Optimizing the Pulsing Denial-of-Service Attacks

Xiapu Luo and Rocky K. C. Chang
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong, SAR, China
{csxluo|csrchang}@comp.polyu.edu.hk

Abstract

In this paper we consider how to optimize a new generation of pulsing denial-of-service (PDoS) attacks from the attackers' points of views. The PDoS attacks are 'smarter' than the traditional attacks in several aspects. The most obvious one is that they require fewer attack packets to cause a similar damage. Another is that the PDoS attacks can be tuned to achieve different effects. This paper concentrates on the attack tuning part. In particular, we consider two conflicting goals involved in launching a PDoS attack: (1) maximizing the throughput degradation and (2) minimizing the risk of being detected. To address this problem, we first analyze the TCP throughput and quasi-global synchronization phenomenon caused by the PDoS attack. We then propose a family of objective functions to incorporate the two conflicting goals, and obtain the optimal attack settings. To validate the analytical results, we have carried out extensive experiments using both ns-2 simulation and a test-bed. The overall experimental results match well with the analytical results.

1. Introduction

According to the CSI/FBI Computer Crime and Security Survey in 2004 [12], the Denial of Service (DoS) attacks on network infrastructure and applications are considered to be the most damaging computer crime. The conventional DoS attacks flood a victim with a large number of useless packets in order to deplete victim's bandwidth or system resources, e.g., reflector DoS attack [18], SYN flooding [9]. Their main purpose is to degrade or completely block the legitimate usage of the resources as long as possible. On the defense side, the unusually large amount of attack traffic exposes themselves to the existing detection algorithms [9, 19]. Moreover, since attackers usually customize the attack packets, e.g. spoofed source address, unusual TCP flags, etc, various feature-based detection mechanisms have been proposed [3, 11, 17].

On the other hand, new DoS attacks and defense mechanisms have been proposed recently [4, 16]. For example, a new generation of DoS attack, coined as the *Pulsing DoS attacks* (PDoS), has been proposed and analyzed in [13]. By exploiting the TCP congestion control mechanism, a PDoS attacker can cause severe degradation in TCP performance by sending out continuous, short bursts of attack packets to a router. As a result, the TCP connections traversing the router are forced to frequently enter the timeout state (`TO`) or fast retransmit/fast recovery (`FR`) state.

There are three main differences between PDoS attacks and the traditional flooding-based DoS attacks. First, by adjusting the attack parameters, the PDoS attacker can cause different levels of damage, ranging from degradation-of-service to absolute denial-of-service. Second, since the average traffic attack rate of a PDoS attack is much smaller than the flooding-based attacks, it can evade the detection methods designed for flooding-based attacks [19]. Third, the number of attack packets required by a PDoS attack is small enough that the attacker can customize them with correct values in order to elude the feature-based detection methods [3, 11, 17].

Two classes of PDoS attacks have been identified in [13]: timeout-based attack and AIMD-based attack. In this paper we consider only the AIMD-based attack (AIMD stands for additive-increase, multiplicative-decrease), because it offers more flexibility for an attacker to control the attack's impact. The main contribution of this paper is to propose a model for optimizing the AIMD-based PDoS attack. Through analytical modelling and optimization, we have shown that the PDoS attack is indeed very versatile and can trade-off the level of damage for the risk of being detected.

1.1 Related work

The research on PDoS attacks began with the *shrew attack*, which can be considered as a timeout-based PDoS attack [10, 13]. A *shrew* attacker attempts to constrain a TCP

sender to the TO state by dispatching attack pulses whenever the sender retransmits lost packets after a timeout period. To defend against the timeout-based PDoS attack, it is proposed to randomize the timeout value in [7]. However, this method cannot defend the AIMD-based attack, because the attack's timing does not rely on the TCP timeout values. Another method to detecting timeout-based attacks uses dynamic time wrapping to isolate and discover the rectangular attack pulses [8]. However, this method would be rendered ineffective when the duration of attack pulse is shorter than the sampling period.

On another front, a reduction-of-quality attack (RoQ) has been proposed which also targets at TCP flows going through a router [15]. The RoQ attack forces the active queue management (AQM) scheme employed in a router to go into the transient state, and then increases packet loss rate by sending periodic attack pulses. The analysis in [15] mainly considers the RED-like AQM and the effect of the transient state on the TCP throughput.

The rest of the paper is organized as follows. In section 2, we first review the AIMD-based PDoS attack and the relevant results reported in [13]. Besides, we present new results on the TCP throughput under a PDoS attack and identify the *quasi-global synchronization* caused by the attack. In section 3, we first present a model to capture a PDoS attacker's intention in terms of the damage and the risk of being detected. From there, we formulate an optimization problem to obtain the best attack parameters. In section 4, we present the experimental results obtained from ns-2 simulation and a test-bed. The results are generally in good match with the analytical results. Section 5 finally concludes this paper with some pointers to future work.

2. An analysis of the AIMD-based PDoS attack

2.1. A review of the PDoS attack

A PDoS attacker sends out intermittent attack pulses to induce a sequence of *false congestion signals* to victim TCP senders, so that their congestion windows (cwnd) are consistently constrained to a low value. We formally model the sequence of attack pulses using $\mathbb{A}(T_{extent}(n), R_{attack}(n), T_{space}(n), N)$, where

- $T_{extent}(n), n = 1, 2, \ldots, N$, is the width of the nth attack pulse.
- $R_{attack}(n), n = 1, 2, \ldots, N$, is the sending rate of the nth attack pulse in *bps* (bits per second).
- $T_{space}(n), n = 1, 2, \ldots, N-1$, is the time between the end of the nth attack pulse and the beginning of the $(n+1)$th attack pulse. If $T_{space}(n) = 0, \forall n$, the PDoS attack is the same as a flooding-based attack.
- N is the total number of pulses sent during an attack.

To simplify the following analysis, we assume that the attack pulses are identical, i.e., $T_{extent} = T_{extent}(n)\ \forall n$ and $R_{attack} = R_{attack}(n)\ \forall n$.

