Performance Analysis of TCP/AQM Under Denial-of-Service Attacks

Xiapu Luo, Rocky K. C. Chang, and Edmond W. W. Chan
Department of Computing, The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong, SAR, China
{csxluo|csrchang|c1490305}@comp.polyu.edu.hk

Abstract

The interaction between TCP and various Active Queue Management (AQM) algorithms has been extensively analyzed for the last few years. However, the analysis usually assumed that routers and TCP flows are not under any network attacks. In this paper, we investigate how the performance of TCP flows is affected by denial-of-service (DoS) attacks under the Drop Tail and various AQM schemes. In particular, we consider two types of DoS attacks—the traditional flooding-based DoS (FDDoS) attacks and the recently proposed Pulsing DoS (PDoS) attacks. Both analytical and simulation results support that the PDoS attacks are more effective than the FDDoS attacks under the same average attack rate. Moreover, the Drop Tail surprisingly outperforms the RED-like AQMs when the router is under a PDoS attack, whereas the RED-like AQMs perform better under a severe FDDoS attack. On the other hand, the Adaptive Virtual Queue algorithm can retain a higher TCP throughput during PDoS attacks as compared with the RED-like AQMs.

1 Introduction

The congestion control model used in the Internet today consists of two parts—the source's flow control algorithms that adjust the sending rate in response to congestion signals, and the router's queue management schemes that provide the congestion signals [24]. On the end-to-end congestion control side, TCP, the most widely deployed congestion-reactive protocol, employs additive-increase-multiplicative-decrease (AIMD) algorithms to regulate its sending rate [24]. On the network side, routers use various queue management schemes to provide low latency by dropping packets either when the queue is full, e.g., the Drop Tail scheme, or through an active dropping scheme, e.g., the active queue management (AQM) algorithms [2].

However, the end-to-end congestion control mechanism can be severely disrupted by *misbehaving flows*, which could be congestion-unresponsive flows (e.g., UDP) or denial-of-service (DoS) attack packets. To handle the former, various fairness-oriented AQM schemes have been proposed, such as CHOKe and RED-PD [12], which are based on the fact that congestion-responsive flows can be distinguished from congestion-unresponsive flows. However, the same cannot be said for the latter. That is, DoS attack packets and legitimate packets are generally indistinguishable.

This paper's main objective is to evaluate the impact of DoS attacks on the TCP performance under the Drop Tail scheme and 4 other well-known AQM schemes: RED [19], PI [3], REM [18], and AVQ [21]. We consider 2 types of DoS attacks: the traditional flooding-based DoS attack and the emerging smart DoS attacks, e.g. *Shrew* [8], *RoQ* [11], and *PDoS* [9]. We will mainly consider the PDoS attack in the class of smart DoS attacks, but some of the attack scenarios also correspond to the Shrew attacks.

Most of the previous work related to this paper is on the analysis of the TCP/AQM performance without considering DoS attacks. For instance, many new active queue management schemes have been proposed to improve the performance of TCP flows [3, 18, 21]. Recent surveys and further references can be found in [22, 23, 24]. On the other hand, various techniques have been proposed to defend against the DoS attacks. For example, QoS regulation techniques were employed in [1] to mitigate the effect of DoS attacks on server and network. More information on traditional DoS attack can be found [5, 14]. For the new kind of low-rate attacks, a two-stage algorithm was designed to detect PDoS attacks [9] and a DTW-based approach was proposed to detect Shrew attacks [6]. However, the analysis of the impact of DoS attacks on the TCP performance is largely absent. The one closest to this paper is given in [11] which discusses the impact of RoQ attacks on TCP/RED.

The main contribution of this paper is a thorough evaluation of the DoS attacks' impacts on the TCP performance under different queue management schemes. We have employed both analytical modelling and simulation to achieve our goal. The results obtained from the analysis are very revealing. For example, we have found that the RED-like AQMs, which perform very well when there is no attack, in fact suffer from more serious throughput degradation during PDoS attacks than the Drop Tail and AVQ schemes. Another is that we have proposed 2 new metrics—attack power and attack cost—for a quantitative comparison of the 2 types of attacks. Based on these 2 metrics, we have confirmed that the PDoS attack is indeed much more effective than the flooding-based DoS attack.

The rest of this paper is organized as follows. In section 2, we first review the flooding-based DoS and PDoS attacks, and the models for them. In section 3, we define the power and cost of a DoS attack, and present analytical models for the TCP throughput under the attack-free and attack scenarios. In section 4, we present the NS-2 simulation results to validate the analytical models derived in section 3 and to evaluate the attack's impact on the Drop Tail scheme and the 4 AQM schemes. We finally conclude the paper with future work in section 5.

2 Modelling the DoS Attacks

The DoS attack aims at exhausting a victim's system resources, such as memory and CPU, or network bandwidth. In this paper we consider the latter case for which an attacker can employ different ways to flood a victim with spoofed packets [4]. Therefore, the victim sees a sudden surge in the traffic rate coming into its links, causing a high packet dropping rate. We refer this class of attacks to as the flooding-based DoS (FDDoS) attack, and model it as a traffic source with a constant bit rate R_{FDDoS}.

