I.J. Wireless and Microwave Technologies, **2011, 6, 59-64**
Published Online December 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2011.06.09

Available online at http://www.mecs-press.net/ijwmt

Defending Against LDoS Attacks Using Fair AQM

Bianqin Wang[a], Shunzheng Yu[b]

[a,b] *Education & Experiment Center, Sun Yat-sen University, Guangzhou Higher Education Mega Center, China*
[a] *School of Information Science and Technology, Sun Yat-sen University, Guangzhou Higher Education Mega Center, China*

Abstract

According to the instant high rate and high intensity of LDoS attacks, this paper explores using fair queue management mechanism to mitigate their effect. We perform simulation experiments to evaluate the performance of fair AQM FRED and CHOKe under LDoS attacks. The simulation results show that they are able to reduce the impact of the attacks in various degrees. FRED outperforms CHOKe in throttling the attacks, but it is slightly inferior to CHOKe in time performance.

Index Terms: LDoS attacks; FRED; CHOKe; fairness

1. Introduction

Low-rate Denial-of-Service (LDoS) attacks are different from the traditional Flooding-based Denial of Service (FDoS) attacks. They don't have to maintain high pulse intensity of the attack flows to exhaust all available resource of the victim but send a large number of packets periodically in a given short time interval to reduce the performance of the victim by taking advantage of the well-known vulnerability in the adaptive mechanism of some network protocols or application services (for example, the TCP's congestion control mechanism). LDoS attacks only send out attack packets in certain time interval of a period and do nothing in the other time of the same period. This characteristic of intermittent attack keeps the average rate of the attack flows low, just nearly the same as the normal flow's. Due to concealment, LDoS attacks no long have the abnormal statistical features to elude detection.

Research on LDoS attacks has some achievements. Kuzmanovic et al. [1,2] presented the first potential LDoS attack model-Shrew attack which utlized only a small amount of data lead to the victim's denial of service and low quality of service. Then Xiapu et al. [3] did a deep research on Shrew attack, defined Pulsing-based Denial of Service (PDoS) attack and presented a novel two-stage scheme to detect PDoS attacks on a victim network by wavelet analysis. Guirguis et al. [4,5] proposed the attack of Reduction of Quality (RoQ), which combined the

* Corresponding author.
E-mail address: [a]wangbq@mail.sysu.edu.cn; [b]syu@mail.sysu.edu.cn

vulnerability in TCP's congestion control mechanism with router's queue management mechanism to reduce the performance of a specific router. Haibin et al. [6] used a distributed mechanism to detect and identify LDoS attacks. Chen Yu et al. [7] presented a defense method called for collaborative detection and filtering (CDF) of shrew DDoS attacks using spectral analysis.

Owing to deception and variation of parameters of the LDoS-attack modes, it is very difficult to achieve accurate abnormal features in either time domain, frequency domain or wavelet domain. Moreover, how to realize online detection of LDoS attacks and instant reaction, which will mitigate their impact on normal flows, is still a challenge to these methods.

One purpose of fair strategy is to implement fair allocation of network resource among different flows when network congestion takes place. That is, flows with the same condition should share the same network resource. Although the average rate of LDoS attacks is low, the instant rate is still relatively high during attack, which is a kind of short-term abusive of network resource.

Active Queue Management (AQM) [8] can supervise competition for network resource by fair bandwidth allocation among different flows to mitigate network congestion. It can monitor the change of buffer size of routers and audit the fairness of different flows in sharing bandwidth and their influence on congestion, and take a step forward, to decide when to drop, which packet to drop. Thus, choosing the appropriate AQM can control LDoS attacks effectively and lower their impact on normal flows.

Different AQM has different control effects over LDoS attacks. Fair AQM is able to segregate and control the attack flows by identifying them to reduce their influence on normal flows. The paper will focuses on analysis of the performance of both FRED and CHOKe by evaluating packet loss rate, throughput and packet delay under LDoS attacks, which will provide valid evidence in choosing and employing fair AQM. Restraining LDoS attacks by fair AQM only requires extending the function of routers without too much additional hardware and software overhead. Therefore, it is an effective approach to defense against LDoS attacks.