A TCP sender utilizes an AIMD algorithm to adjust its cwnd when it enters the FR state or congestion avoidance state [14]. Although TCP is the prime target of PDoS attacks, it is useful to examine more general AIMD algorithms, because many TCP-friendly protocols also use similar algorithms [23]. We therefore consider a general AIMD algorithm $AIMD(a, b),\ a > 0,\ 1 > b > 0$, based on which a sender will decrease its cwnd from W to $b \times W$ whenever it enters the FR state, and will increase its cwnd from W to $W + a$ MSS (Maximum Segment Size) per round-trip time (RTT) until receiving another congestion signal. TCP Tahoe, TCP Reno, and TCP New Reno all use $AIMD(1, 0.5)$. Moreover, many TCP implementations do not send an ACK for every received packet. Instead, they send a delayed ACK after receiving d consecutive full-size packets, where d is typically equal to 2 [14]. In this case the sender's cwnd is only increased by $\frac{a}{d}$ MSS per RTT.

In an AIMD-based attack, the attacker attempts to cause a victim TCP sender to frequently enter the FR state. If each attack pulse induces some packet losses in a TCP connection, but a sufficient number of duplicate ACKs can still be received by the TCP sender, the cwnd will drop by $(1-b)\%$. After that, the cwnd will increase by a MSS every RTT [14]. Since it will take at least $\frac{(1-b)d}{a}W$ number of RTTs to restore the cwnd back to W after a decrease from W to bW, the cwnd will decrease to a low value (degradation-of-service attack) after periodic packet losses caused by the attack pulses. Moreover, when the cwnd is dropped to a certain level, there may not be enough duplicate ACKs to trigger the fast recovery process. Thus, the AIMD-based attack can also cause frequent timeouts.

In this paper we consider only AIMD-based attack with a fixed attack period, i.e. $T_{space} = T_{space}(n), \forall n$, which is depicted in Fig. 1. We also denote the attack period by $T_{AIMD} = T_{space} + T_{extent}$. It has been proved that under such an attack the sender's cwnd will be converged to W_C [13], which is given by

$$W_C = \frac{a}{(1-b) \times d} \times \frac{T_{AIMD}}{RTT}. \quad (1)$$

2.2. TCP's throughput under AIMD-based PDoS attacks

Proposition 1 gives the throughput of a victim TCP connection under a PDoS attack.

Proposition 1. *The throughput of a victim TCP connection under an AIMD-based PDoS attack with a period of*

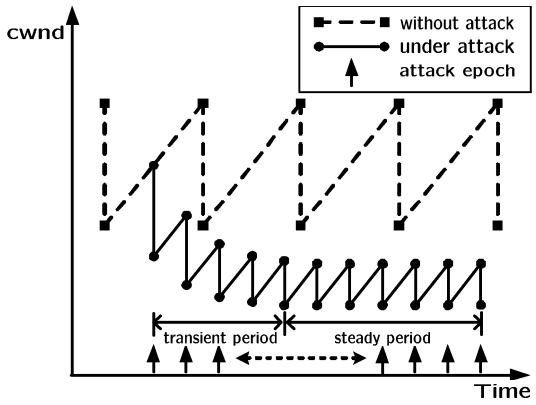


Figure 1. An AIMD-based attack with fixed attack periods.

T_{AIMD} is given by

$$\Psi_{attack} = \{ \sum_{n=1}^{N_{attack}-1} (bW_n + \frac{a}{2d}\frac{T_{AIMD}}{RTT})\frac{T_{AIMD}}{RTT} + \frac{a(1+b)}{2d(1-b)}(\frac{T_{AIMD}}{RTT})^2(N - N_{attack})\}S_{packet}, \quad (2)$$

where

- *$W_n, n = 1, \ldots, N$, is the* `cwnd` *value of the victim TCP connection* just before *the nth attack epoch.*
- *RTT is the round-trip time of the TCP connection.*
- *N_{attack} is the minimum number of attack pulses required to reduce* `cwnd` *from W_1 to W_C.*
- *S_{packet} is the packet size in bytes.*

Proof. We divide the whole TCP process into 2 distinct phases as shown in Fig. 1. The first phase is the transient period, which starts from the beginning of the attack and ends when the `cwnd` converges to W_C. During this period, the attacker will send N_{attack} attack pulses and there are $N_{attack} - 1$ free-of-attack intervals for TCP sender to transmit packets. During the interval between the ith and the $(i+1)$th attack epoch ($1 \leq i < N_{attack}$), the TCP sender can send $(bW_i + \frac{a}{2d}\frac{T_{AIMD}}{RTT})\frac{T_{AIMD}}{RTT}$ packets. Therefore, the first item within the curly brackets in Eq. (2) gives the number of packets sent during the transient state.

The second phase, which is referred to as the steady period, follows immediately after the transient phase. In this phase, the `cwnd` exhibits a periodic sawtooth pattern. There are a total of $N - N_{attack}$ such periods, each of which begins after the ith attack epoch and ends before the $(i+1)$th attack epoch ($N_{attack} \leq i < N$). The number of packets transmitted during each period is $(bW_C + W_C)\frac{T_{AIMD}}{2RTT} = \frac{a(1+b)}{2d(1-b)}(\frac{T_{AIMD}}{RTT})^2$. Therefore, the second item within the curly brackets in Eq. (2) gives the number of packets sent during the steady period. □

From Eq. (1) and Eq. (2), we can see that if the attacker can restrict the `cwnd` to a very small value, e.g. $W_C \ll W_1$, the throughput of TCP will, consequently, be throttled.

2.3. The quasi-global synchronization phenomenon

An interesting phenomenon caused by the PDoS attack is that the incoming traffic exhibits a periodic fluctuation, as illustrated in Fig. 2. In the absence of any attack, a *global synchronization* may occur when multiple TCP flows share the same bottleneck link and experience packet loss almost simultaneously [24, 6]. Similar synchronization phenomenon can also be caused by a PDoS attack which induces packet losses in the victim TCP connections simultaneously. Although this *quasi-global synchronization* phenomenon resembles that under the nonattack situation, their periods are different in that the former one is dictated by the attack parameters, while the latter one by the capacity of the bottleneck.