Recently, a new breed of low-rate DoS attacks has been proposed, e.g. [8, 9], which targets at congestion-reactive protocols, such as TCP. In this paper, we consider the PDoS attack [9] which generalizes the Shrew attack proposed in [8]. Unlike the FDDoS attacks, a PDoS attacker sends a train of attack pulses to induce a sequence of *false* congestion signals—TCP timeouts and duplicate TCP acknowledgments (ACKs)—to victim TCP senders. If the attack pulses are spaced sufficiently small, the TCP senders' congestion window (`cwnd`) will be persistently constrained to a low value.

In the following, we briefly recap the model and some of the results obtained for the PDoS attack from [9]. We model the sequence of attack pulses as $\mathbb{A}(T_{extent}(n), R_{attack}(n), T_{space}(n), N)$, where

- N is the total number of attack pulses sent during an attack.
- $T_{extent}(n), n = 1, 2, \ldots, N$, 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).

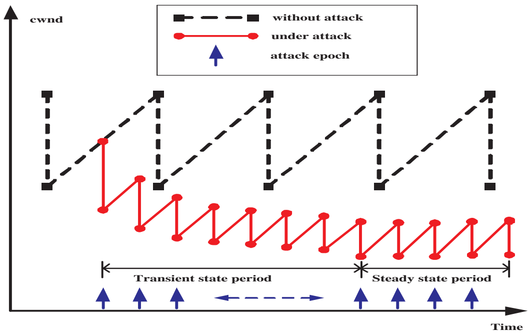


Figure 1. An AIMD-based attack with a train of periodic, fixed attack pulses.

- $T_{space}(n), n = 1, 2, \ldots, N - 1$, is the time between the end of the nth attack pulse and the beginning of the $(n + 1)$th attack pulse.

Note that if $T_{space}(n) = 0$, $\forall n$, the corresponding PDoS attack becomes a FDDoS attack. To simplify the following analysis, we assume that $T_{extent}(n) = T_{extent}$, $T_{space}(n) = T_{space}$, and $R_{attack}(n) = R_{attack}$, $\forall n$, i.e., a train of periodic, identical pulses.

A PDoS attack forces a victim TCP flow to frequently enter either the timeout (`TO`) state (timeout-based attacks) or the fast recovery (`FR`) state (AIMD-based attacks). In the latter, the attack exploits the AIMD algorithm employed by TCP flows to fast recover from network congestions. In general, an AIMD algorithm can be specified by $AIMD(a, b)$, $a > 0$, $1 > b > 0$, in which a sender decreases its `cwnd` from W to bW whenever it enters the `FR` state, and increases its `cwnd` from W to $W + a$ per round-trip time (RTT) until receiving another congestion signal. Many TCP variants, such as Tahoe, Reno, and New Reno, employ $AIMD(1, 0.5)$.

Moreover, many TCP implementations do not send an ACK for every received packet. Instead, they send a delayed ACK after receiving d consecutive full-sized packets, where d is typically equal to 2 [10]. In this case, the sender's `cwnd` is only increased by $\frac{a}{d}$ per RTT. Since it will take at least $\frac{(1-b)d}{a}W$ number of RTTs to restore the `cwnd` back to W after a decline from W to bW, the `cwnd` will be reduced to a low value after periodic packet losses caused by the attack pulses, which is depicted in Fig. 1. 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 may also cause frequent timeouts.

3 Performance Modelling

In this section, we present analytical results on the TCP throughput degradation brought by the DoS attacks. Consider that there are N_f legitimate TCP flows traversing through a router, and a DoS attack causes the router to drop packets. Assume that the bandwidth of the router's outgoing link is given by R_{bottle}. We define in Defs. 1-2 *attack power* and *attack cost* for a DoS attack, respectively. While the attack power measures the impact of the attack on the legitimate TCP flows, the attack cost measures the intensity of the attack, in terms of the attack rate normalized by the bottleneck bandwidth. In the rest of this paper, whenever we compare the PDoS attack and the FDDoS attack based on their power, we assume that they have the same attack cost.

Definition 1. *The power of a DoS attack, denoted by Γ, is defined as*

$$\Gamma = 1 - \frac{\sum_{i=1}^{N_f} \Psi^i_{attack}}{\sum_{i=1}^{N_f} \Psi^i_{normal}}, \tag{1}$$

where Ψ^i_{attack} and Ψ^i_{normal} denote the amount of data (bytes) sent by the ith TCP flow in the presence of and in the absence of a DoS attack within the same period, respectively.

