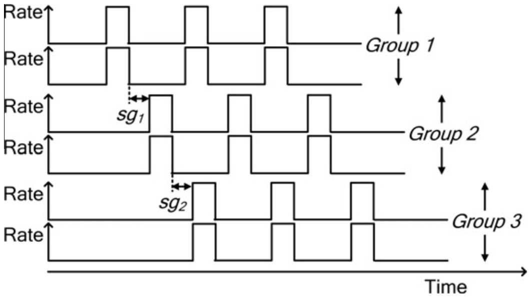


Fig. 2. An LDDoS attack. There are 3 LDDoS flow groups in this LDDoS attack. sg_1 is the starting gap between LDDoS flow groups 1 and 2. sg_2 is the starting gap between LDDoS flow groups 2 and 3. In modeling LDDoS attacks, we assume that both sg_1 and sg_2 are equal to a constant in an LDDoS attack.

An LDDoS attack consists of multiple LDoS attack flows, $F_1,\ F_2,\ \ldots\ F_n$, that may originate from different machines distributed on the Internet. Assuming that $T_a,\ T_b$, and R_b are identical for every LDoS flow F_i, we define an LDDoS flow group as a set of attack flows that have the same starting time *s*. The starting time of an LDDoS flow group is just the starting time of each flow in the group. We assume that the starting gap between consecutive LDDoS flow groups remains constant in an LDDoS attack. For example, in Fig. 2, sg_1 represents the starting gap between LDDoS flow groups 1 and 2, and sg_2 represents the starting gap between LDDoS flow groups 2 and 3. Our assumption corresponds to that both sg_1 and sg_2 are equal to a constant in an LDDoS attack. This starting gap is denoted as σ. We further assume that each group has an identical number, *m*, of flows. Based on the definition and assumption above, we describe an LDDoS attack using four parameters (n, g, m, σ), where *n* is the total number of flows in the attack, *g* is the number of attack flow groups, and *m* is the number of members in an LDDoS flow group. It is clear that $n = mg$ based on the above assumptions.

It is worth noting that the aforementioned assumption only eases the classification of LDDoS attacks and is not required by our detection system.

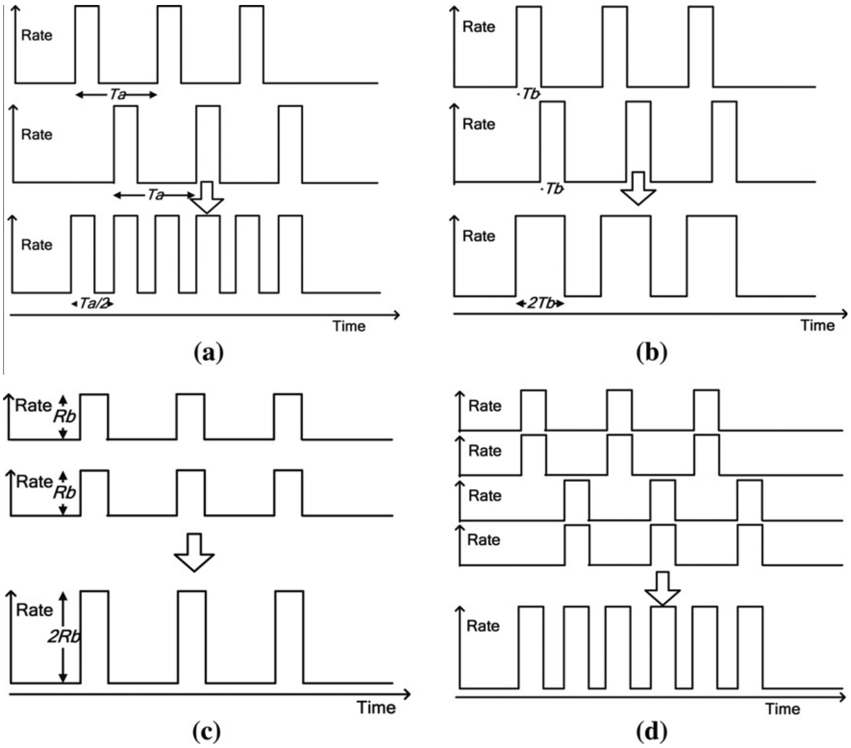


Fig. 3. LDDoS attack categories (a) Attack Frequency Intensification (AFI), (b) Attack burst Width Intensification, (c) Attack burst Rate Intensification, and (d) Mixed Intensification. The unit for Rate is Bytes/s and the unit for time is second.

Based on these assumptions and definitions, we classify LDDoS attacks into four categories (Fig. 3), according to how the three characteristics T_a, T_b, and R_b are being distributed among multiple flows in an LDDoS attack. Although our classification is by no means complete, it is enough for us to analyze the characteristics of the LDDoS attacks and to conduct experiments to evaluate our approach.

1. Attack Frequency Intensification (AFI) LDDoS attack($n > 0$, $g = n$, $m = 1$, $\sigma = T_a/g$)
   The first category represents the LDDoS attacks whose aggregate attack period is equally distributed among n flows. The attack frequency of the aggregate flow is intensified by n times, compared to the frequency of each attack flow.
2. Attack burst Width Intensification (AWI) LDDoS attack($n > 0$, $g = n$, $m = 1$, $\sigma = T_b$)
   The second category corresponds to the case when the aggregate burst width of an LDDoS attack is equally distributed among n flows. An attack burst of a flow is immediately followed by a burst from another flow. In this case, the attack burst width of the aggregate attack flow is intensified by $n$times.
3. Attack burst Rate Intensification (ARI) LDDoS attack($n > 0$, $g = 1$, $m = n$, $\sigma = 0$)
   The third category describes the LDDoS attacks in which n flows start at the same time, and the burst rate of the aggregate attack flow is intensified n times.
4. Mixed Intensification (MI) LDDoS attack
   ($n > 0$, $g > 1$, $m > 1$, $\sigma \geqslant 0$)

Table 1
Aggregate flow of LDDoS attacks.

LDDoS categories	Aggregate flow		
	T_a^+	T_b^+	R_b^+
AFI	T_a/n	T_b	R_b
AWI	T_a	$T_b \times n$	R_b
ARI	T_a	T_b	$R_b \times n$
MI	$(T_a/n, T_a)$	$(T_b, T_b \times n)$	$(R_b, R_b \times n)$

The last category can be considered as the combination of the previous three.

Let T_a, T_b, and R_b be the parameters for a single attack flow, and T_a^+, T_b^+, and R_b^+ be the parameters for the aggregate flow of an LDDoS attack. Table 1 demonstrates the relationship between T_a^+, T_b^+, R_b^+, and T_a, T_b, R_b.

3. The CPR-based approach

In this section, we first propose the metric of CPR and then describe our CPR-based approach to detect and filter LDDoS attack flows.

3.1. Congestion participation rate

A major difference between normal TCP flows and LDDoS flows is that normal TCP flows actively avoid network congestion due to TCP's congestion control

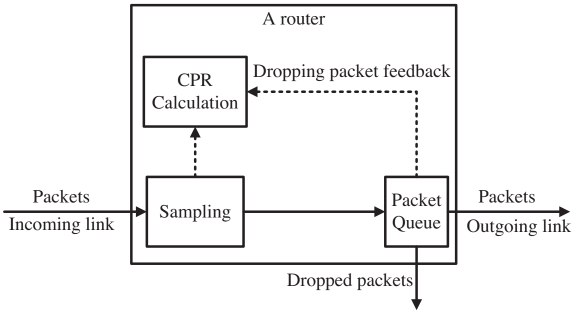


Fig. 4. Calculation Congestion Participation Rate (CPR) on a router.

mechanism, whereas LDDoS flows actively induce network congestion to degrade network performance.

Motivated by this difference, we propose Congestion Participation Rate (CPR) to distinguish between an LDDoS flow and a normal TCP flow. Consider a router through which different flows (normal TCP flows and LDDoS attack flows) pass (see Fig. 4), at time t, we sample the incoming link of the router for a duration d and count the number of packets from every flow F_i, denoted as $S_{i,t}$. At the same time, we monitor the packet queue in the router. If there is at least a packet dropped at the packet queue (because the queue is full) during d, we consider that the outgoing link of the router is congested at time t. After a series of sampling T, we define the CPR of flow F_i as

$$\theta_i = \sum_{t \in T*} S_{i,t} \Big/ \sum_{t \in T} S_{i,t} \tag{1}$$

where T^* is the set of sampling epochs when the outgoing link is congested. In other words, the θ_i is the ratio of the incoming packets in congestion to the total incoming packets from flow F_i. It is worth noting that the packets are sampled at the incoming link before they enter the packet queue or be dropped when the packet queue is full (due to the congestion of the outgoing link). Thus the packet number measured here is for the packets sent by a flow to the router. It is normally larger than the number of the packets from the flow that are forwarded by the router, as some of the packets may be dropped due to congestion.