The peaks of the incoming traffic rate, which consists of attack pulses and legitimate TCP packets, are usually of high-rate and short-duration [10, 8, 13], whose length is determined by T_{extent}. The valleys of the traffic rate are due to the congestion control algorithm of the affected TCP flows. Whenever an attack pulse arrives at the router, its instantaneous high volume traffic will fill the queue and induce packet drops. Depending on the volume of attack packets and the duration of congestion period, some TCP flows may timeout while others may enter the `FR` state. Of course, some TCP flows may survive the attack without experiencing any packet loss. Therefore, there is still TCP traffic between two consecutive attack pulses. These fluctuations have a severe impact on the TCP performance, e.g. decrease in throughput and increase in jitter, etc.

To visualize this quasi-global synchronization phenomenon, we have conducted both ns-2 simulation [1] and test-bed experiments, and the results are shown in Fig. 3(a) and Fig. 3(b), respectively. In order to display the results clearly, the incoming traffic has been first normalized so that the mean value is zero and then transformed into a piecewise aggregate approximation [5].

Fig. 3(a) captures a one-minute snapshot of the incoming traffic in the ns-2 simulation, including the packets from 24 victims TCP flows and those belonging to a PDoS attack with $T_{extent} = 50ms$, $T_{space} = 1950ms$, $R_{attack} = 100Mps$. Not only can we observe the anticipated fluctuations but also its period. For example, there are 30 pinnacles evenly distributed within a duration of 60 seconds, implying a periodic signal with a period of $60/30 = 2s$. This

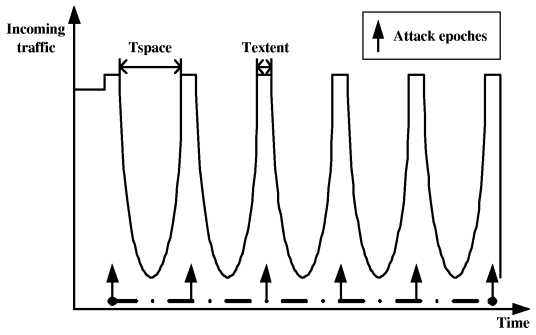
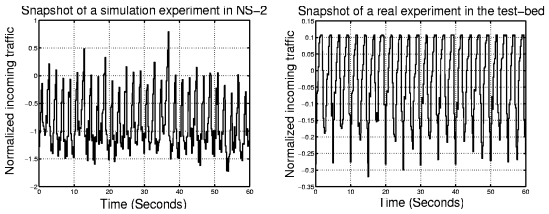


Figure 2. The periodic pattern of the incoming traffic during a PDoS attack.



(a) The ns-2 simulation results

(b) The test-bed experimentation results

Figure 3. The quasi-global synchronization phenomenon caused by a PDoS attack.

period is in fact equal to the period of the PDoS attack, i.e., $T_{AIMD} = 2s$.

Fig. 3(b) displays a one-minute snapshot of the incoming traffic in the test-bed experiment, consisting of packets from 15 victims TCP flows and those belonging to a PDoS attack with $T_{extent} = 100ms$, $T_{space} = 2400ms$, $R_{attack} = 50Mbps$. Fig. 3(b) exhibits the quasi-global synchronization phenomena in which 24 pinnacles are evenly spaced. The period of the incoming traffic is $60/24 = 2.5s$, which is also equal to the period of the attack pulses, i.e., $T_{AIMD} = 2.5s$.

3. How to launch a smarter PDoS attack?

As mentioned earlier, a very distinguishing feature of the PDoS attack is that its parameters can be tuned to achieve different attack objectives. For example, an attacker can choose to inflict a certain level of damage to the victim TCP connections and yet to evade the attack detection in place. The smartness of the PDoS attacks therefore lie on the flexible choices of the attack parameters which is the primary focus of this section.

First of all, since the primary objective of a PDoS attack is to cause throughput degradation, we use $\Gamma \in (0, 1)$ to measures the throughput degradation in the midst of an attack, normalized by the throughput without the attack.

$$\Gamma = 1 - \frac{\Psi_{attack}}{\Psi_{normal}}, \tag{3}$$

where $0 < \Psi_{attack} < \Psi_{normal}$. Ψ_{normal} is the TCP throughput in the absence of attacks, while Ψ_{attack} is the TCP throughput under an attack. When the attack is severe enough, Γ approaches to 1.

On the other hand, a PDoS attacker may also want to evade detection schemes that are based on the surveillance of the network traffic for anomalous patterns. For the purpose of modelling the attacker's preference in this aspect, we define an average attack rate normalized by R_{bottle}, the capacity of bottleneck in *bps*, by

$$\gamma = \frac{R_{attack} T_{extent}}{R_{bottle} T_{AIMD}}. \tag{4}$$

In the analysis, we consider $\gamma \in (0, 1)$, because for $\gamma \geq 1$ the PDoS attack would become the traditional flooding-based attack which does not attempt to evade attack detection.

Since DoS attacks can be detected based on the drastic increase in the traffic rate, we use $(1 - \gamma)^\kappa$, $\kappa > 0$, to measure an attacker's risk preference. When $\kappa > 1$, the attacker can be considered as *risk-averse*. That is, the attacker becomes less willing to take the risk of being exposed as the attack rate increases. When $0 < \kappa < 1$, the attacker is considered as *risk-loving*, which means that the attacker is more eager to cause more damage than to the concealment of the attack. To be complete, the attacker is considered as *risk-neutral* when $\kappa = 1$.

Fig. 4 depicts $(1 - \gamma)^\kappa$ as a function of γ for the three cases. The rate of the increase in the slopes of the curves differentiates the three kinds of an attacker's behavior in terms of the risk preference. Furthermore, there are two interesting limiting cases. For $\lim_{\kappa \to 0}(1 - \gamma)^\kappa = 1$, the attacker is entirely unconcerned about the risk of being detected, and the traditional flooding-based attack is a good example in this category. For $\lim_{\kappa \to \infty}(1 - \gamma)^\kappa = 0$, the attacker's decision is totally dominated by the risk of being detected to the extent that he would not even launch an attack.