Definition 2. *The cost of a DoS attack, denoted by γ, is defined as*

$$\gamma = \frac{R_{DoS}}{R_{bottle}},$$

where R_{DoS} is the average attack rate of the DoS attack. For a FDDoS attack, we have $\gamma = \frac{R_{FDDoS}}{R_{bottle}}$, while the cost of a PDoS attack is given by

$$\gamma = \frac{R_{attack} T_{extent}}{R_{bottle} T_{attack}}, \tag{2}$$

where $T_{attack} = T_{extent} + T_{space}$ is the period of the PDoS attack.

In the following we will derive $\sum_{i=1}^{N_f} \Psi^i_{normal}$ (Prop. 1) and $\sum_{i=1}^{N_f} \Psi^i_{attack}$. In the latter, we first present a simple model for the FDDoS attacks (Prop. 2) which has been validated by simulation results, and then the results for the PDoS attacks (Prop. 3, Lemma 1, and Prop. 4).

Proposition 1. *Since TCP flows will make a full use of the bottleneck bandwidth in the absence of attacks [20], we have*

$$\sum_{i}^{N_f} \Psi^i_{normal} = R_{bottle} T_{total}/8, \tag{3}$$

where T_{total} denotes the period that the TCP flows are under a DoS attack. For the case of PDoS attacks with N pulses, $T_{total} = (N-1)T_{attack}$.

Proposition 2. *In the presence of a FDDoS attack with $R_{attack} = \beta R_{bottle}$, $0 < \beta \leq 1$, $\sum_{i=1}^{N_f} \Psi^i_{attack} = (1 - \beta) \sum_{i=1}^{N_f} \Psi^i_{normal}$.*

Proof. Since we model a FDDoS attack as a traffic source with constant bit rate, its impact on the normal traffic is approximately the same as reducing the available bandwidth by the attack rate. According to the TCP congestion control mechanism, the TCP throughput will increase when there is additional bandwidth to transfer packets. Thus, the TCP flows will make a full use of the remaining bandwidth. □

As for the PDoS attacks, the analysis is not straightforward, since the effect of the attack depends on the attack power. Thus, we consider 2 specific scenarios, which are referred to as the *perfect timeout* PDoS (PT-PDoS) attack and the *perfect AIMD* PDoS (PA-PDoS) attack. In a PT-PDoS attack, each legitimate TCP flow is forced to enter the TO state by a PDoS attack pulse. Therefore, this scenario corresponds to the most severe impact inflicted by the attack. In a PA-PDoS attack, each legitimate TCP flow is forced to enter the FR state by an attack pulse. Obviously, there are many other possibilities, depending on the PDoS attack power.

In Prop. 3 we first recall the result for the PA-PDoS attack from [25], and then obtain the result for the PT-PDoS attacks in Prop. 4 which can be proved with the aid of another result in Lemma 1.

Proposition 3. *The amount of data sent by the ith legitimate TCP flow under a PA-PDoS attack can be approximated by*

$$\Psi^i_{attack} = \frac{a(1+b)T^2_{attack} S_p}{2d(1-b)RTT_i^2}(N-1), \tag{4}$$

where RTT_i denotes the RTT of the ith legitimate TCP flow, and S_p is the data packet's size which is assumed to be the same for all legitimate flows.

Lemma 1. *The maximal congestion window of the ith legitimate TCP flow in the steady state, denoted by W_i^U, can be estimated by*

$$W_i^U = \frac{2R_{bottle}}{(1+b)S_p}\left(\sum_{j=1}^{N_f} \frac{1}{RTT_j}\right)^{-1}. \tag{5}$$

Proof. According to [20], the N_f TCP flows share the bandwidth R_{bottle} in an inverse proportion to their RTTs. That is, $\sum_{i=1}^{N_f} BW_i = R_{bottle}$ *and* $\frac{BW_i}{BW_j} = \frac{RTT_j}{RTT_i}$. Therefore,

$$BW_i = \frac{R_{bottle}}{RTT_i}\left(\sum_{j=1}^{N_f} \frac{1}{RTT_j}\right)^{-1}, \tag{6}$$

where BW_i is the bandwidth obtained by the ith flow in a no-attack scenario. By assuming that all TCP flows stay in the congestion avoidance state, we have $\frac{(W_i^U + W_i^L)}{2} \frac{T}{RTT_i} S_p = BW_i T$ and $W_i^L = bW_i^U$ for a period T. By solving the equation for W_i^U, we can obtain Eq. (5). □

Proposition 4. *The amount of data sent by the ith legitimate TCP flow under a PT-PDoS attack, denoted by $\Psi^{worst}_{attack,i}$, is given by*

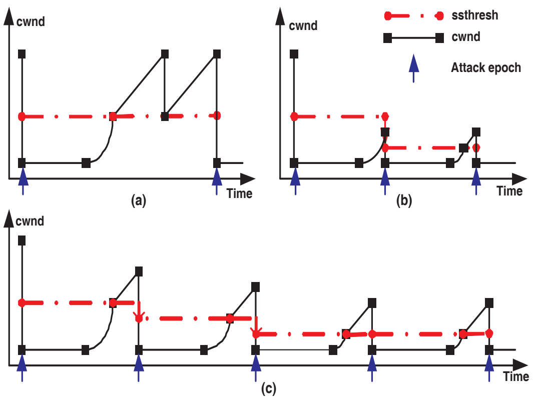


Figure 2. The trajectory of `cwnd` and `ssthresh` under PDoS attacks.