Since LDDoS flows actively induce network congestion while normal TCP flows actively avoid network congestion, the CPR of an LDDoS flow is expected to be considerably bigger than that of a normal TCP flow.

3.2. The detecting and filtering approach

Fig. 5 shows the block diagram of our CPR-based approach for detecting and filtering LDDoS. This approach is expected to be deployed on a router where we want to detect and filter LDDoS attack flows. In the diagram, the flow size estimation module is a functional module that is already included in most Internet routers (for example Cisco NetFlow [10]). The CPR metric mainly works in the LDDoS Attack Detection module.

The router keeps CPR for every flow. A flow is considered as an LDDoS attack flow if its CPR is higher than a threshold τ. We investigate this threshold analytically in Section 3.3 and use experiments to demonstrate how to determine the threshold in practice in Section 4.4.

When an LDDoS flow is detected, we drop all packets from the detected flow until its CPR becomes smaller than the threshold τ. A normal TCP flow may be mistakenly interpreted as an attack flow if it is initiated and starts to send data while the network is congested. To mitigate it, after a flow is regarded as an LDDoS flow, we keep on measuring its CPR. If its CPR becomes smaller than the threshold, the flow will once again be considered as a normal flow and its packets will pass the router successfully. A normal TCP flow that was misclassified as an attack flow will reduce its CPR by sending data when the network is not congested.

In the extreme scenario, the LDDoS attack can throttle all normal TCP flows and the aggregate rate of the LDDoS flows is also very close to the bottleneck bandwidth of network. There is almost no packet will be dropped in the scenario and consequently no network congestion can be observed. When no network congestion can be observed from a scenario, the CPRs of both LDDoS flows and normal TCP flows all tend to be 0 according to the definition in (1). This is problematic for the CPR-based approach to distinguish LDDoS flows. To solve this problem, we turn on the Random Early Detection (RED) [11] queue management mechanism on the router where we install the CPR-based module. In the extreme scenario mentioned above, the RED mechanism actively drops packets from the pulsing LDDoS flows. Normally, RED drops packets in order to send congestion signals back to the sources. As will be covered in Section 4, the CPR-based approach with RED effectively detects LDDoS attacks in this scenario. We configure the RED mechanism to be based on packet count as opposed to packet bytes because the packet-count RED is more sensitive to the small packets sent in an LDDoS attack flow.

3.3. Bounds of the congestion participation rates

In this subsection, we theoretically analyze the average CPR difference between normal TCP flows and LDDoS flows, which is critical for our approach to identify attack flows. More precisely, we conduct a worst-case analysis by computing the minimum average CPR difference between normal TCP flows and LDDoS flows. It is worth noting that, this section does not aim to establish a complete analytical model for CPR bounds, but to analyze the relationship between CPR bounds and several network parameters that are directly measureable. As stated in Section 3.3.3, to get the exact value of those CPR bounds, one sill needs to experimentally measure several network parameters.

Since LDDoS flows tend to have a higher CPR than normal TCP flows, the minimum average CPR difference is obtained by subtracting the upper bound of normal TCP flows' average CPR from the lower bound of the average CPR of LDDoS flows. We obtain the theoretical equations for these two bounds in this subsection and verify the results through experiments in a real network in Section 5. Table 2 lists the notation used in our analysis.

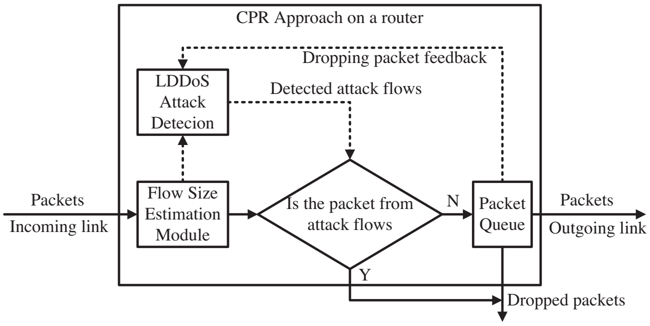


Fig. 5. Block diagram of the CPR-based approach.

Table 2
Notations and abbreviations for network traffic.

Notations	Definition
r_{TCP}	The average ratio of packets dropped to all incoming packets from normal TCP flows
r_{LDDoS}	The average ratio of packets dropped to all incoming packets from LDDoS flows
B	The maximum bandwidth of the outgoing link of a router (Mbps)
ρ_{TCP}	The average aggregate peak incoming rate (Mbps) of normal TCP flows
ρ_{LDDoS}	The average aggregate peak incoming rate (Mbps) of LDDoS flows
δ_{TCP}	The average aggregate packet-dropping rate (Mbps) of normal TCP flows
δ_{LDDoS}	The average aggregate packet-dropping rate (Mbps) of LDDoS flows
z	The average packet size
T^*	The packet-dropping period of time
T	The total time period

When a network's bandwidth is sufficiently high, packets are unlikely to be dropped in the network, and an LDDoS attack cannot throttle normal TCP traffic either. Therefore we focus on the scenario that a network's bandwidth is lower than users' demands, i.e., the network is relatively high loaded. This can be represented as $\rho_{TCP}+\rho_{LDDoS} > B$.

Consider the following scenarios. In a time period T, a set of normal TCP flows and LDDoS attack flows with average aggregate peak incoming rates ρ_{TCP} and ρ_{LDDoS} go through a router with outgoing bandwidth B. Since $\rho_{TCP}+\rho_{LDDoS} > B$, network congestion and packet dropping will happen. Let $T^* \subseteq T$ be the time when the outgoing link is in congestion and packets are dropped. We investigate the CPR through the ratio r of dropped packets to all incoming packets in this time period T. More specifically, we want to establish the relationship between CPR and r. Note that r is the packet drop ratio that can be measured directly from network traffic.

3.3.1. Upper bound of the CPR for normal TCP flows

Although many sophisticated TCP models have been proposed [12–15], we found that the simple model proposed in this subsection is sufficient for determining the upper bound of normal TCP flows' CPR.

We first consider the situation when there is no LDDoS attack flow, i.e., all flows are normal TCP flows. The situation when attack flows are present will be described next.

When there are only normal TCP flows, the traffic in time period T can be depicted in Fig. 6, which is obtained from our testbed experiment results. We assume that the packet dropping is all due to network congestion. When the outgoing link is congested, the buffers of routers are fully filled, and ρ_{TCP} is higher than B.That is the reason for packet dropping. Thus, we have (2).

$$\delta_{TCP} = \rho_{TCP} - B \tag{2}$$

From Fig. 6, one can see that the number of packets dropped in time period T is $(T^* \times \delta_{TCP})/z$, whereas the total number of incoming packets in T is $(T^* \times \rho_{TCP})/z+((T-T^*)\times B)/z$, which gives us r_{TCP} in (3).

Intuitionally, r_{TCP} is the ratio of the shaded area to the area enclosed by the Rate curve and the Time axis (from 0 to T) in Fig. 7, i.e., the area enclosed by the bold curves.

$$\begin{aligned} r_{TCP} &= \frac{T^* \times \delta_{TCP}/z}{T^* \times \rho_{TCP}/z + (T-T^*)\times B/z} \\ &= \frac{T^* \times \delta_{TCP}}{T^* \times \rho_{TCP} + (T-T^*)\times B} \end{aligned} \tag{3}$$

According to the definition of CPR in (1), we have the CPR of the aggregate TCP flows, θ_{TCP}, as

$$\begin{aligned} \theta_{TCP} &= \frac{T^* \times \rho_{TCP}/z}{T^* \times \rho_{TCP}/z + (T-T^*)\times B/z} \\ &= \frac{T^* \times \rho_{TCP}}{T^* \times \rho_{TCP} + (T-T^*)\times B} \end{aligned} \tag{4}$$

Fig. 8 graphically illustrates the calculation of θ_{TCP} in (4). θ_{TCP} is the proportion of the shaded area to the area enclosed by the bold curves.

Eqs. (3) and (4) yield the relationship between the CPR of the aggregate TCP flows, θ_{TCP}, and r_{TCP} in (5):

$$\theta_{TCP} = \begin{cases} \frac{\rho_{TCP}}{\delta_{TCP}} \times r_{TCP} & (\delta_{TCP} > 0) \\ 0 & (\delta_{TCP} = 0) \end{cases} \tag{5}$$

In (5), when $\delta_{TCP} = 0$, no packet dropping is observed. Consequently, $T^* = 0$ and $\theta_{TCP} = 0$.