Now we can ready to combine the two metrics for characterizing the damage of a PDoS attack and the attacker's risk preference into an attack gain, denoted by G_{attack}. Therefore, for any given finite value of κ, an attacker can optimize the attack by maximizing G_{attack}. The optimization problem formulation is presented next.

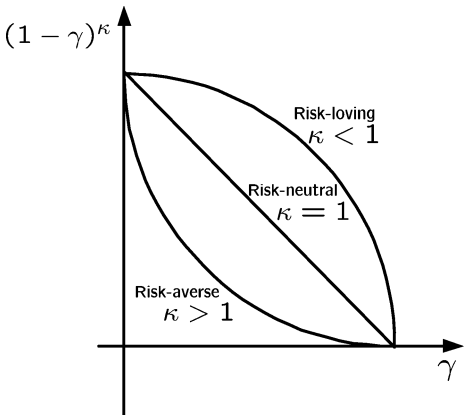


Figure 4. Three kinds of attacker's behavior modelled by $(1-\gamma)^\kappa$.

$$G_{attack} = \Gamma(1-\gamma)^\kappa = \left(1 - \frac{\Psi_{attack}}{\Psi_{normal}}\right) \times \left(1 - \frac{R_{attack}T_{extent}}{R_{bottle}T_{AIMD}}\right)^\kappa. \quad (5)$$

3.1. A PDoS attack optimization problem

In the following we formulate the PDoS attack problem as a nonlinear optimization problem in which the only constraint is to ensure that $0 < \gamma < 1$ for the reasons mentioned earlier.

$$\begin{cases} maximize & G_{attack} \\ subject\ to & 0 < \gamma < 1. \end{cases} \quad (6)$$

Let $T_{space} = \mu T_{extent}$, where $\mu > 0$ is the reciprocal of PDoS's duty cycle. $C_{attack} = \frac{R_{attack}}{R_{bottle}}$ is the ratio of each pulse's sending rate to the bandwidth of bottleneck. Then we can rewrite γ as

$$\gamma = \frac{C_{attack}}{1+\mu}. \quad (7)$$

In the next two lemmas we present the analytical expressions for Ψ_{attack} and Ψ_{normal}, from which we can obtain a computable expression for Γ. Moreover, we can transform the optimization problem.

Lemma 1. *Since TCP flows will make full use of the bottleneck bandwidth when there is no PDoS attack [22], Ψ_{normal} can be approximated by*

$$\Psi_{normal} = R_{bottle}(N-1)T_{AIMD}/8,\ R_{bottle} > 0. \quad (8)$$

Proof. According to the TCP congestion control mechanism, a TCP sender will increase its `cwnd` every RTT until it experiences a packet loss. In other words, the throughput of the TCP flows will increase whenever there is additional bandwidth to transfer packets. As a result, their aggregated throughout will be approximately equal to the capacity of network [22, 6]. Since the total period of a PDoS attack with N pulses is $(N-1)T_{AIMD}$ and the R_{bottle} is in *bps*, the throughput in *bytes* is given in Eq. (8). □

Lemma 2. *The aggregated throughput of N_{flow} TCP connections under a PDoS attack $\mathbb{A}(T_{extent}, R_{attack}, T_{space}, N)$ can be approximated by*

$$\Psi_{attack} = \frac{a(1+b)T_{AIMD}^2 S_{packet}}{2d(1-b)}(N-1)\sum_{i=1}^{N_{flow}} \frac{1}{RTT_i^2}. \quad (9)$$

Proof. It has been shown that the `cwnd` of a typical TCP ($AIMD(1, 0.5)$) can be brought to the converged value by using fewer than 10 attack pulses [13]. Therefore the period of transient state will be very short. To simplify the following analysis, we use W_C in Eq. (1) to approximate the W_n during the transient state. By substituting $W_n = W_C$ and Eq. (1) into Eq. (2) and summing up the throughput of all victim TCP flows during the $(N-1)$ free-of-attack intervals, we obtain Eq. (9). □

Proposition 2. *Under a PDoS attack, the normalized throughput degradation Γ is given by*

$$\Gamma = 1 - \frac{\Psi_{attack}}{\Psi_{normal}} = 1 - \frac{C_\Psi}{\gamma}, \quad (10)$$

where

$$C_\Psi = \frac{4a(1+b)T_{extent}S_{packet}C_{attack}}{(1-b)dR_{bottle}}\sum_{i=1}^{N_{flow}} \frac{1}{RTT_i^2}. \quad (11)$$

Proof. By substituting Eq. (8) and Eq. (9) into Eq. (3) and some algebraic simplification. □

Note that since $\Gamma \in (0,1)$, we have $0 < \frac{C_\Psi}{\gamma} < 1$ and therefore $C_\Psi < \gamma$. Moreover, since $\gamma \in (0,1)$, we have $0 < C_\Psi < 1$. Thus, the optimization problem in Eq. (6) becomes

$$\begin{cases} maximize & \left(1 - \frac{C_\Psi}{\gamma}\right)(1-\gamma)^\kappa \\ subject\ to & 0 < C_\Psi < \gamma < 1 \end{cases} \quad (12)$$

3.2. Optimized PDoS attack parameters

In this section we first present the solution to the optimization problem in Eq. (12). From there we can immediately obtain 3 corollaries regarding the optimal values of γ for the three types of PDoS attackers. After that, we present the optimal value of μ, which enables the attacker to decide the length of the attack period.

Proposition 3. *The solution to the optimization problem stated in Eq. (12) is given by*

$$\gamma_{-}^{*} = \frac{C_\Psi(1-\kappa) - \sqrt{C_\Psi^2(1-\kappa)^2 + 4\kappa C_\Psi}}{-2\kappa}. \quad (13)$$

Proof. We first obtain two roots to $\frac{\partial G_{attack}}{\partial \gamma} = 0$:

$$\gamma_{\pm}^{*} = \frac{C_\Psi(1-\kappa) \pm \sqrt{C_\Psi^2(1-\kappa)^2 + 4\kappa C_\Psi}}{-2\kappa}. \quad (14)$$

It is clear that γ_{+}^{*} is not a feasible solution, because its value is less than zero. On the other hand, γ_{-}^{*} is a feasible solution based on the following three results.