1. *If* $T_{maxcvg,i} < T_{period,i}$,

$$\Psi^{worst}_{attack,i} = (T_{period,i} - RTO_i)BW_i \left\lfloor \frac{(N-1)T_{attack}}{T_{period,i}} \right\rfloor.$$

2. *If* $T_{mincvg,i} \le T_{period,i} \le T_{maxcvg,i}$,

$$\Psi^{worst}_{attack,i} = (\frac{3d}{8a}W^2_{c,i} + \frac{\frac{1}{2}\tau_d W_{c,i} - 1}{\tau_d - 1})S_p \left\lfloor \frac{(N-1)T_{attack}}{T_{period,i}} \right\rfloor.$$

3. *If* $RTO_i < T_{period,i} < T_{mincvg,i}$,

$$\Psi^{worst}_{attack,i} = [1 + \frac{a}{2d}(\frac{T_{period,i} - RTO_i}{RTT_i} - 1)^2 + 2(\frac{T_{period,i} - RTO_i}{RTT_i} - 1)]S_p \left\lfloor \frac{(N-1)T_{attack}}{T_{period,i}} \right\rfloor.$$

4. *If* $T_{period,i} = RTO_i$,

$$\Psi^{worst}_{attack,i} = 0.$$

where,

$$\begin{aligned} T_{period,i} &= \lceil \frac{RTO_i}{T_{attack}} \rceil T_{attack}, \\ T_{maxcvg,i} &= [\frac{\ln(W^U_i/2)}{\ln(\tau_d)} + \frac{d}{2a}W^U_i]RTT_i + RTO_i, \\ T_{mincvg,i} &= (1 + \frac{2d}{a})RTT_i + RTO_i, \\ W_{c,i} &= 2e^{\frac{-LambertW(LamC)RTT_i + \ln(\tau_d)(Tperiod,i - RTO_i)}{RTT_i}}, \\ LamC &= \frac{a}{d}\ln(\tau_d)\, e^{\frac{\ln(\tau_d)\left(T_{period,i} - RTO_i\right)}{RTT_i}}. \end{aligned}$$

And RTO_i *is the retransmission timeout value of the ith flow and LambertW denotes the Lambert's* $\mathbb{W}$ *function [17].*

Proof. In the worst case, each attack pulse forces all TCP flows to enter the `TO` state. Therefore, Ψ_{attack} is equal to the amount of data sent during the period starting from the end of a timeout to the beginning of the next attack pulse, i.e. $T_{period,i} - RTO_i$, which we call a *run*. Consequently, the amount of data sent by the legitimate TCP flows under a PDoS attack with N pulses is equal to that sent during $\left\lfloor \frac{(N-1)T_{attack,i}}{T_{period,i}} \right\rfloor$ runs. Similar to the previous analysis of TCP [13, 7], we assume that each TCP flow's RTT is a constant value. Moreover, we assume that the RTO is a constant value, because the TCP sender recomputes the RTO value only after retransmitting the lost packets.

The first scenario is depicted in Fig. 2(a). As shown, this scenario corresponds to the case where $T_{period,i}$ is long enough, so that the TCP sender recovers the lost packets as well as increases its `cwnd` to the maximal value of W^U_i. As a result, the `ssthresh` maintains its maximal value of $\frac{W^U_i}{2}$. Therefore, we may use the ratio $\frac{T_{period,i} - RTO_i}{T_{period,i}}$ to estimate the throughput degradation. A similar method has been applied to analyze the Shrew attack in [8].

The second scenario is depicted in Fig. 2(b). In this scenario, $T_{period,i}$ is short enough that its `cwnd` cannot reach 4. Thus, the `ssthresh` will be constrained to the minimal value of 2 [10]. Accordingly, during the attack-free period, the TCP sender enters the congestion avoidance phase after sending a packet, because the `cwnd` will reach the `ssthresh` after receiving an ACK. As a result, the amount of data sent in a *run* is the summation of data segments sent in the slow start phase, and those sent in the congestion avoidance phase, i.e. $\frac{1}{2}[2 + 2 + \frac{a}{d}(\frac{T_{period,i} - RTO_i}{RTT_i} - 1)](\frac{T_{period,i} - RTO_i}{RTT_i} - 1)S_p = [\frac{a}{2d}(\frac{T_{period,i} - RTO_i}{RTT_i} - 1)^2 + 2(\frac{T_{period,i} - RTO_i}{RTT_i} - 1)]S_p$.