2. Fair AQM

Random Early Detection (RED) [9] is the first AQM which is widely focused and researched. RED can notify connections of congestion by dropping packets at the gateway. Its main goal is to provide congestion avoidance by controlling the average queue size *avg*. The RED calculates the *avg*, using a low-pass filter with an exponential weighted moving average (1). Then the *avg* is compared with two thresholds, a minimum threshold *minth* and a maximum threshold *maxth*. When the *avg* is less than the minth, no packets are dropped; when the *avg* exceeds the *maxth*, every arriving packet is dropped; when the *avg* is between the *minth* and the *maxth*, each arriving packet is dropped with probability P_a (2), where P_a is a function of the *avg*.

- calculating average queue size *avg*

$$avg = (1 - w_q) \times avg + w_q \times q \tag{1}$$

Where *q* is the instant queue length, the weight *wq* determines the time constant of the low-pass filter.

- calculating packet loss probability P_a

$$p_b = \begin{cases} 0 & avq < \min_{th} \\ \max_p (avg - \min_{th}) / (\max_{th} - \min_{th}) & \min_{th} \le avg \le \max_{th} \\ 1 & avg > \max_{th} \end{cases} \tag{2}$$

$$p_a = p_b / (1 - count \times p_b)$$

Where the count is the number of packets buffered into the current queue since the last dropped packet. As the count increases, the possibility of the next packet which is discarded increases slowly.

The calculation of the probability P_a depends on the current queue length and has nothing to do with per-flow state. That is, in a certain time interval, the loss probability P_a of all the incoming packets remains the same, no matter how much bandwidth their flows share. To improve the fairness of different flows sharing bandwidth found in RED, some improved methods are proposed, among which FRED and CHOKe are typical.

2.1. Fred

Flow Random Early Detection (FRED) [10] is improved from RED, which elevates the fairness in bandwidth allocation completely based on flow state information. Taking FRED as queue mechanism, different flows have different packer loss probability.

A flow which has packets in the current queue is called an active flow. And some new variants are introduced such as *minq*(minimum number of packets each flow should be allowed to buffer), *maxq*(maximum number of packets each flow should be allowed to buffer) and *avgcq*(average number of packets in each flow in the current queue). FRED maintains state information for each active flow, including a count of buffered packets $qlen_i$ and variable $strike_i$ which counts the number of times the flow has failed to respond to congestion notification. FRED penalizes flows with high strike values.

FRED takes ($qlen_i \geq maxq$) ‖ ($avg \geq maxth$ && $qlen_i > 2*avgcq$) ‖ ($qlen_i \geq avgcq$ && $strike_i > 1$) as the condition to identify and punish ill-behaved flows. When the condition is satisfied, the $strike_i$ plus 1, and drops the current packet at the same time. FRED never lets a flow buffer more than *maxq* packets, and counts the number of times each flow tries to exceed *maxq* in the $strike_i$ variable. Flows with high strike values are not allowed to queue more than *avgcq* packets. It allows adaptive flows to send bursts of packets, but prevents ill-behaved flows from consistently monopolizing the buffer space.

FRED needs routers to maintain state information for active flows. When flows increase, the burden on routers would become heavier. Consequently, FRED has poor scalability.

2.2. CHOKe

CHOKe(CHOose and Keep for responsive flows, CHOose and Keep for unresponsive flows) [11] is an nearly fair AQM which needs no flow state information while employs flow identification mechanism to determine whether to drop a packet or not.

When a packet comes into a congested router, randomly retrieve one packet from the current queue and compare it with the incoming packet. If they belong to the same flow, they are dropped together; otherwise, the retrieved packet is put back into the current queue without any change, and the incoming packet is dropped with a certain probability which is calculated in the same way as in RED.

Packets from a flow using more bandwidth will be dropped with a greater probability. On one hand, flows using more bandwidth have more packets in the current queue, so when retrieving packets randomly from the current queue, the packets from these flows are more likely to be chosen; on the other hand, in a time unit, more packets from the flows are coming into routers.

CHOKe controls unresponsive or ill-behaved flows with a minimum overhead since it is stateless and easy to implement. Unfortunately, it utilizes one time comparison to identify flows sharing more bandwidth. With the increase of flows, the number of packets of each flow in the current queue would be reduced, and CHOKe's efficiency would be decreased too.