- $\gamma_{-}^{*} > 0$: This can be proved by observing that $\sqrt{C_\Psi^2(1-\kappa)^2 + 4\kappa C_\Psi} > C_\Psi(1-\kappa)$ and putting this into γ_{-}^{*}.
- $\gamma_{-}^{*} < 1$: We prove this by contradiction by assuming that $\gamma_{-}^{*} \geq 1$. Then we have $\sqrt{C_\Psi^2(1-\kappa)^2 + 4\kappa C_\Psi} - C_\Psi(1-\kappa) \geq 2\kappa$. After some simplification, the inequality is reduced to $C_\Psi \geq \kappa + C_\Psi(1-\kappa) \Rightarrow C_\Psi \geq 1$, which contradicts the first constraint in Eq. (12). Thus, $\gamma_{-}^{*} < 1$ in order to satisfy the first constraint.
- $\gamma_{-}^{*} > C_\Psi$: We prove this also by contradiction by assuming that $\gamma_{-}^{*} \leq C_\Psi$. Thus, we get $\sqrt{C_\Psi^2(1-\kappa)^2 + 4\kappa C_\Psi} - C_\Psi(1-\kappa) \leq 2\kappa C_\Psi \Rightarrow 1 \leq C_\Psi$, which also contradicts the first constraint in Eq. (12). Thus, $\gamma_{-}^{*} > C_\Psi$ in order to satisfy the first constraint.

Now we can prove that γ_{-}^{*} is the only solution to Eq. (12) by observing that $\frac{\partial G_{attack}}{\partial \gamma}$ is a continuous function and

$$Sign\left(\frac{\partial G_{attack}}{\partial \gamma}\right) = \begin{cases} > 0 & \text{if} \quad \gamma \in (C_\Psi, \gamma_{-}^{*}) \\ = 0 & \text{if} \quad \gamma = \gamma_{-}^{*} \\ < 0 & \text{if} \quad \gamma \in (\gamma_{-}^{*}, 1). \end{cases} \quad (15)$$

□

Corollary 1. *For a risk-averse attacker ($\kappa > 1$), the optimal attack parameter is given by $\gamma_{-}^{*} = C_\Psi$ as κ goes to infinity, i.e., $\lim_{\kappa\to\infty} \gamma_{-}^{*} = C_\Psi$.*

Proof. According to L'Hospital's rule, $\lim_{\kappa\to\infty} \gamma_{-}^{*} = \lim_{\kappa\to\infty} \frac{\partial(C_\Psi(1-\kappa) - \sqrt{C_\Psi^2(1-\kappa)^2 + 4\kappa C_\Psi})/\partial\kappa}{\partial(-2\kappa)/\partial\kappa} = C_\Psi$. □

Corollary 2. *For a risk-loving attacker ($\kappa < 1$), the optimal attack parameter is given by $\gamma_{-}^{*} = 1$ as κ goes to 0, i.e. $\lim_{\kappa\to 0} \gamma_{-}^{*} = 1$.*

Proof. According to L'Hospital's rule, $\lim_{\kappa\to 0} \gamma_{-}^{*} = \lim_{\kappa\to 0} \frac{\partial(C_\Psi(1-\kappa) - \sqrt{C_\Psi^2(1-\kappa)^2 + 4\kappa C_\Psi})/\partial\kappa}{\partial(-2\kappa)/\partial\kappa} = 1$. □

Corollary 3. *For a risk-neutral attacker ($\kappa = 1$), the optimal attack parameter is given by $\gamma_{-}^{*} = \sqrt{C_\Psi}$.*

Proof. Substituting $\kappa = 1$ into Eq. (13). □

According to the analysis conducted in section 2, if each attack pulse would cause packets losses in different TCP flows, then the remaining TCP throughput is mainly determined by the attack period $T_{AIMD} = (1+\mu)T_{extent}$ as shown in Eq. (2). In other words, when R_{attack} and T_{extent} are given, we can determine the optimal value of μ that achieves the tradeoff between the damage and the risk preference.

Proposition 4. *The optimal $\mu_{optimal}$ is given by*

$$\mu_{optimal} = \frac{-2\kappa C_{attack}}{C_\Psi(1-\kappa) - \sqrt{C_\Psi^2(1-\kappa)^2 + 4\kappa C_\Psi}} - 1. \quad (16)$$

Proof. By substituting Eq. (13) into Eq. (7). □

Corollary 4. *For a risk-neutral attacker,*

$$\mu_{optimal} = \sqrt{\frac{C_{attack}}{T_{extent} C_{victim}}} - 1, \quad (17)$$

where

$$C_{victim} = \frac{4a(1+b)S_{packet}}{(1-b)dR_{bottle}} \sum_{i=1}^{N_{flow}} \frac{1}{RTT_i^2}. \quad (18)$$

Proof. By substituting $\kappa = 1$ into Eq.(16). □

4. Performance Evaluation

We have conducted extensive experiments on both ns-2 2.28 simulation environment [1] and a test-bed to verify the optimal solution and investigate the effect of different parameters on the results. The simulation settings and results are given in section 4.1, while the results obtained from test-bed are given in section 4.2.

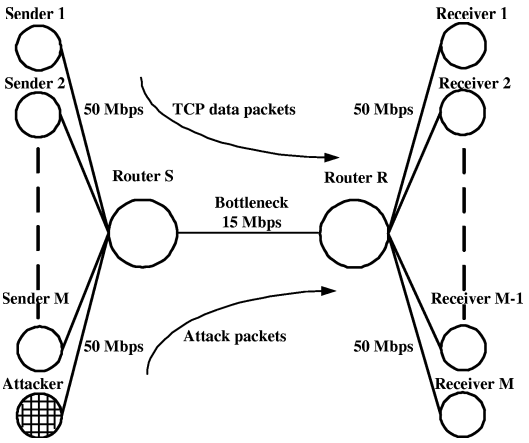


Figure 5. The topology of the simulation model.