The third scenario is depicted in Fig. 2(c) in which the PDoS attack will drive the `ssthresh` to a constant value. Consequently, the `cwnd` reaches $W_{c,i}$ before the next attack pulse' arrival, and the `ssthresh` converges to $\frac{1}{2}W_{c,i}$. In order to estimate the amount of data sent in the slow start phase, we consider 2 common cases: $d = 1$ and $d = 2$. When $d = 1$, the number of packets sent in the nth RTT is 2^n during the slow start phase [10]. When $d = 2$, we employ a simple model proposed in [15] to approximate the number of packets sent in the nth RTT to 1.5^n. In order to give a unified presentation, we let

$$\tau_d = \begin{cases} 2 & \text{if} \quad d = 1, \\ 1.5 & \text{if} \quad d = 2. \end{cases} \qquad (7)$$

As a result, we obtain the following equations:

$$\begin{gathered} (\tau_d)^x = \tfrac{1}{2}W_{c,i}, \\ \tfrac{a}{d}y = \tfrac{1}{2}W_{c,i}, \\ (x + y)RTT + RTO = T_{period,i}, \\ x > 0,\ y > 0. \end{gathered}$$

By solving these equations, we obtain the value of $W_{c,i}$.

Therefore, the TCP data sent in each run consists of those sent during the slow start phase, i.e. $S_p \sum_{i=0}^{x}(\tau_d)^i = S_p \frac{\tau_d^{x+1} - 1}{\tau_d - 1}$, and those sent during the congestion avoidance phase, i.e. $S_p(\frac{W_{c,i}}{2} + W_{c,i})\frac{y}{2} = S_p \frac{3d}{8a} W^2_{c,i}$. Moreover, as

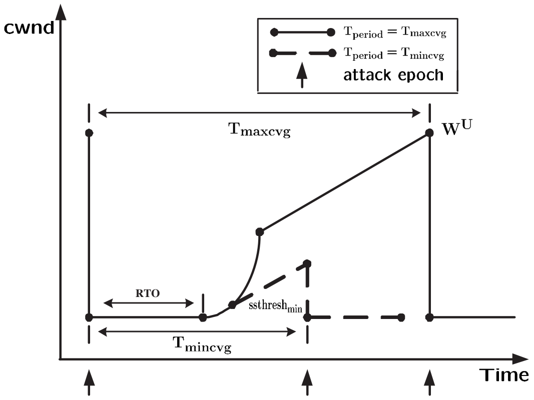


Figure 3. The relationship between W^U, T_{maxcvg}, $ssthresh_{min}$, and T_{mincvg}.

shown in Fig. 3, we can compute $T_{maxcvg,i}$ and $T_{mincvg,i}$ when $W_{c,i} = W_i^U$ and $W_{c,i} = 2ssthresh_{min} = 4$ [10], respectively.

The last scenario ($T_{period,i} = RTO_i$) is in fact a Shrew attack [8], and the legitimate TCP throughput is degraded to zero. □

4 Simulation Experimentation

We have conducted extensive NS-2 simulation experiments to validate our analytical results and to evaluate the impact of DoS attacks on different AQMs. The network topology used in the simulations is depicted in Fig. 4. The network consists of M pairs of TCP senders and receivers. All the links, except for the bottleneck between routers S and R, are $50Mbps$. The two routers are connected through a link of $10Mbps$. There are 10 legitimate TCP flows traversing through the bottleneck link, all of which are based on TCP New Reno, and their RTTs range from 20ms to 460ms as suggested in [8]. The $minRTO$ of each flow is equal to 1s according to the recommendation in [16]. Based on the scripts provided by [8], all the simulation experiments were performed in the NS-2 2.28 environment. The queue size (QS) is 100 packets and the AQMs' parameters are listed in Table 1.

Table 1. Parameters for the 4 AQMs.

AQMs	Customized Parameters
RED	$max_{th} = 0.8QS$, $min_{th} = 0.2QS$, $max_p = 0.1$, $w_q = 0.002$, gentle=ture
REM	$b^* = 0.6QS$, $\gamma = 0.001$, $\phi = 1.001$
PI	$q_{ref} = 0.6QS$, $a = 0.00001822$, $b = 0.00001816$
AVQ	$\alpha = 0.15$, $\gamma = 0.98$

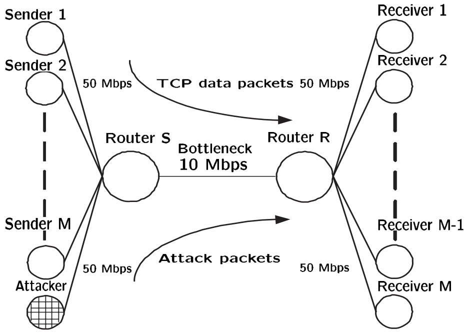


Figure 4. The network topology for the simulation studies.