From (2) and (5), we get (6):

$$\theta_{TCP} = \frac{\rho_{TCP}}{\rho_{TCP} - B} \times r_{TCP} = \frac{1}{1 - B/\rho_{TCP}} \times r_{TCP}(\rho_{TCP} > B) \ \ldots \tag{6}$$

From (6), we can see that the average CPR of normal TCP flows without attacks is determined by B/ρ_{TCP} and r_{TCP}, which in practice can be obtained from network traffic. In other words, once we obtain B, ρ_{TCP} and r_{TCP}, we can use (6) to get θ_{TCP}.In Section 5, the accuracy of (6) is validated by real network traffic.

When LDDoS attack flows are present with normal TCP flows in the network, normal TCP flows would be forced by the LDDoS flows to send packets only in the off-period of the attack (when the network is not congested) or may even stop sending. Thus the CPR of normal TCP flows in this case is smaller than the one in the first case as shown in (6). Therefore θ_{TCP} in (6) represents the upper bound of the CPR of normal TCP flows.

3.3.2. Lower bound of the CPR for LDDoS flows

The lower bound of the average CPR for the aggregate LDDoS flows should be achieved in the extreme scenarios described in Section 3.2, i.e., when the aggregate rate of attack flows is very close to the bottleneck bandwidth. Fig. 9 illustrates the scenarios where two aggregate bursts with rate ρ_{LDDoS} very close to bandwidth B were sent one after the other with an interval between them. The network is congested during two bursts T_1 and T_2. However, since ρ_{LDDoS} is close to B, the number of dropped packets may be very small if no Active Queue Management (e.g., RED) is being deployed. As previously described, our CPR-based approach employs RED to actively drop packets from the pulsing LDDoS flows. In Fig. 9, T_1^* and T_2^* represent the packet-dropping period (mainly due to the RED) in the two bursts. Note that in the figure, both T_1^* and T_2^* are represented as a single period for illustrative purposes. In fact, there may be multiple short periods distributed within each burst. However, this simplification does not affect our analysis below.

In this situation, we can get (7) and (8).

$$r_{LDDoS} = \frac{(T_1^* + T_2^*) \times \delta_{LDDoS}/z}{(T_1 + T_2) \times \rho_{LDDoS}/z} = \frac{(T_1^* + T_2^*) \times \delta_{LDDoS}}{(T_1 + T_2) \times \rho_{LDDoS}} \tag{7}$$

$$\theta_{LDDoS} = \frac{(T_1^* + T_2^*) \times \rho_{LDDoS}/z}{(T_1 + T_2) \times \rho_{LDDoS}/z} = \frac{(T_1^* + T_2^*)}{(T_1 + T_2)} \tag{8}$$

From (7) and (8), we can get (9).

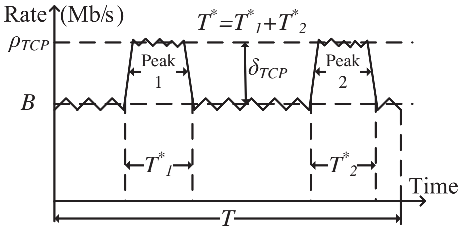


Fig. 6. Normal TCP traffic (when there is no attack).

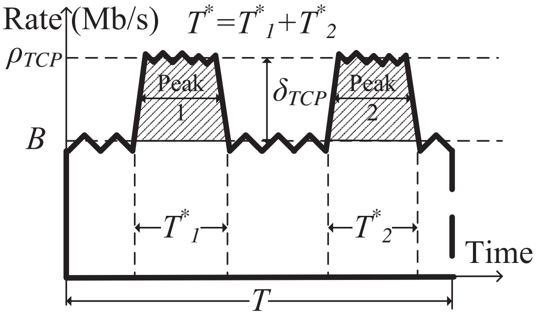


Fig. 7. Illustration of r_{TCP}.

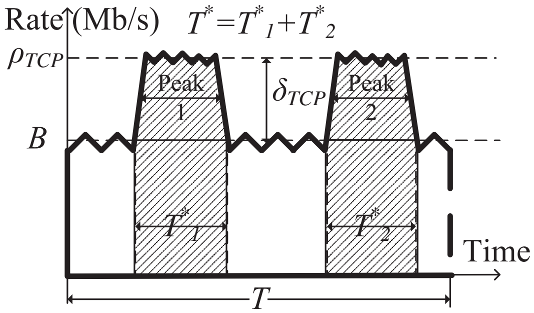


Fig. 8. Illustration of θ_{TCP}.

$$\theta_{LDDoS} = \begin{cases} \frac{\rho_{LDDoS}}{\delta_{LDDoS}} \times r_{LDDoS}(\delta_{LDDoS} > 0) \\ 0(\delta_{LDDoS} = 0) \end{cases} \tag{9}$$

In (9), when $\delta_{LDDoS} = 0$, no packet dropping is observed. Consequently, $T^* = 0$ and $\theta_{LDDoS} = 0$. If no RED is deployed, r_{LDDoS} and δ_{LDDoS} could drop to 0, resulting in the CPR for the LDDoS flow to be 0 as well. Under these conditions no attack flows will be detected. However, when we turn on RED on the router, packets will be dropped and attack flows will be detected. r_{LDDoS} and δ_{LDDoS} in (9) are mainly determined by the RED mechanism in practice, as will be described in the next section.

3.3.3. Minimum average CPR difference between TCP and LDDoS flows

The minimum average CPR difference is obtained by subtracting the upper bound of normal TCP flows' average CPR (θ_{TCP}) from the lower bound of the average CPR of LDDoS flows (θ_{LDDoS}). In order to calculate θ_{TCP} and θ_{LDDoS} based on (6) and (9), several parameters need to be measured from network traffic, including B, $\rho_{TCP}, r_{TCP}, \rho_{LDDoS}$, δ_{LDDoS}, and r_{LDDoS}.

Note that ρ_{TCP} and ρ_{LDDoS} , which respectively represents the average aggregate peak incoming rate of normal TCP flows and LDDoS flows, are normally higher than B – the bandwidth of the outgoing link of a router. They could be measured at a router by sampling packets from its incoming link before those packets enter the packet queue or be dropped when the packet queue is full (packets that

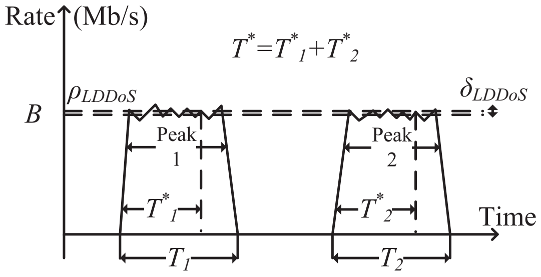


Fig. 9. The meticulously planned LDDoS attack traffic.

manage to enter the packet queue will be finally sent to the outgoing link of the router).

We measure these parameters in three test-bed experiments in Section 5 and show the calculation results – θ^e_{TCP} and θ^e_{LDDoS} – in Table 6. According to Table 6, the minimum-average-CPR differences between normal TCP and LDDoS flows (θ^e_{DIF}, which equals to $\theta^e_{LDDoS} - \theta^e_{TCP}$) are 62.05%, 69.49%, and 73.46% for those three test-bed experiments, resulting in an overall minimum-average-CPR difference of 68.3%.

4. Simulation experiments

In this section we present our experiment results obtained from ns-2 [16]. Section 4.1 tests the influence of the RED mechanism on the approach. Section 4.2 evaluates the performance of the CPR-based approach in the presence of different LDDoS attacks. Section 4.3 evaluates the performance of the approach in a challenging scenario – distinguishing LDDoS flows from short-lived HTTP flows. Section 4.4 evaluates the trade-off between the detection rate and the false positive rate of our CPR-based approach that provides a quantitative guideline to determine the CPR threshold τ in practice.

Dumbbell network topologies are commonly used in congestion control studies [17]. Fig. 10 shows the experimental topology, which consists of two routers (R_0, R_1), 30 users ($User_1 \cdots User_{30}$), 20 attackers ($Attacker_1 \cdots Attacker_{20}$), 30 servers ($Server_1 \cdots Server_{30}$), and a victim server (Victim Server). The link between two routers is the bottleneck link with a bandwidth of 5 Mbps and one-way propagation delay of 6 ms. All the other links have a bandwidth of 10Mbps and a one-way propagation delay of 2 ms. In this topology, $User_i$ communicates with $Server_i$ ($i = 1 \cdots 30$) using FTP (generated by *Application/FTP* using Newreno TCP in ns-2), and 20 attackers send UDP packets (generated by *Application/Traffic/CBR* in ns-2) to attack the Victim Server. The queue size of the bottleneck link is 50. A RED based on packet count is deployed at router R_0 on the queues of the bottleneck link. Other links use DropTail queues. A CPR-based detection module is installed at router R_0 where most normal TCP packets are dropped when an LDDoS attack is present. For comparison, we also install a module based on Cumulative Amplitude Spectrum (CAS) [18] at R_0, CAS uses Discrete Fourier Transform (DFT) to locate anomalies caused by LDDoS flows.