3. Similation Results

We evaluate the performance of FRED and CHOKe under LDoS attacks using NS2 (Network Simulator version 2)[12]. The performance metrics include throughput, packet loss rate, and packet delay of normal flows

as well as packet loss rate of attack flows in a bottleneck link. Our goal is to keep the throughput high, packet loss rate and packet delay low for normal flows, while the packet loss rate for attack flows as high as possible.

The network topology is shown in Fig. 1. There are four normal TCP nodes (TCP1~TCP4), one attack node (Attack) and two routers (Router1, Router2). The link between Router1 and Router1 is bottleneck link.

According to the statistics, 95% bytes and 90% packets in the Internet are transmitted in TCP protocol [13]. Hereby, in the simulation, all the normal flows are TCP flows. And the congestion control mechanism is TCP Reno. The other parameters are as follows: (i) the bandwidth and delay of links of the normal flows which are generated by FTP with 1kB size of per packet are 15Mbps and10ms respectively. All user nodes initiate connections at zero second and stop until the simulation ends; (ii) the bandwidth and delay of the link between two routers are 3Mbps and 10ms respectively; (iii) the attack flow is UDP flow which is generated by CBR flow generator. The size of each packet is 1kB. The attack period is 0.7seconds and the attack sustains 0.2 seconds one time. Five attacks are launched in one period; (iv) simulation time is 5 seconds. Except for the bottleneck link, the other links use DropTail as queue management mechanism.

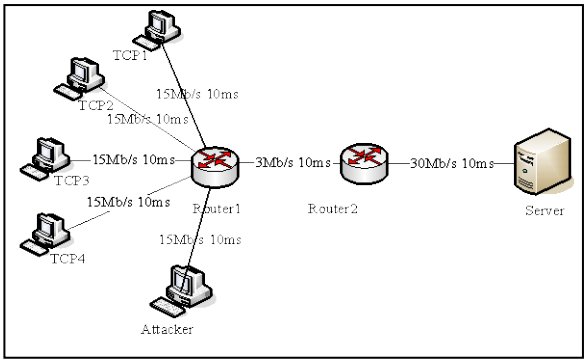

Fig 1. Network Configuration.

The bottleneck link takes FRED, CHOKe and RED (represents poor fair AQM) respectively as queue mechanism. To compare them effectively, the bottleneck queue size is set to 15 packets, whose maximum is 15 packets and minimum is 5 packets. The simulation results are shown in Fig. 2, Fig. 3 and Fig. 4.

Packet loss rate: As Fig. 2 shows, among three queue mechanisms, the packet loss rate of the normal flows is the lowest while the attack flow's is the highest using FRED. However, under RED, the packet loss rate of the normal flows is the highest while the attack flow's is the lowest. In case of CHOKe, the packet loss rate of the normal flows and attack flows are between RED and FRED. The results indicate that FRED outperforms CHOKe and RED in throttling LDoS attacks, while RED failed to mitigate LDoS attacks effectively. Through CHOKe can drop the attack flow's packets effectively, it also drops more normal flow's packets.

Throughput: Fig. 3 depicts the throughput of the normal flows under three queue mechanisms. The throughput is the maximum using FRED, while using RED, the throughput is the minimum, and decreases when time elapses. In case of CHOKe, the throughput is between the former two, and its distribution is similar with FRED.

Packet delay: As Fig. 4 shows, FRED has the maximum average packet delay, while RED has the minimum. Conversely, FRED has the minimum delay jitter while RED has the maximum delay jitter. The packet delay and its jitter of CHOKe are still between them.

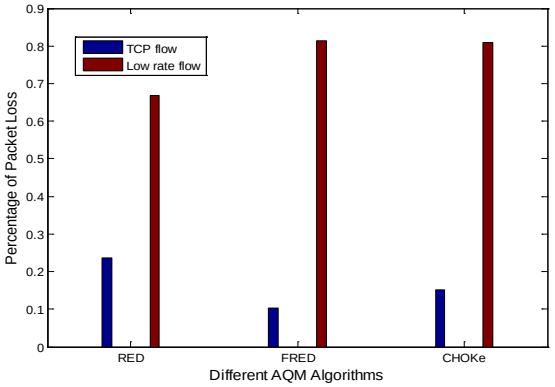

Fig 2. Packet Loss Rate.