4.1 The experiments

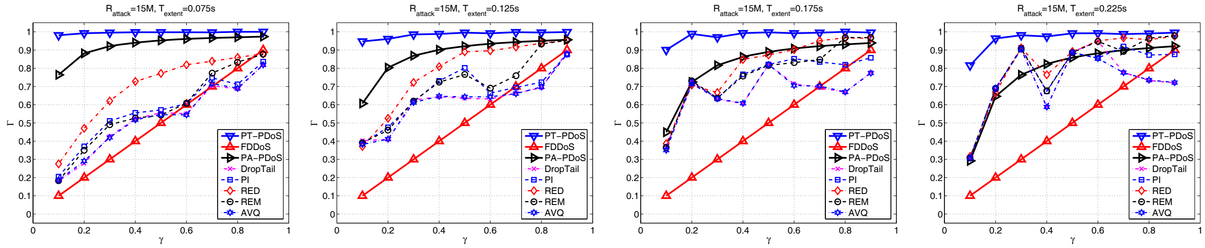
Figs. 5-6 plot the attack power Γ verses the attack cost γ for the FDDoS and PDoS attacks for 2 different values of R_{attack}. Each figure has 4 sub-figures showing different values of T_{extent} for the PDoS attack scenarios, which obviously do not affect the FDDoS attack results. For the FDDoS attack, we only present the analytical results (the solid straight lines), because they match very well with the simulation results. As for the PDoS attacks, the 2 solid lines are obtained from the analytical results for the PT-PDoS and PA-PDoS attacks which are derived without considering specific queue management schemes. On the other hand, the 5 dashed lines are obtained from the simulation results for the 5 queue management schemes under the PDoS attack.

Figs. 7-8 present the simulation results for the packet dropping rates, denoted as ζ, for the PDoS and FDDoS attacks, respectively. To clearly explain the results, we have also included the corresponding graphs for the attack power. In Fig. 7, the PDoS attacks were launched with $T_{extent} = 125ms$ and $R_{attack} = \{20, 30\}Mbps$. We have computed ζ separately for the legitimate TCP packets (denoted by TP) and for the attack packets (denoted by AP). For example, *RED-TP* refers to the ζ for the legitimate TCP packets and RED is in use. This is similarly done for the FDDoS attacks in Fig. 8.

Fig. 9 gives the packet dropping probabilities used in the 3 RED-like AQM algorithms measured during the PDoS attacks with $T_{extent} = 125ms$, $R_{attack} = \{10, 20, 30\}Mbps$, and $\gamma = 0.3$. As we shall see, this set of results is useful in explaining why RED drops more legitimate TCP packets than REM and PI do.

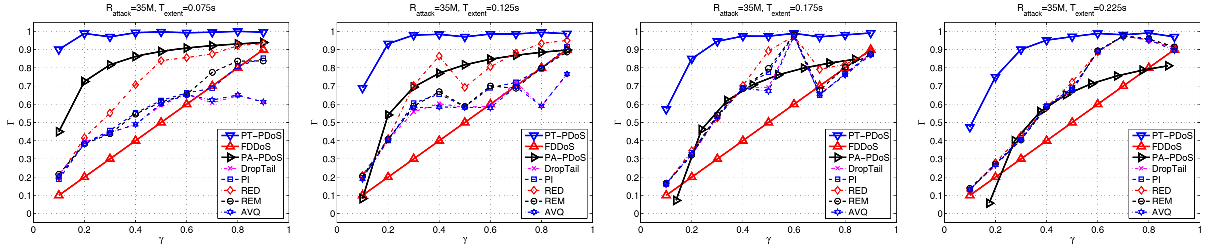
4.2 The PDoS attack power

According to Figs. 5-6, the results for the PT-PDoS attack can be regarded as the upper bound for the PDoS at-



(a) $T_{extent} = 75ms$ (b) $T_{extent} = 125ms$ (c) $T_{extent} = 175ms$ (d) $T_{extent} = 225ms$

Figure 5. The DoS attack power with $R_{attack} = 15Mbps$.



(a) $T_{extent} = 75ms$ (b) $T_{extent} = 125ms$ (c) $T_{extent} = 175ms$ (d) $T_{extent} = 225ms$

Figure 6. The DoS attack power with $R_{attack} = 35Mbps$.

tack power. Moreover, the figures show abrupt changes in the attack power for some parameter settings, e.g., $\gamma = 0.3$ in Fig. 5(d) and $\gamma = 0.6$ in Fig. 6(c). In these cases, the attack periods ($T_{attack} = 1125ms, 1021ms$) are very close to that of the Shrew attack [8]. Therefore, the PDoS attacks would drive the TCP flows into the `TO` state as soon as the TCP senders' retransmission timers expire, thus causing a very severe throughput degradation. These special attack parameters are referred to as *Shrew points* in [25].

For a given R_{attack}, the simulation results approach to those given by the PA-PDoS and PT-PDoS attacks as T_{extent} increases. That is, the PDoS attack power increases with T_{extent}, because more attack packets are sent in each attack pulse, which would quickly ramp up the packet dropping probability for the queue management schemes. As a result, more legitimate TCP packets will be dropped.