A set of simulations are conducted and outlined in each of the following subsections. Each simulation lasts for 240 s. LDDoS attack traffic begins at 120 s and continues until 220 s. In Sections 4.1, 4.2, and 4.3, normal TCP traffic (the FTP flows from User to Server) begins at 20 s and ends at 240 s. While in Section 4.4, it begins at a random time between 20 s and 240 s, and ends at 240 s. In this section, we use a 7-tuple $(n, g, m, \sigma, T_a, T_b, R_b)$ to describe the parameters of an LDDoS attack.

4.1. RED on our approach

We first explore the influence of the RED mechanism on normal TCP flows, LDDoS flows and their CPRs. We consider an AFI LDDoS attack with parameters ($n = 20$, $g = 20$, $m = 1$, $\sigma = 1$ s, $T_a = 20$ s, $Tb = 200$ ms, $R_b = 5$ Mbps). Here R_b is chosen to be close to the bandwidth of the bottleneck to represent the extreme situation to our CPR-based approach as described in Section 3.2.

Then using the same parameters we repeat the experiment two times. The first time we employ DropTail on the bottleneck queue at router R_0 and the second time we employ RED. The sampling frequency is $f = 1000$ Hz in both experiments and the results are shown in Table 3.

In Table 3, ρ_a and ρ_{na} represent the average rates of normal TCP flows when there is an attack and when there is no attack, respectively. r_{na} represents the ratio of dropped packets to all packets at router R_0 when there is no attack and r_a is the one when there is an attack. θ_{TCP} represents the average *CPR* of the normal TCP flows and θ_{LDDoS} is for the LDDoS flows.

The results in Table 3 indicate that when there is no attack RED has no obvious influence on the normal TCP flows (ρ_{na} and r_{na} are almost the same for DropTail and RED). However, when LDDoS attacks are present, RED significantly improves the throughput of normal TCP. Note that in this case, RED drops 25% packets, much more than the 7.6% from DropTail. In other words, when the aggregate rate of the LDDoS flows is very close to the bottleneck bandwidth of network, deploying RED is able to drop more attack packets while increasing the throughput of the normal TCP flows. Admittedly RED might also drop more legitimate packets than DropTail here, however it achieves a higher normal-TCP-flow throughput ρ_a than DropTail. The throughput ρ_a of normal TCP flows is considered to be more important than its drop ratio r_a when a network is under an LDDoS attack.

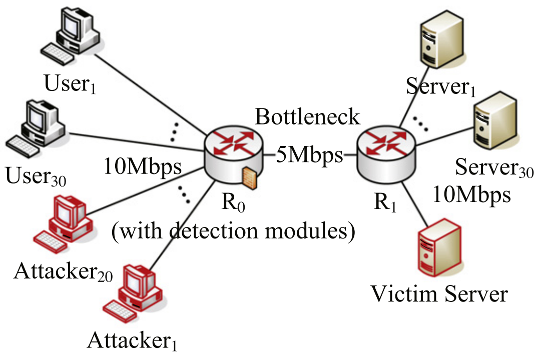


Fig. 10. Topology of the simulation network.

Table 3
RED vs DropTail.

AQM	ρ_{na}	r_{na}	ρ_a	r_a	θ_{TCP}	θ_{LDDoS}
DropTail	4.9 Mbps	10.6%	1.0 Mbps	7.6%	13.5%	14.4%
RED	4.9 Mbps	10.6%	3.1 Mbps	25.0%	12.5%	85.9%

Table 3 shows that under DropTail, the CPR-based approach may not be able to distinguish an LDDoS attack flow from a normal TCP flow since the two CPRs are close to one another. However, it is also clear that the CPR-based approach under RED is able to detect an attack flow since the two CPRs are significantly different under RED.

4.2. LDDoS attacks

In this subsection, we conduct experiments to compare the effectiveness of our proposed CPR-based approach and the CAS approach [18] on detecting LDDoS attack flows under different categories of LDDoS attacks. Parameters of the experiments are listed in Table 4, where for each of the three categories (AFI, AWI and ARI), we select a range of values for one variable and fix all the others. For example, for the AFI LDDoS attack, we vary the attack period T_a for each single attack flow from 20 s to 40 s. The italic values in the table list the parameters corresponding to the aggregate attack flow. They are calculated according to Table 1, which demonstrates the relationship between the parameters of each LDDoS flow (T_a, T_b, and R_b) and their aggregate flow (T_a^+, T_b^+, and R_b^+). The results for the MI LDDoS attack are not presented here and can be provided upon request, but this category is just the combination of other three and the results are quantitatively comparable.

Fig. 11 shows the results for the AFI LDDoS attack. Both the CPR-based approach and the CAS-based approach are quite effective for this category of the LDDoS attack. It is worth noting that the difference between normal TCP flows and LDDoS flows in CPR (around 0.6) is larger than that in CAS (around 0.4). We also observe that the average CPRs do not change too much as T_a increases. This is because the CPR is calculated based on a ratio (Eq. (1)). When the attack period T_a increases, the attackers tend to send fewer packets. However, the ratio of the incoming packets in congestion to the total incoming packets measured by a router may remain unchanged. This explains the similar observations in Figs. 12 and 13.

Fig. 12 shows the results for the AWI LDDoS attack. Our CPR-based approach is still effective – the difference in the average CPR for normal TCP flows and LDDoS flows is evident. However, as T_b decreases (from 10 ms to 0.1 ms), the average CAS of the attack flows under the CAS-based approach also decreases. When T_b is around 0.5 ms, the average CAS of the normal TCP flows and the LDDoS flows is about the same.

Fig. 13 shows the results for the ARI LDDoS attack. Similar to previous results, our CPR-based approach is still effective whereas the effectiveness of the CAS-based approach decreases as R_b decreases (from 0.25 Mbps to 0.01 Mbps).

These experimental results clearly demonstrate that our CPR-based approach works well for all three categories (AFI, AWI, and ARI) of LDDoS attacks whereas the CAS-based approach only works for AFI LDDoS. CAS distinguishes LDoS flows from normal TCP flows using their spectrum difference in low frequency band. It works well for small-scale LDoS attacks. However, an attacker can dramatically reduce the burst width T_b and burst rate R_b of each LDDoS flow by launching large-scale LDDoS attacks, including LDoS attacks with spoofing IP addresses. The spectrum difference in low frequency band between an LDoS flow and a normal TCP flow decreases when T_b and R_b of each LDDoS flow are reduced. For this reason DFT-based approaches, such as CAS, are not effective in detecting large-scale LDDoS attacks.

4.3. HTTP traffic

LDDoS flows can be short lived so that their traffic pattern is similar to that of normal short-lived flows like HTTP flows. Therefore, it is challenging to distinguish LDDoS flows from HTTP flows. In this subsection, we evaluate the performance of the CPR-based approach for distinguishing LDDoS flows from HTTP flows.

HTTP traffic in our experiment is generated by PackMime generator [19], that uses real Internet traces. The network topology is similar to that in Fig. 10, but with only 10 users (this is due to the limit of PackMime), 10 servers, and 10 attackers. Let R_c represent the average number of new connections started per second in PackMime. We investigate the relationship between R_c and the average *CPRs* of normal HTTP flows and LDDoS flows. The experimental results are shown in Fig. 14. One can see from the figure that the difference between the average CPR of normal HTTP flows and that of LDDoS flows increases as $R_c$increases. Thus it is not harder but easier for our CPR-based approach to distinguish LDDoS flows from normal HTTP flows when there are more normal HTTP flows.

4.4. Trade-off of detection rate and false positive rate

In this subsection, we conduct experiments based on MI LDDoS attacks to systematically understand the principle

Table 4
Parameters of LDDoS attacks experiments.