4.1. Simulation experiments and results

The network topology, shown in Fig. 5, consists of M pairs of TCP senders and receivers. All links except the bottleneck between router S and R are $50Mbps$. The two routers are connected through a link of $15Mbps$ with RED. The TCP connections are based on TCP New Reno [21] and their RTTs range from 20ms to 460ms. We use the simulation scripts provided by [10].

Figs. 6, 7, 8 and 9 show the results for different number of TCP flows (15, 25, 35, and 45) under PDoS attacks with $R_{attack} = 25Mbps$, $30Mbps$, $35Mbps$ and $40Mbps$. The analytical results are presented by lines in the figures, while the simulation results are represented by symbols.

4.1.1 Normal-gain, under-gain, and over-gain attacks

Figs. 6-9 show that all analytical results correctly predict the trends of the attack gains. However, the values may not be in exact match, because of the complex interplay between the attack pulses and the queue management mechanisms. For example, under certain attacks, the TCP sender may suffer from more degradation in throughput when it enters the `TO` state instead of the `FR` state. Therefore, we classify the attacks into three categories according to the discrepancies between the experimental and analytical results.

The *normal-gain* attacks refer to those cases in which the simulation and analytical results are in close agreement, such as the case of $R_{attack} = 25Mbps, T_{extent} = 100ms$ in Fig. 6 and the case of $R_{attack} = 35Mbps, T_{extent} = 75ms$ in Fig. 8. The PDoS attack with such parameter settings can effectively constrain the TCP throughput to a low value by causing them to frequently enter the `FR` state.

The *under-gain* attacks refer to those cases in which the analytical results over-estimate the actual attack gains, such as the cases when $T_{extent} = 50ms$ in Figs. 6-8. The cause for the discrepancy is due to the fact the attack rate is not high enough to affect all the TCP flows. Moreover, we can observe that the longer the duration of each attack pulse is, the more severe the PDoS attack inflicts on the normal TCP flows. This is because more legitimate packets will be dropped under such attacks when the attack packets occupy more buffer and/or use more computational resources from the router.

The *over-gain* attacks refer to those cases in which the analytical results under-estimate the actual attack gains, such as the case of $R_{attack} = 40Mbps, T_{extent} = 100ms$ in Fig. 9. This is because such attacks can force more TCP flows to enter the `TO` state instead of the `FR` state due to its high attack rate. Therefore, the analytical results consistently under-estimate the extent of the throughput reduction.

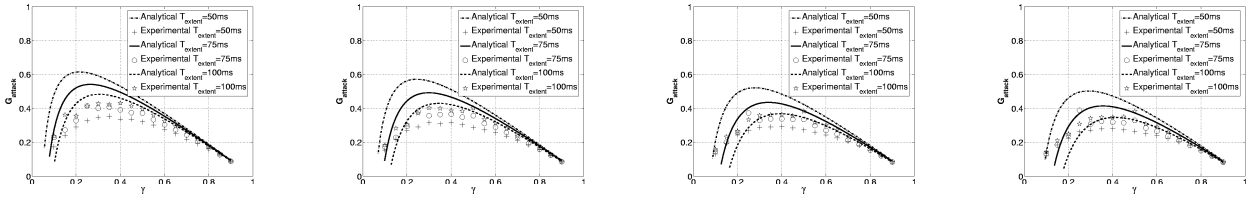
4.1.2 Maximization points

Figs. 6-9 show that for the normal-gain and over-gain attacks, most of the experimental results generally match very well with the analytical results in the maximization points. The exceptions are due to the shrew attacks that will be discussed in the following subsection. However, for the under-gain attacks, they do not match as well, because the number of attack packets is too small to block the bottleneck and therefore not all the legitimate TCP flows are affected by the attack.

The figures also show that the experimental results located on the right-hand side of the maximization points are closer to the analytical results than those on the left-hand side of the maximization points. This is because when γ increases, more attack packets will be sent in each pulse, which will take up more resources in the bottleneck. Therefore, more legitimate TCP flows will be affected by the attack and consequently their throughput will be decreased as predicted from the analysis.

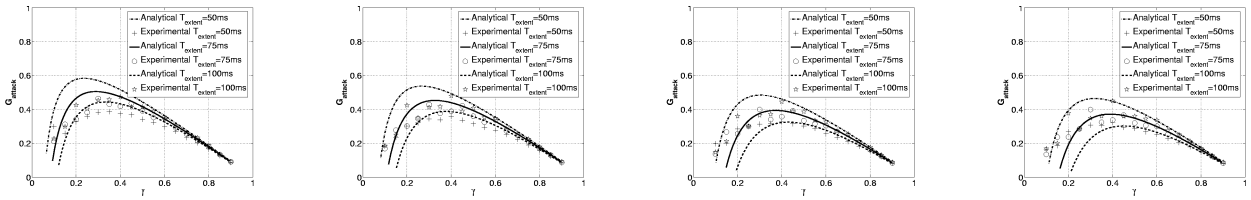
4.1.3 Shrew attacks

According to the analysis in [13], the AIMD-based attack and the timeout-based attack share the similar attack scheme but they exploit different aspects of the TCP congestion control mechanism. There are in fact some attack cases that correspond to the shrew attacks. That is, if a PDoS attack's T_{AIMD} is approximately equal to $\frac{minRTO}{n}$, $n \in [1, minRTO]$, where $minRTO$ is TCP's minimum retransmission timeout value, then the attack may constrain the TCP sender to the `TO` state, instead of the `FR` state assumed by the analytical model. As a result, the actual throughput degradation will be grossly under-estimated by the analysis. Even if some TCP flows may survive these



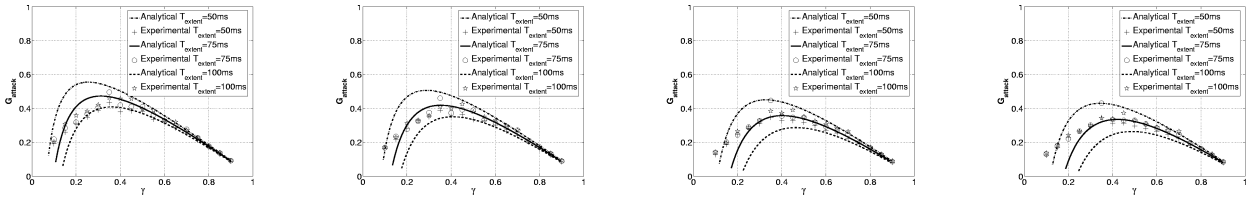
(a) 15 TCP flows. (b) 25 TCP flows. (c) 35 TCP flows. (d) 45 TCP flows.