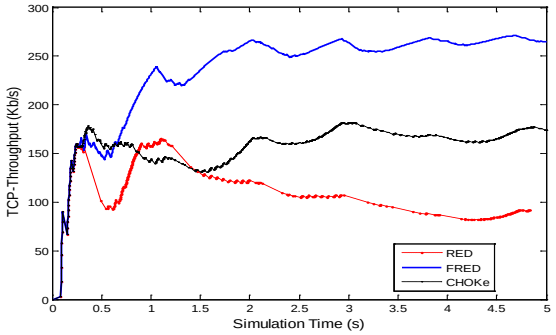

Fig 3. TCP Throughput Comparison.

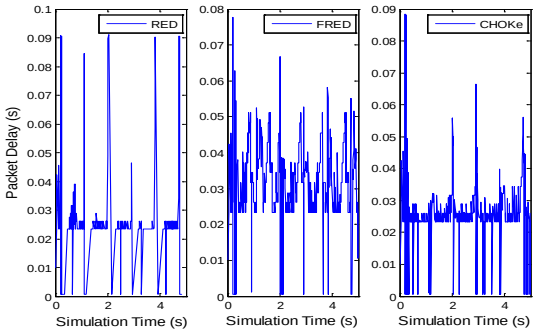

Fig 4. Packet Delay.

4. Similation Results

Research on detection and defense against LDoS attacks is still immature. According to the instant high rate and high intensity of LDoS attacks, this paper investigates using fair queue management mechanism to weaken the impact of the attacks. The simulation results show that both FRED and CHOKe are able to reduce the effect of the attacks in various degrees. FRED outperforms CHOKe in throttling LDoS attacks, but it is slightly inferior to CHOKe in time performance. Comparing with other detection and defense methods, Countering against LDoS attacks by fair AQM does not require modification of existing TCP protocols. Moreover, the overhead required for routers is less.

Acknowledgments

We would like to acknowledge support for this work from the State Key Program of NSFC-Guangdong Joint Funds (U0735002), the National High Technology Research and Development Program of China (2007AA01Z449) and NSFC General Project (60970146).

References

[1] A Kuzmanovic, EW Knightly. Low-rate TCP-targeted denial of service attacks. //Proceedings of ACM SIGCOMM 2003, Karlsruhe, Germany, 2003:75~86.
[2] A Kuzmanovic, EW Knightly. Low-rate TCP-targeted denials of service attacks and counter strategies. IEEE/ACM Transactions on Networking, 2006, 14(4):683~696.
[3] X Luo, RCK Chang. On a new class of pulsing denial-of-service attacks and the defense //Proceedings of NDSS 2005, San Diego, CA, 2005.
[4] M Guirguis, A Bestavros, I Matta. Exploiting the transients of adaptation for RoQ attacks on Internet resources. //Proceedings of ICNP 2004, Berlin, Germany, 2004:184~195.
[5] M Guirguis, A Bestavros, I Matta, et al. Reduction of Quality (RoQ) attacks on Internet end-systems. //Proceedings of IEEE INFOCOM 2005, Miami, Florida, 2005, 2:1362~1372.
[6] H Sun, JCS Lui, DKY Yau. Distributed mechanism in detecting and defending against the low-rate TCP attack. Computer Networks, 2006, 50(13):2312~2330.
[7] Y Chen, K Hwang. Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. Journal of Parallel and Distributed Computing, 2006, 66(9):1137~1151.
[8] B Braden, D Clark. IETF RFC2309 Recommendations on queue management and congestion avoidance in the Internet. 1998.
[9] S Floyd, V Jacobson. Random early detection gate way for congestion avoidance. IEEE/ACM Transactions on Networking, 1993, 1(4):397~413.
[10] D Lin, R Morris. Dynamics of Random Early Detection. ACM SIGCOMM'97, 1997, 127~137.
[11] R Pan, B Prabhankar, K Psounis. CHOKe: Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. IEEE INFOCOM 2000, Mar, 2000, 2:942-951.
[12] Network Simulator version 2 (NS2). http://www.isi.edu/nsnam/ns/.
[13] K Thompson, GJ Miller, R Wilder. Wide-area Internet traffic patterns and characteristics. IEEE Network, 1997, 11(6):10~23.