Categories	LDDoS attack				Single flow			Aggregate flow		
	n	g	m	σ	T_a (s)	T_b (ms)	R_b (Mbps)	T_a^+ (s)	T_b^+ (ms)	R_b^+ (Mbps)
AFI	20	20	1	$T_a/20$	[20,40]	200	5	*[1,2]*	*200*	*5*
AWI	20	20	1	T_b	1	[0.1,10]	5	*1*	*[2]*	*5*
ARI	20	1	20	0	1	200	[0.01,0.25]	*1*	*200*	*[0.2,5]*

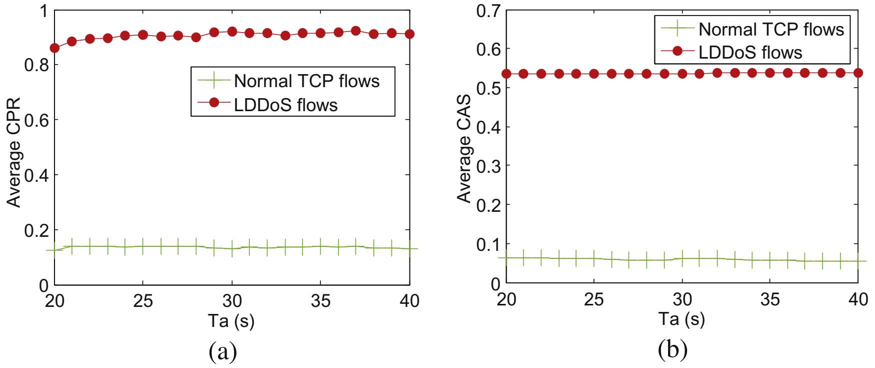


Fig. 11. AFI LDDoS attack experiment.

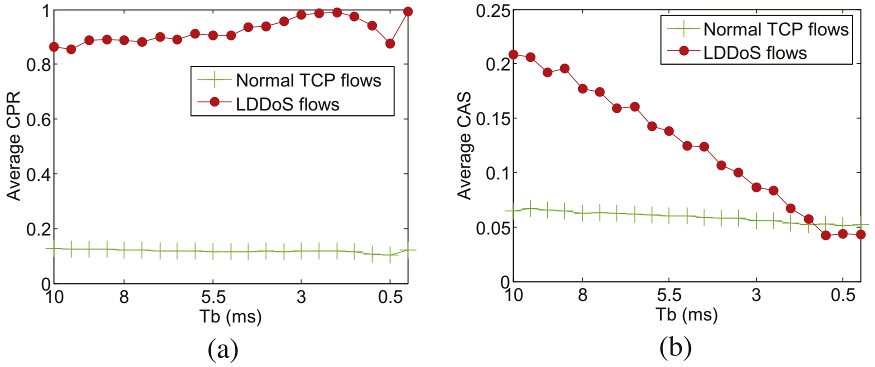


Fig. 12. AWI LDDoS attack experiment.

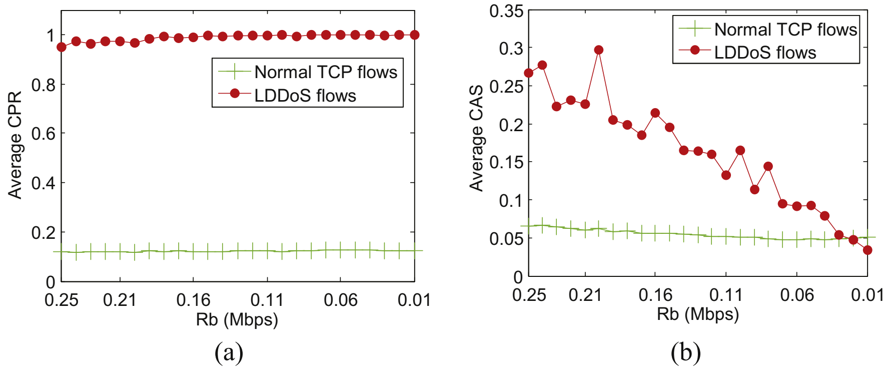


Fig. 13. ARI LDDoS attack experiment.

of our CPR-based approach. The parameters of the LDDoS attack are $(n = 20, g = 5, m = 4, T_g, T_a, T_b, R_b)$, where $T_a \in [0.5, 5]$ s, $T_b \in [25]$ ms, and $R_b \in [0.75, 1.5]$ Mbps. It took over 36 h to conduct hundreds of experiments that still use the dumbbell network shown in Fig. 10. In these experiments, each user starts to send normal TCP traffic at a random time between 20 s and 240 s, and stops sending at 240 s. This simulates users' random behavior on the Internet.

Fig. 15a depicts the probability distribution of CPR of normal TCP flows and LDDoS flows. Clearly, the CPRs of these two different flows are distributed on the two ends between 0 and 1, with a small overlap in the middle.

Fig. 15a also provides experimental guidance for choosing the CPR τ threshold in practice. From the figure, for any given CPR $x \in (0, 1)$, one can obtain the experimental likelihood (H_x) that the average CPR of LDDoS flows is higher than x, which is the ratio of the red area on the right side

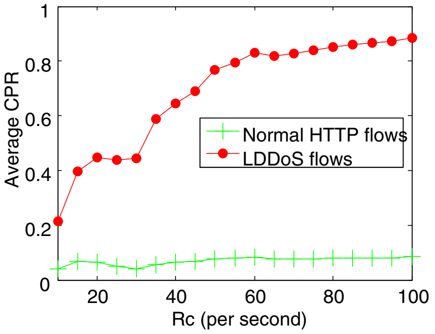


Fig. 14. LDDoS attacks with HTTP traffic.

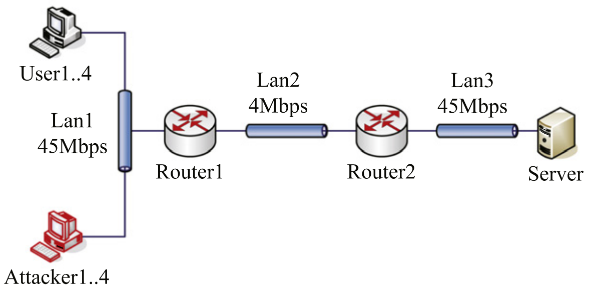


Fig. 16. Topology of the test-bed.

of x to the whole red area. Similarly, one can obtain the likelihood (L_x) that the average CPR of normal TCP flows is smaller than x. Based on this, one can choose the CPR threshold τ to be x to achieve detection rate H_x with a false positive rate $(1 - L_x)$. Obviously, a higher x corresponds to a lower H_x and a higher L_x. This trade-off is indicated in the Receiver Operating Characteristic (ROC) curve in Fig. 15b, where each point on the ROC curve corresponds to a CPR $x \in (0,1)$ in Fig. 15a. For example, when $x = 0.63$, we can achieve a 100% detection rate with a 1.625% false positive rate.

5. Real network and internet trace experiments

In addition to the simulation conducted using ns-2, we set up a test-bed shown in Fig. 16 to further evaluate the performance of the CPR-based approach. Entities in the test-bed are all PCs, whose function, OS and installed software are listed in Table 5. In the test-bed experiments, normal TCP flows include short-lived TCP (e.g., SSH) and long-lived TCP (e.g., FTP). We use these two kinds of TCP flows to represent the mixed TCP traffic on the real network. We modify the TFN2K [20], a well-known DDoS attack tool, to label its attack flows by the ID field in the IP header. We also add LDDoS attack function to the TFN2K and correct its header-checksum algorithm. We name the refined program as TFN2K4R (Tribe Flood Network 2000 For Research). TF2K4R is used to generate labeled LDDoS traffic. The labels are used to verify the detection rate of our CPR-based approach.

Table 5
Setup of test-bed machines.

Entities	Function	OS	Software
User1···4	Client	Debian 4	FTP, SCP
Attacker1···4	Attack	Debian 4	TFN2K4R
Router1 & 2	Packet forwarding Packet dumping	Debian 4	Tcpdump
Server	Server	Debian 4	vsftpd, ssh

In the experiments User1 and User2 use FTP (long-live TCP flows) to upload a 32 MB file to Server while User3 and User4 use SCP (short-lived TCP flows) to transfer 50 copies of a 0.5 MB file to Server. For the attack, we fix the first six parameters in the 7-tuple $(n, g, m, \sigma, T_a, T_b, R_b)$ and vary the last parameter R_b. The first six parameters are fixed as $n = 4$, $g = 1$, $m = 4$, $\sigma = 0$, $T_a = 1$ s, and $T_b = 200$ ms. The setting of these fixed parameters means that four attackers each simultaneously start an LDoS attack $(T_a = 1\text{ s},\ T_b = 200\text{ ms},\ R_b,\ s = 5\text{ s})$. A total of three experiments are conducted, each with a different R_b. Table 6 presents the average CPR for different flows, where θ_{FTP} is the average CPR for long-lived FTP flows (User1 and User2), θ_{SCP} is the average CPR for short-lived SCP flows (User3 and User4), and θ_{LDDoS} is the average *CPR* for LDDoS flows.θ^e_{TCP} is the estimated average CPR of normal TCP flows calculated using (6) and θ^e_{LDDoS} is the estimated average CPR of LDDoS flows calculated using (9). θ^e_{DIF} is the estimated average CPR difference between normal TCP flows and LDDoS flows, which equals to $\theta^e_{LDDoS} - \theta^e_{TCP}$.