Figure 6. The analytical results and experimental results under PDoS attacks with $R_{attack} = 25Mbps$.



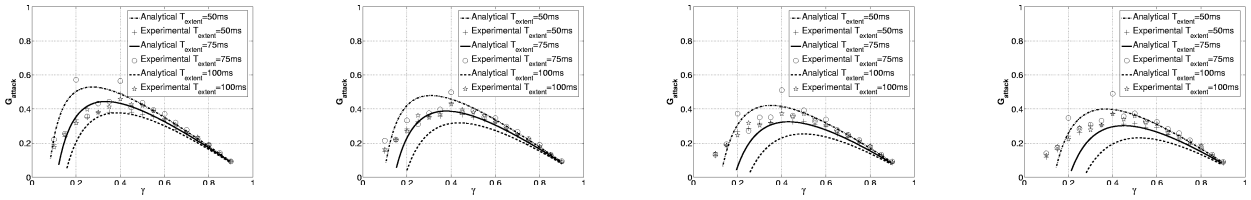
(a) 15 TCP flows. (b) 25 TCP flows. (c) 35 TCP flows. (d) 45 TCP flows.

Figure 7. The analytical results and experimental results under PDoS attacks with $R_{attack} = 30Mbps$.



(a) 15 TCP flows. (b) 25 TCP flows. (c) 35 TCP flows. (d) 45 TCP flows.

Figure 8. The analytical results and experimental results under PDoS attacks with $R_{attack} = 35Mbps$.



(a) 15 TCP flows. (b) 25 TCP flows. (c) 35 TCP flows. (d) 45 TCP flows.

Figure 9. The analytical results and experimental results under PDoS attacks with $R_{attack} = 40Mbps$.

timeout-based attack because of their large RTTs [10], they will s

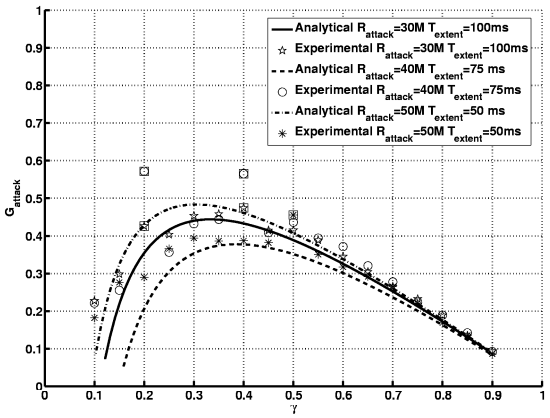


Figure 10. Relationship between the PDoS attacks and the shrew attacks.

We show some of the cases in Fig. 10 and we use □ to mark them. For the normal-gain attack with $R_{attack} = 30Mbps$ and $T_{extent} = 100ms$, the shrew-attack points are $T_{AIMD} = 500ms, 1000ms$ in which the attack gains are much higher than what are anticipated by the analysis. For the over-gain attack with $R_{attack} = 40Mbps$ and $T_{extent} = 75ms$, all points except the 2 shrew-attack points match well the trend given by the analysis. For the under-gain attack with $R_{attack} = 50Mbps$ and $T_{extent} = 50ms$, once again the shrew-attack point of $T_{AIMD} = 1000/3ms$ gives a higher attack gain than the analytical result.

4.2. Test-bed experiments and results

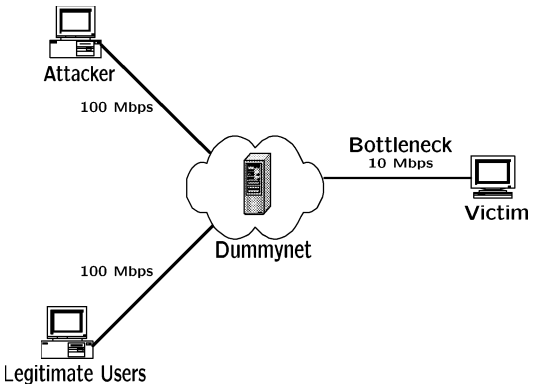


Figure 11. The topology of the test-bed.

The topology for the test-bed experiments is shown in Fig. 11. We use `Dummynet` [20] to simulate the network by setting the bottleneck to $10Mbps$ and introducing $150ms$ delay. We use `Iperf` [2] to generate legitimate TCP flows. In the absence of attacks, the victim TCP flows will occupy all the bandwidth. The link between `Dummynet` and the victim is 10 Mbps, whereas the links connecting the legitimate users and the attacker to the `Dummynet` are 100 Mbps. In this setting, the legitimate user is running Linux Fedora with kernel 2 v2.6.5-1.358, whose RTO_{min} is 200 ms. We have conducted experiments under RED. The buffer size is set according to the rule-of-thumb $B = RTT \times R_{bottle}$ [6], and the RED parameters are configured as: $min_{th} = 0.2 * B$, $max_{th} = 0.8 * B$, $w_q = 0.002$, $max_p = 0.1$, and $gentle\_ = true$.