Another interesting result is that the trend of the simulation results obtained for RED coincides very well with that of PA-PDoS attack in some cases, such as Fig. 5(c) and Fig. 6(b). Recall that a PA-PDoS attack forces each TCP flow to enter the `FR` state. On the other hand, RED uses an uniform dropping mechanism to avoid consecutive packet dropping [19], which therefore affects more TCP flows during a PDoS attack. Hence, the simulation results for RED are in good match with the analytical results obtained for the PA-PDoS attacks.

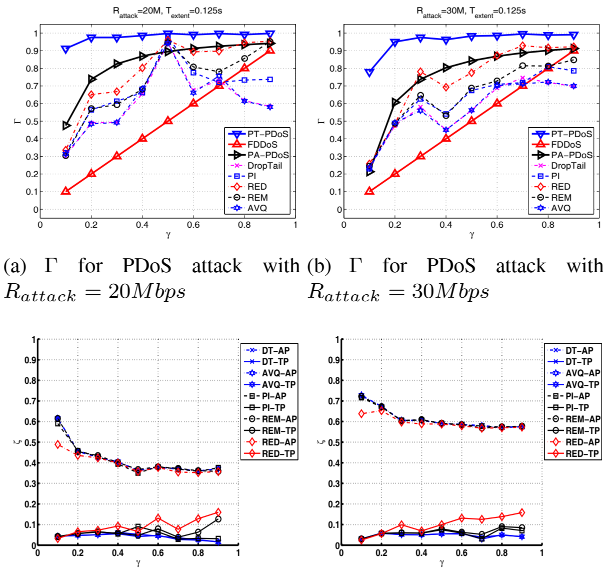
4.3 The resilience level of DropTail and AQMs under PDoS attacks

Based on the throughput degradation results in Figs. 5-6, we can compare the resilience levels of the queue management schemes to the PDoS attacks. The figures have concluded the following order of resilience level for the 5 schemes: $\{AVQ, DropTail\} \geq \{PI, REM\} \geq RED$. The ones within $\{\}$ are considered to have a very similar resilience level.

In Fig. 7(c-d), the curves for the attack packets (AP) are all clustered together in the range of $\zeta = 0.35 - 0.6$. The curves for the legitimate packets (TP), on the other hand, lie below the curves for the attack packets. That is, the packet dropping rates for the attack packets are always higher than that for the legitimate packets. Besides, DropTail and AVQ drop relatively more attack packets but less TCP packets, while the RED-like AQMs drop relatively less attack packets but more TCP packets. In particular, RED drops the least number of attack packets but the largest number of TCP packets on average. This result is due to its random drop mechanism which would let the attack packets pass through the router even when the queue is full. These attack packets also push up the packet dropping probability for the legitimate TCP packets. On the contrary, DropTail and AVQ would drop all the subsequent attack packets whenever the queue is full, thus effectively dampening the power of the attack pulse.

Fig. 9 reveals 2 factors responsible for the inferior performance of RED as compared with REM and PI. First, the abrupt arrivals of the attack packets increase RED's average queue length drastically, thus resulting in a very high packet dropping probability for both attack packets and legitimate TCP packets. However, RED's uniform dropping cannot drop the attack packets quickly enough, which instead increases the dropping of legitimate TCP packets. Second, RED's packet dropping probability decreases more slowly than REM and PI, because both REM and PI use the instantaneous queue length to compute the packet dropping probability.

Furthermore, as R_{attack} or T_{extent} increases, the results for AVQ and DropTail are almost the same, because they essentially have the same packet dropping strategy, except for the use of a virtual queue in AVQ. Similarly, REM and PI have very similar results, because both are designed based on the idea of proportional-integral controller.



(a) Γ for PDoS attack with $R_{attack} = 20Mbps$

(b) Γ for PDoS attack with $R_{attack} = 30Mbps$

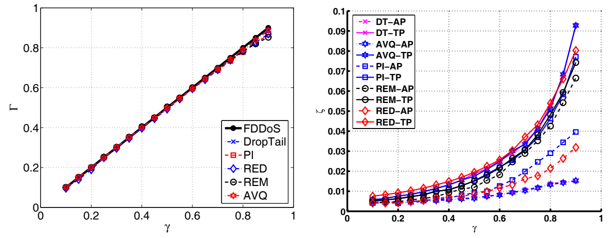
(c) ζ for PDoS attack with $R_{attack} = 20Mbps$

(d) ζ for PDoS attack with $R_{attack} = 30Mbps$

Figure 7. The attack power and packet dropping rates under PDoS attacks with $T_{extent} = 125ms$.