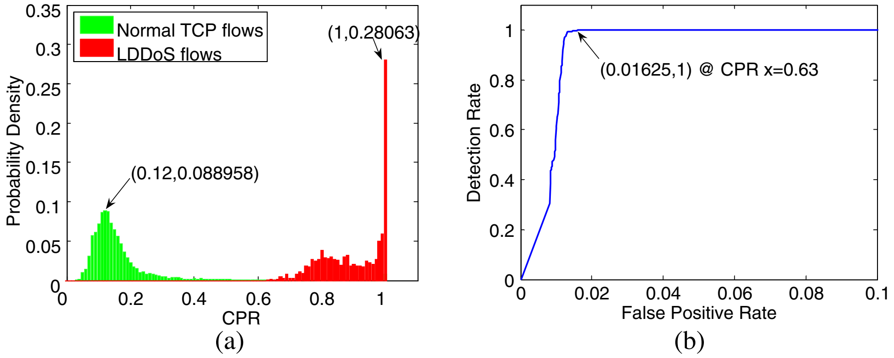


Fig. 15. Receiver Operating Characteristic (ROC) experiment.

Table 6
Analysis of real network experiments.

R_b (Mbps)	θ_{FTP} (%)	θ_{SCP} (%)	θ_{LDDoS} (%)	θ^e_{TCP} (%)	θ^e_{LDDoS} (%)	θ^e_{DIF} (%)
0.25	3.27	4.19	67.1	4.15	66.2	62.05
0.5	5.95	5.43	78.7	6.51	76.0	69.49
1	8.02	8.37	84.9	7.84	81.3	73.46

Table 6 shows a clear difference between the average CPR of LDDoS flows (θ_{LDDoS}) and normal TCP flows (θ_{FTP} and θ_{SCP}). It is also noticeable that the calculated θ^e_{TCP} and θ^e_{LDDoS} are quite close to the measured CPR (θ_{FTP}, θ_{SCP} and θ_{LDDoS}), which indicates that our analysis in Section 3.3 is reasonable.

We further use real Internet traffic to validate our CPR-based approach. The LBNL/ICSI Enterprise Trace [21] is collected at a medium-sized enterprise site – one of the most possible places to install our approach. We use (6) to estimate the average CPR of the TCP flows contained in the traces. The collected trace did not record dropped packets, thus we cannot calculate the drop-packet-ratio r_{TCP} needed in (6). Instead, we use the out-of-sequence packets [22] to approximate r_{TCP}. The average CPR of the normal TCP flows for the three traces selected from [21] are 0.93%, 0.77%, and 0.70%, respectively. This is because the traces were collected when the network was not congested (the ratio of out-of-sequence packet is very small) and their CPR values tend to be zero. Since currently there is no LDDoS attack trace publically available, using real attack traffic to validate our CPR-based approach remains one of our future work.

6. Discussion

Our CPR-based approach is an online algorithm that can be easily implemented in a router because its basic mechanism is simply counting packets, and then calculating the CPR based on Eq. (1), whose accuracy has been verified in the previous section. In this section we discuss practical issues related to the implementation of the CPR-based approach.

6.1. Flow table sizes

Compared to existing approaches on detecting LDDoS attacks [9,18,23–27], our CPR-based approach is capable of detecting whether a flow is an attack flow or a normal TCP flow. A flow table is required to maintain the CPR of all the flows passing through the router where the CPR-based approach is installed. Our CPR-based approach requires 19.44 MB memory to maintain 11,341,289 flows in an ISP trace with OC48 speed (conducted in.[28–30]) using a bloom filter calculator [31] with the probability of false positive of 0.001. However, the CPR-based module should be installed on routers to which the bottleneck link, to the potential victim, is connected. Such routers are unlikely to be ISP routers since they are unlikely to be bottleneck [32]. Consequently the number of flows our CPR-based approach needs to maintain will be considerably less than the one in the ISP trace leading to smaller size memory requirements. Additionally, as will be discussed in the next subsection, Multilevel Bloom Filter techniques have been proposed, which can be used to further reduce the flow table size.

Besides the flow table, our CPR-based approach also needs a flow size estimation, which can be achieved through existing solutions implemented at current routers such as Cisco NetFlow [10].

6.2. IP address spoofing

According to the way it generates spoofed source addresses, IP address spoofing can be classified into two types: fixed spoofing and random spoofing. Fixed spoofing generally does not cause a large-space overhead for our approach, as its source addresses are chosen from a predefined list [33]. While random spoofing would drastically increase the flow table size needed in our approach because a flow is defined based on IP addresses. There are already a number of solutions to tackle IP spoofing [34], such as Network Ingress Filtering [35]. However, since none of the solutions are widely deployed, we describe an alternative technique "Multilevel Bloom Filters" that can be integrated into our approach to mitigate the large-space-overhead problem caused by random spoofing.

Multilevel Bloom Filters (MBF) have been shown to be an effective approach for mitigating the large-space-overhead problem by providing a trade-off between the space requirement and the false positive rate [36]. The MBF technique is inspired by the success of the Stochastic Fair Blue (SFB) [36] algorithm, which is a scalable approach to record and update the state information of flows through Multilevel Bloom Filters. According to [36], the MBF effectively gives the algorithm N^L unique "buckets" using $L \times N$ number of bins (L levels with N bins in each level). That is, we can save $(1 - L \times N/N^L) \times 100\%$ of required buffer or memory space by using Multilevel Bloom Filters. For example, we can save up to $(1–2\times23/23^2) = 91.3\%$ of the required memory by using a MBF with $L = 2$ and $N = 23$. In our recent work.[37], we have employed MBF techniques and observe an efficient memory save using the approach with an affordable false positive rate.

6.3. UDP traffic

Our CPR-based approach has been demonstrated to work effectively in distinguishing normal TCP flows from LDDoS flows. Since UDP flows normally do not reduce their transmission rate when the network is congested, we believe that the current CPR-based approach will treat a UDP flow as an attack flow if the UDP flow behaves nonresponsively. Further investigation on differentiating UDP flows and attack flows, and the fairness issue will be investigated in future work.

6.4. Integration of RED

The motivation of employing the RED [11] queue management mechanism together with our approach is to improve the performance of our approach in the extreme scenario described in Section 3.2. The the RED [11] mechanism enable our CPR-based approach to still be effective in detecting LDDoS attack flows in that very extreme scenario. Although the RED mechanism is found to be notably vulnerable to LDoS attacks [37,38], it does not introduce vulnerabilities, because our approach detects and filters attack flows while simultaneously protecting RED from being exploited.

6.5. Adversarial analysis

Finally we briefly discuss how difficult it would be for an attacker to evade our CPR-based approach.

The various types of LDDoS attacks shown in Fig. 3 may adopt different detection evasion strategies while causing the same damage to normal flows. For example, AFI LDDoS attacks could enlarge the attack period T_a for each attack flow; AWI LDDoS attacks could narrow the attack burst width T_b of each attack flow; ARI LDDoS attacks could reduce the attack burst rate R_b for every attack flow and MI LDDoS attacks could employ one or several of the aforementioned strategies. We have evaluated the performance of the proposed approach under AFI, AWI, and ARI LDDoS attacks in Section 4.2 and against MI LDDoS attacks in Section 4.4. Experimental results show that the Congestion Participation Rate (CPR) is a reliable and robust metric for identifying LDDoS attack flows for all LDDoS attack categories listed in Fig. 3.

Generally, an attacker might want to lower the CPR value by sending fake pulses that do not cause network congestion. Although fake pulses may lower the CPR value of an LDDoS flow, as long as the flow still sends packets during congestion periods, it will be detected since it still has a higher CPR than a normal TCP flow.

The only way to completely evade our CPR-based detection is to send all the attack packets using the TCP congestion control mechanism, in other words, to make an LDDoS flow behave like a normal TCP flow. In this case, the LDDoS flow hardly achieves any obvious attack effects. To summarize, to the attacker, there is a trade-off between mounting an effective attack but being detected or evading our approach but loosing attack effectiveness.

7. Related work

LDoS attacks were proposed by Kuzmanovic [2] in 2003, which were also called shrew [2] attacks and Pulsing DoS (PDoS) attacks [9]. LDDoS attacks are LDoS attacks that are launched from hosts distributed on the Internet. Since LDoS was initially proposed in [2], a series of variants of LDoS attacks have been discussed, including:

- Reduction of Quality (RoQ) attacks [39] that exploit the performance vulnerability during a system's adaptation process. RoQ attacks could dramatically reduce the service quality of a network element, or deprive it of a large amount of its capacity, by only occupying a small fraction of its capacity [39].
- LDoS attacks targeting application servers (LoRDAS attacks) [40]. LoRDAS attacks can reduce the availability of an application server in a controlled manner by generating pulsing service request, using only low-rate traffic.