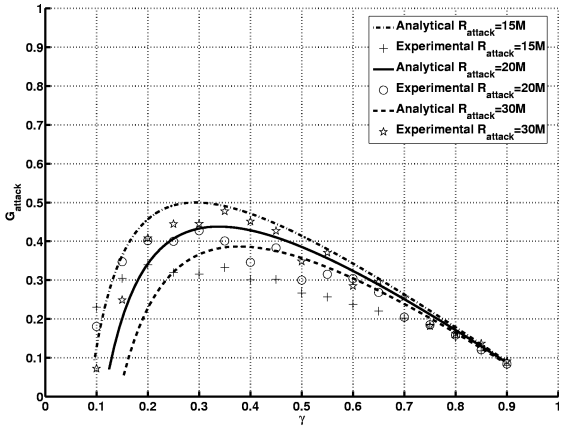


Figure 12. Experiment results obtained in the test-bed.

There are a total of 10 victim TCP flows under 3 kinds of PDoS attacks, which have the same $T_{extent} = 150ms$ but different R_{attack} values. The results are shown in Fig. 12, where all of them match the trends of the analytical results. Moreover, the normal-gain attack is observed when R_{attack} is equal to $20Mbps$. The analytical results usually under-estimate the attack gains when R_{attack} is increased to $30Mbps$. On the other hand, the analytical results usually over-estimate the attack gains when R_{attack} is decreased to $15Mbps$.

5. Conclusions and future work

In this paper, we have investigated how to optimize the PDoS attacks. In particular, we formulate the attack objective based on the maximizing the TCP throughput degradation and minimizing the risk of being detected. As far as we know, this is the first time to study such a tradeoff using an analytical framework. By adjusting the parameters in the at-

tack objective, we can analyze the resulted attacks for different types of attackers who may be risk-averse, risk-loving, or risk-neutral. Moreover, we have obtained the optimized attack parameters for a given attacker's behavior. We have validated the analytical results using both ns-2 simulation and a test-bed.

A limitation of our model is that it does not capture the impact of possible timeouts on the TCP throughput, which would be caused by high-intensity attack pulses. Hence, one of the future works is to extend the analytical models to incorporate the timeout effects. On the other hand, we have discovered that a PDoS attacker can achieve a higher attack gain by attacking a RED router than attacking a drop-tail router. We will report these results in a forthcoming paper and will propose enhancement to the RED algorithms.

Acknowledgment

The work described in this paper was partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 5080/02E) and a grant from the Areas of Excellence Scheme established under the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-01/99). We also thank the anonymous reviewers and Mr. Edmond Chan for their careful reading of the manuscript.

References

[1] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[2] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf 1.7.0. http://dast.nlanr.net/Projects/Iperf/, 2004.

[3] C. Jin, H. Wang, and K. Shin. Hop-count filtering: an effective defense against spoofed DDoS traffic. In *Proc. of ACM Conf. Computer and Communications Security (CCS)*, 2003.

[4] C. Douligeris and A. Mitrokotsa. DDoS attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, 44(5):643–666, Apr. 2004.

[5] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proc. of ACM SIGMOD*, pages 151–162, May 2001.

[6] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *Proc. of ACM SIGCOMM*, pages 281–292, 2004.

[7] G. Yang, M. Gerla, and M. Sanadidi. Defense against low-rate TCP-targeted denial-of-service attacks. In *Proc. of IEEE Symp. on Computers and Communications*, 2004.

[8] H. Sun, J. Lui, and D. Yau. Defending against low-rate TCP attack: Dynamic detection and protection. In *Proc. of IEEE Intl. Network Protocols (ICNP)*, 2004.

[9] H. Wang, D. Zhang, and K. Shin. Detecting SYN flooding attacks. In *Proc. of IEEE INFOCOM*, 2002.

[10] A. Kuzmanovic and E. Knightly. Low-rate TCP-targeted denial of service attacks (the shrew vs. the mice and elephants). In *Proc. of ACM SIGCOMM*, Aug. 2003.

[11] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred. Statistical approaches to DDOS attack detection and response. In *Proc. of DARPA Information Survivability Conf. and Exposition*, 2003.

[12] L. Gordon, M. Loeb, W. Lucyshyn, and R. Richardson. CSI/FBI computer crime and security survey. http://www.gocsi.com, 2004.

[13] X. Luo and R. Chang. On a new class of pulsing denial-of-service attacks and the defense. In *Proc. of Annual Network and Distributed System Security Symposium (NDSS)*, Feb. 2005.

[14] M. Allman, V. Paxson, and W. Stevens. TCP congestion control. RFC 2581, Apr. 1999.

[15] M. Guirguis, A. Bestavros, and I. Matta. Exploiting the transients of adaptation for RoQ attacks on Internet resources. In *Proc. of IEEE Intl. Network Protocols (ICNP)*, 2004.

[16] J. Mirkovic and P. Reiher. A taxonomy of DDoS attacks and defense mechanisms. *ACM SIGCOMM Computer Communications Review*, 34(2):39–54, Apr. 2004.

[17] K. Park and H. Lee. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets. In *Proc. of ACM SIGCOMM*, Aug. 2001.

[18] V. Paxson. An analysis of using reflectors for distributed denial-of-service attacks. *Computer Communication Review*, 31(3), Jul. 2001.

[19] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *Computer Communication Review*, Jul. 2002.

[20] L. Rizzo. Dummynet: a simple approach to the evaluation of network protocols. *ACM Computer Communication Review*, 27(1), Jan. 1997.

[21] S. Floyd, T. Henderson, and A. Gurtov. The NewReno modification to TCP's fast recovery algorithm. RFC 3782, Apr. 2004.

[22] S. Fredj, T. Bonald, A. Proutiere, G. Regnie, and J. Roberts. Statistical bandwidth sharing: a study of congestion at flow level. In *Proc. of ACM SIGCOMM*, Aug. 2001.

[23] Y. Yang and S. Lam. General AIMD congestion control. In *Proc. of IEEE Intl. Network Protocols (ICNP)*, 2000.

[24] L. Zhang and D. Clark. Oscillating behavior of network traffic: a case study simulation. *Internetworking: Research and Experience*, 1(2):101–112, Dec. 1990.