4.4 The resilience level of DropTail and AQMs under FDDoS attacks

Fig. 8(a) shows that the simulation results for the FDDoS attack are very close to the analytical results. Fig. 8(b) shows that the packet dropping rates for the attack packets and the TCP packets under DropTail and the 4 AQMs are very similar when γ is small, but they diverge as γ increases. Moreover, the difference is smaller for the RED-like AQMs when compared with DropTail and AVQ. This shows that the RED-like AQMs can achieve a higher resilience level than DropTail and AVQ, which is opposite to the results obtained under the PDoS attacks. This can be explained by the fact that TCP flows always try to make a full use of the available bandwidth. Therefore, the random drop mechanism employed by the RED-like AQMs would offer a better chance for the TCP flows to use the extra bandwidth by dropping the attack packets, while the DropTail and AVQ do not have such mechanism.



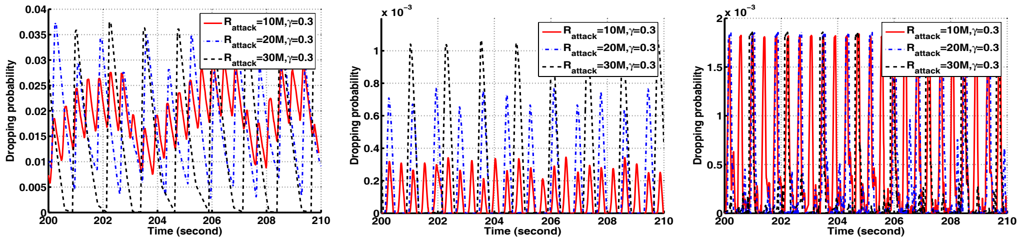
(a) Γ under FDDoS

(b) ζ under FDDoS

Figure 8. The attack power and packet dropping rates under FDDoS attacks.

5 Conclusions and Future Work

In this paper, we have modelled the impact of the FDDoS and PDoS attacks on the TCP throughput under different queue management schemes, including DropTail and 4 AQM schemes. There are several important results obtained from the analytical and simulation results. First, under a PDoS attack, the RED-like AQMs suffer from a higher throughput degradation than the DropTail and AVQ do, because the latter discards the incoming packets only when the (virtual) queue is full. Second, the packet dropping rates under the queue management schemes behave quite differently for the FDDoS and PDoS attacks. During a PDoS attack, the packet dropping rates for the attack packets are almost the same, while they are different for the legitimate TCP packets. In particular, both DropTail and AVQ tend to drop fewer legitimate TCP packets but more attack packets as compared with the RED-like AQMs. However, the results are opposite for a FDDoS attack. Third, the PDoS attack is indeed more effective than the traditional FDDoS attack, because the former has a much higher attack power and a smaller attack cost. In the future work, we intend to improve the existing AQMs to mitigate the impact of PDoS attacks based on the analysis in this paper.



(a) Packet dropping probability for RED.
(b) Packet dropping probability for REM.
(c) Packet dropping probability for PI.

Figure 9. Packet dropping probabilities for RED, REM, and PI under PDoS attacks.

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), 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), and a grant from the Cisco University Research Program Fund at Community Foundation Silicon Valley. We also thank the anonymous reviewers for their comments.

References

[1] A. Garg and A. Reddy. Mitigation of DoS attacks through QoS regulation. In *Proc. IWQoS*, 2002.

[2] B. Braden et al. Recommendations on queue management and congestion avoidance in the Internet. RFC 2309, Apr. 1998.

[3] C. Hollot, V. Misra, D. Towsley, and W. Gong. On designing improved controllers for AQM routers supporting TCP flows. In *Proc. IEEE INFOCOM*, 2001.

[4] R. Chang. Defending against flooding-based, distributed denial-of-service attacks: A tutorial. *IEEE Communications Magazine*, 40(10), 2002.

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

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

[7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proc. ACM SIGCOMM*, Sep. 1998.

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

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

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

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

[12] M. Huggard, M. Robin, A. Bitorika, and C. McGoldrick. Performance evaluation of fairness-oriented active queue management schemes. In *Proc. IEEE MASCOTS*, 2004.

[13] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3), Jul. 1997.

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

[15] N. Cardwell, S. Savage, and T. Anderson. Modeling TCP latency. In *Proc. of IEEE INFOCOM*, 2000.

[16] V. Paxson and M. Allman. Computing TCP's retransmission timer. RFC 2988, Nov. 2000.

[17] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth. On The Lambert W Function. *Advances in Computational Mathematics*, 5:329–359, 1996.

[18] S. Athuraliya, V. Li, S. Low, and Q. Yin. REM: Active queue management. *IEEE Network*, May 2001.

[19] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Networking*, 1(4), 1993.

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

[21] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue algorithm for active queue management. *IEEE/ACM Trans. Networking*, April 2004.

[22] S. Liu, T. Basar and R. Srikant. Controlling the Internet: A survey and some new results. In *Proc. of IEEE Conference on Decision and Control*, 2003.

[23] S. Ryu, C. Rump, and C. Qiao. Advances in Internet congestion control. *IEEE Communications Surveys and Tutorials*, 5(1), 2003.

[24] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[25] X. Luo and R. Chang. Optimizing the pulsing denial-of-service attacks. In *Proc. IEEE DSN*, 2005.