These LDoS variants are not directly addressable by the CPR-based approach proposed in this paper. Adopting the CPR-based approach to address these new variants is part of our future work.

Presently approaches to detect and filter LDoS attacks mainly consider two characteristics of LDoS attacks. One is the pulsing high rate characteristic and the other is the periodical characteristic.

Kuzmanovic [2] and Sarat [23] proposed approaches that explore the pulsing high rate characteristic of LDoS attacks. Active Queue Management (AQM) mechanisms (such as SRED [41]) were used to mitigate LDoS attacks. Their approaches were easy to deploy and effective at improving the performance of normal TCP flows in the presence of LDoS attacks. However, the adaptive mechanisms in AQM algorithms are also targets of LDoS attacks. RED-like algorithms [11,36,42], as typical representatives of AQM algorithms, have already been found to be considerably vulnerable to LDoS attacks [37,38].

Shevtekar [24] proposed an approach based on the traffic anomaly of all the expired flows to detect LDoS attacks, considering the pulsing (short-lived) characteristic of the LDoS attacks. This approach has the capability to detect LDoS attacks even when the source and destination IP addresses are spoofed. Unfortunately it only identifies the presence of attacks, and not the identity of the attack flows.

Sun's approach [25] based on Dynamic Time Warping (DTW) examined the periodical characteristic of LDoS attacks. This approach can detect LDoS attacks that employ variable attack pulsing periods. However, since this DTW-based approach used the similarity between real-time LDoS flows and the sampled LDoS flows, it is only effective for the *aggregate* flow of LDDoS attacks. In other words, it can only detect the presence of an LDDoS attack, but fails to identify whether a given flow is an LDDoS flow or not.

Discrete Fourier Transform (DFT)-based approaches proposed by Chen [18] and Wei [26] consider both the periodical characteristic and the pulsing high rate characteristic of LDoS attacks. They explore the difference between the traffic spectrum of attack flows and that of normal flows. DFT-based approaches are considered to be one of the most efficient approaches in detecting LDoS attacks. However, they have difficulties in identifying single flows in large-scale LDDoS attacks.

Luo [9] proposed an approach based on wavelet transform after considering the influence of LDoS attacks on the input TCP data traffic and output TCP ACK traffic. This approach considers both the attack flows' characteristics and their influence on network traffic. An extension to this approach is their Vanguard detection system that employs more metrics to detect various LDoS attacks [43]. These approaches are both limited by their ability to only detect the aggregate flow of an LDDoS attack.

The Shrew Attack Protection (SAP) mechanism.[44] mitigates the LDoS attack by giving priority to flows with high packet loss rate. While SAP can maintain high throughput for TCP flows under certain LDoS attacks and always prevent TCP sessions from closing, it has obvious performance degradation when the attack uses the ports protected by SAP.

The generalized entropy metric and the information distance metric were proposed in [45] to detect LDDoS attacks. This approach has advantages in terms of detection speed and false positive rate. The impracticality of this approach is rather unfortunate with successful implementation relying on gaining full control of all the routers in the network.

A mathematical model [46] and a defense technique [47] were proposed for LDoS attacks targeting application servers [40]. The model and defense technique are stimulating, but more experiments and analyses are needed to test their effectiveness for LDDoS attacks targeting TCP traffic [2] that are studied in this paper.

In a large-scale LDDoS attack, the attack period of each single attack flow could be very long. Its peak rate could be very small, and the pulsing period could be very short. Thus the average rate of every single attack flow could be very low, even lower than a normal flow in a large-scale LDDoS attack. This is the main reason that existing approaches can only detect the aggregate flow of an LDDoS attack, rather than a single attack flow. The Congestion Participation Rate (CPR)-based approach proposed in this paper can detect single LDDoS attack flows.

In addition to existing LDoS detection approaches, flow-level Active Queue Management (AQM) mechanisms are also relevant to our approach as they detect and limit unresponsive [48] flows, including FRED [49], RED-PD [50], SFB [51], CHOKe[52], etc. However, most existing flow-level AQMs aim to maintain fairness among traffic flows instead of detecting LDDoS flows. Moreover, some of them (RED-PD[50] and SFB[51]) have already been identified as being vulnerable to LDoS attacks [37,38].

8. Conclusions

In this paper, we have proposed an effective and efficient approach to detect and filter TCP-targeted LDDoS attacks [2,9] based on a novel metric – Congestion Participation Rate (CPR). The CPR-based approach can achieve per-flow-level detection of LDDoS attacks. We have analytically expressed the upper bound of the average CPR for normal TCP flows and the lower bound of the average CPR for LDDoS flows, using several network parameters that are directly measureable. We have implemented the CPR-based approach and conducted comprehensive experiments in both ns-2 and test-bed. The experiment results have demonstrated that, compared to the existing Discrete Fourier Transform (DFT)-based approach, the CPR-based approach is effective for all LDDoS attacks considered, while the DFT-based approach is effective for a limited set of LDDoS attack types.

We should note here that the CPR-based approach requires the router where it is deployed to turn on the Random Early Detection (RED) [11] queue management mechanism, to work properly on an extreme scenario described in Section 3.2. The RED mechanism is already supported by most existing routers (such as the WRED in Cisco routers).

In the future work, we will reduce the required memory size by implementing MBF techniques in the CPR-based approach. Another promising direction we hope to achieve is the deep integration of our CPR-based approach with the RED mechanism. Lastly, the CPR metric proposed in this paper could be applied to detect a number of variants of LDDoS attacks, such as LDDoS attacks against application servers [40] and LDDoS attacks against 3G/WiMax wireless networks [53].

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (Nos. 61070198, 60970034, 61170287, and 60903040) and the Engineering and Physical Sciences Research Council of UK (No. EP/G037264/1).

References

[1] C. Douligeris, A. Mitrokotsa, DDoS attacks and defense mechanisms: classification and state-of-the-art, Computer Networks 44 (2004) 643–666.

[2] A. Kuzmanovic, E.W. Knightly, Low-rate TCP-targeted denial of service attacks – (the shrew vs. the mice and elephants), in: ACM SIGCOMM, Karlsruhe, Germany, 2003, pp. 75–86.

[3] G. Loukas, G. Oke, Protection against denial of service attacks: a survey, Computer Journal 53 (2010) 1020–1037.

[4] M. Li, An approach to reliably identifying signs of DDOS flood attacks based on LRD traffic pattern recognition, Computers & Security 23 (2004) 549–558.

[5] R.K.C. Chang, Defending against flooding-based distributed denial-of-service attacks: a tutorial, IEEE Communications Magazine 40 (2002) 42–51.

[6] T. Peng, C. Leckie, K. Ramamohanarao, Survey of network-based defense mechanisms countering the DoS and DDoS problems, ACM Computing Surveys 39 (2007).

[7] P. Owezarski, On the impact of DoS attacks on internet traffic characteristics and QoS, in: Proceedings of the International Conference on Computer Communications and Networks (ICCCN), 2005, pp. 269–274.

[8] M. Delio, New Breed of Attack Zombies Lurk, 2011.

[9] X. Luo, R.K.C. Chang, On a new class of pulsing denial-of-service attacks and the defense, in: Network and Distributed System Security Symposium (NDSS), San Diego, CA, 2005, pp. 2–5.

[10] CiscoSystems, NetFlow Services Solutions Guide, 2007. <http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.pdf>.

[11] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (1993) 397–413.

[12] J. Padhye, V. Firoiu, D.F. Towsley, J.F. Kurose, Modeling TCP Reno performance: a simple model and its empirical validation, IEEE/ACM Transactions on Networking 8 (2000) 133–145.

[13] I. Yeom, A.L.N. Reddy, Modeling TCP behavior in a differentiated services network, IEEE/ACM Transactions on Networking 9 (2001) 31–46.

[14] N. Cardwell, S. Savage, T. Anderson, Modeling TCP latency, in: IEEE INFOCOM 2000.

[15] M. Allman, W.M. Eddy, S. Ostermann, Estimating loss rates with TCP, SIGMETRICS Perform, Evaluation Review 31 (2003) 12–24.

[16] S. McCanne, S. Floyd, The Network Simulator – ns-2, 2008. <http://www.isi.edu/nsnam/ns/>.

[17] A. Shevtekar, N. Ansari, Do low rate DoS attacks affect QoS sensitive VoIP traffic? in: IEEE International Conference on Communications (ICC), 2006, pp. 2153–2158.

[18] Y. Chen, K. Hwang, Collaborative detection and filtering of shrew DDoS attacks using spectral analysis, Journal of Parallel and Distributed Computing 66 (2006) 1137–1151.

[19] J. Cao, W.S. Cleveland, Y. Gao, K. Jeffay, F.D. Smith, M. Weigle, Stochastic models for generating synthetic HTTP source traffic, in: IEEE INFOCOM, 2004, pp. 1546–1557.

[20] CERT, Advisory CA-1999-17 Denial-of-Service Tools, 2000. <http://www.cert.org/advisories/CA-1999-17.html>.

[21] V. Paxson, R. Pang, M. Allman, M. Bennett, J. Lee, B. Tierney, LBNL/ICSI Enterprise Tracing Project (collection), 2007. <http://imdc.datcat.org/collection/1-0132-C=LBNL%2FICSI+Enterprise+Tracing+Project>.

[22] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, C. Diot, Pocket-level traffic measurements from the sprint IP backbone, IEEE Network Magazine 17 (2003) 6–16.

[23] S. Sarat, A. Terzis, On the effect of router buffer sizes on low-rate denial of service attacks, in: International Conference on Computer Communications and Networks (ICCCN), San Diego, CA, 2005, pp. 281–286.

[24] A. Shevtekar, N. Ansari, A router-based technique to mitigate reduction of quality (RoQ) attacks, Computer Networks 52 (2008) 957–970.

[25] H.B. Sun, J.C.S. Lui, D.K.Y. Yau, Defending against low-rate TCP attacks: Dynamic detection and protection, in: IEEE International Conference on Network Protocols (ICNP), Berlin, GERMANY, 2004, pp. 196–205.

[26] W. Wei, Y.B. Dong, D.M. Lu, G. Jin, H.L. Lao, A Novel mechanism to defend against low-rate denial-of-service attacks, in: S.Z.D.D.C.H.T.B.W.F.Y. Mehrotra (Ed.), IEEE International Conference on Intelligence and Security Informatics (ISI), San Diego, CA, 2006, pp. 261–271.

[27] A. Kuzmanovic, E.W. Knightly, Low-rate TCP-targeted denial of service attacks and counter strategies, IEEE/ACM Transactions on Networking 14 (2006) 683–696.

[28] A. Kumar, M. Sung, J. Xu, J. Wang, Data streaming algorithms for efficient and accurate estimation of flow size distribution, in: ACM SIGMETRICS, 2004, pp. 177–188.

[29] C. Cranor, T. Johnson, O. Spataschek, V. Shkapenyuk, Gigascope: a stream database for network applications, in: SIGMOD, 2003, pp. 627–651.

[30] B. Grot, W. Mangione-Smith, Good memories: enhancing memory performance for precise flow tracking, in: ANCHOR, 2005.

[31] Thomas, Bloomfilter Calculator, 2009. <http://hur.st/bloomfilter>.

[32] A. Akella, S. Seshan, A. Shaikh, An empirical evaluation of wide-area Internet bottlenecks, in: ACM SIGCOMM Conference on Internet Measurement (IMC), Miami Beach, FL, USA, 2003, pp. 101–114.

[33] C. Wei, Y. Dit-Yan, Defending against TCP SYN flooding attacks under different types of IP spoofing, in: Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL), 2006, pp. 38–38.

[34] T. Ehrenkranz, J. Li, On the state of IP spoofing defense, ACM Transactions on Internet Technology 9 (2009).

[35] P. Ferguson, D. Senie, RFC2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, 2000.

[36] F. Wu-Chang, D.D. Kandlur, D. Saha, K.G. Shin, Stochastic fair blue: a queue management algorithm for enforcing fairness, in: IEEE INFOCOM, 2001, pp. 1520–1529.

[37] C. Zhang, J. Yin, Z. Cai, W. Chen, RRED: robust RED algorithm to counter low-rate denial-of-service attacks, IEEE Communications Letters 14 (2010) 489–491.

[38] X.P. Luo, R.K.C. Chang, E.W.W. Chan, Performance analysis of TCP/AQM under denial-of-service attacks, in: IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Atlanta, GA, 2005, pp. 97–104.

[39] M. Guirguis, A. Bestavros, I. Matta, Exploiting the transients of adaptation for RoQ attacks on Internet resources, in: IEEE International Conference on Network Protocols (ICNP), Berlin, Germany, 2004, pp. 184–195.

[40] G. Macia-Fernandez, J.E. Diaz-Verdejo, P. Garcia-Teodoro, Evaluation of a low-rate DoS attack against iterative servers, Computer Networks 51 (2007) 1013–1030.

[41] T.J. Ott, T.V. Lakshman, L. Wong, SRED: stabilized RED, in: IEEE INFOCOM, 1999.

[42] S.S. Kunniyur, R. Srikant, An adaptive virtual queue (AVQ) algorithm for active queue management, IEEE/ACM Transactions on Networking 12 (2004) 286–299.

[43] X. Luo, E.W.W. Chan, R.K.C. Chang, Detecting pulsing denial-of-service attacks with nondeterministic attack intervals, Eurasip Journal on Advances in Signal Processing (2009).

[44] C.-W. Chang, S. Lee, B. Lin, J. Wang, The taming of the shrew: mitigating low-rate TCP-targeted attack, IEEE Transactions On Network Service Management 7 (2010).

[45] Y. Xiang, K. Li, W. Zhou, Low-rate DDoS attacks detection and traceback by using new information metrics, IEEE Transactions on Information Forensics and Security 6 (2011) 426–437.

[46] G. Macia-Fernandez, J. Diaz-Verdejo, P. Garcia-Teodoro, Mathematical model for low-rate DoS attacks against application servers, IEEE Transactions on Information Forensics and Security 4 (2009) 519–529.

[47] G. Macia-Fernandez, R.A. Rodriguez-Gomez, J.E. Diaz-Verdejo, Defense techniques for low-rate DoS attacks against application servers, Computer Networks 54 (2010) 2711–2727.

[48] S. Floyd, K. Fall, Promoting the use of end-to-end congestion control in the Internet, IEEE/ACM Transactions on Networking 7 (1999) 458–472.

[49] D. Lin, R. Morris, Dynamics of random early detection, SIGCOMM Computer Communication Review 27 (1997) 127–137.

[50] R. Mahajan, S. Floyd, D. Wetherall, Controlling high-bandwidth flows at the congested router, in: Proceedings of IEEE International Conference on Network Protocols (ICNP), 2001, pp. 192–201.

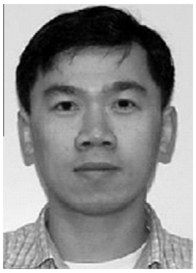
[51] W.-c. Feng, K.G. Shin, D.D. Kandlur, D. Saha, The blue active queue management algorithms, IEEE/ACM Transactions on Networking 10 (2002) 513–528.

[52] R. Pan, B. Prabhakar, K. Psounis, CHOKe – a stateless active queue management scheme for approximating fair bandwidth allocation, in: Proceedings of Annual IEEE International Conference on Computer Communications (INFOCOM), 2000, pp. 942–951.

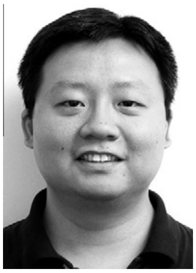
[53] P.P.C. Lee, T. Bu, T. Woo, On the detection of signaling DoS attacks on 3G/WiMax wireless networks, Computer Networks 53 (2009) 2601–2616.

Changwang Zhang received the B.S. and M.S. degrees in computer science from National University of Defense Technology, Changsha, China, in 2007 and 2009, respectively, and is currently pursuing the Ph.D. degree at Security Science Doctoral Research Training Centre, University College London, UK. His research interests include network security, network protocol design and analysis.

Zhiping Cai received his received the B.S., M.S. and Ph.D. degrees in computer science from National University of Defense Technology, Changsha, China, in 1996, 2002 and 2005, respectively. His research interests involve information security and network virtualization. He is a full associate professor of computer science in the National University of Defense Technology. He is a member of the IEEE.



Weifeng Chen received his Ph.D. from University of Massachusetts at Amherst, MS from Chinese Academy of Sciences and BS from Beijing University, all in computer science. His research interests include network security, privacy and protocol design. He is currently an Assistant Professor in the Department of Math and Computer Science at California University of Pennsylvania.



Xiapu Luo received his Ph.D from the Hong Kong Polytechnic University in 2007, and earned his MS and BS from Wuhan University, China, in 1999 and 2002, respectively. He is currently a research fellow in the computing department of the Hong Kong Polytechnic University after spending 2 years at the Georgia Institute of Technology as a postdoctoral fellow. His research interests include information security and network measurement.

Jianping Yin received his M.S. degree and Ph.D. degree in Computer Science from the National University of Defense Technology, China, in 1986 and 1990, respectively. His research interests involve information security, artificial intelligence, pattern recognition, and algorithm design. He is a full professor of computer science in the National University of Defense Technology.