RSFB: a Resilient Stochastic Fair Blue algorithm against spoofing DDoS attacks

Changwang Zhang*, Jianping Yin, and Zhiping Cai
School of Computer Science, National University of Defense Technology, Changsha, Hunan, China
*E-mail: mleoking@gmail.com

***Abstract*— The existing Active Queue Management (AQM) algorithms, including the fairness-aimed ones, are notably vulnerable to spoofing DDoS attacks. We propose a Resilient Stochastic Fair Blue (RSFB) algorithm against spoofing DDoS attacks. The basic idea behind RSFB is to record the responsive normal TCP flows and rescue their dropped packets. Simulations and analysis show that the RSFB algorithm is highly robust and can fully preserve the TCP throughput in the presence of spoofing DDoS attacks.[1]**

I. INTRODUCTION

In order to combat congestion and improve network performance, quite a few Active Queue Management (AQM) algorithms such as Random Early Detection (RED) [1] and its variants are proposed in the past decades. And many recently proposed AQM algorithms, such as Stochastic Fair Blue (SFB) [2] and RED with Preferential Dropping (RED-PD) [3], detect and rate-limit non-responsive flows to enforce fairness amongst different flows. However, as demonstrated by our analysis and simulations that the existing AQM algorithms, including the fairness-aimed ones, are rather vulnerable to spoofing Distributed Denial-of-Service (DDoS) attacks.

Spoofing DDoS attacks are among the hardest ones to detect and track in DDoS attacks, which haven been indentified as a major thread to today's Internet services.

The focus of this paper is on building a resilient AQM algorithm against spoofing DDoS attacks. And we achieve this by proposing the Resilient SFB (RSFB) algorithm, which improves the SFB algorithm. Simulations and analysis demonstrate that the RSFB algorithm is highly robust and can fully preserve the TCP throughput in the presence of spoofing DDoS attacks. The rest of this paper is organized as follows. Section II describes the RSFB algorithm in detail. Section III presents performance evaluation, followed by conclusions in Section IV.

II. RESILIENT SFB (RSFB)

A. Overview of SFB

SFB is an AQM algorithm for enforcing fairness among a large number of flows, which detects and rate-limits non-responsive flows [2]. SFB keeps a marking probability p_m for each incoming flow. At the same time an updating algorithm was proposed for p_m to i) make the non-responsive flow quickly drives its p_m to 1; and to ii) make the responsive flow keeps its p_m around 0. Thus, non-responsive flows are detected for their high p_m, and SFB rate-limits them by dropping their packets. Additionally, SFB employs Bloom Filters to record and update the state information of flows, which include the p_m of flows. This technical makes SFB a scalable means to enforce fairness amongst flows using an extremely small amount of state and buffer space. However, the Bloom filters technical also has its innate drawbacks. The state information of responsive flows could be polluted by non-responsive flows in the Bloom filters. SFB would collapse under spoofing DDoS attacks, which is demonstrated by our analysis and simulations. Thus, we propose the Resilient SFB (RSFB) algorithm against spoofing DDoS attacks.

B. RSFB

Fig. 1 depicts the basic architecture of RSFB. RSFB consist of a normal SFB block and an extra Benign Flow Queue (BFQ) block. The basic idea behind RSFB is to keep the responsive normal TCP flows in the BFQ and rescue their dropped packets.

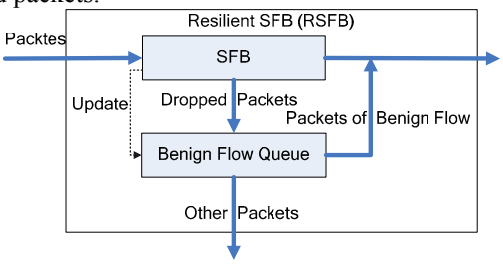

Fig. 1. Architecture of Resilient SFB (RSFB)

Fig. 2 shows the RSFB algorithm in detail, in which *pkt* denotes an incoming packet, *f* is *pkt*'s corresponding flow, and PQ represents the packet queue in the SFB algorithm. The RSFB algorithm is packet driven. When an incoming packet is arrived at the router, it is firstly fed to the SFB block and then we use extra two steps to post-process the packet.

In step 1, we update the Benign Flow Queue (BFQ). BFQ is a modified FIFO queue, of which the Delete operation can delete any element in the queue rather than the head one and the Insert operation can only insert an element at the tail of the queue. The flows with p_m equals 0 are considered as

[1] This work is supported in part by the National Natural Science Foundation of China (No.60603062 and No.60903040) and Natural Science Foundation of Hunan Province (06JJ3035).

benign flows and further inserted into the BFQ. In step 2, we rescue dropped packets from benign flows. After detecting a dropped packet *pkt* from a benign flow, we strive to insert it back to the packet queue (PQ). If PQ is not full, we simply insert *pkt* into PQ. And if PQ is full, we try to drop a packet from non-benign flows and then insert *pkt* into PQ.

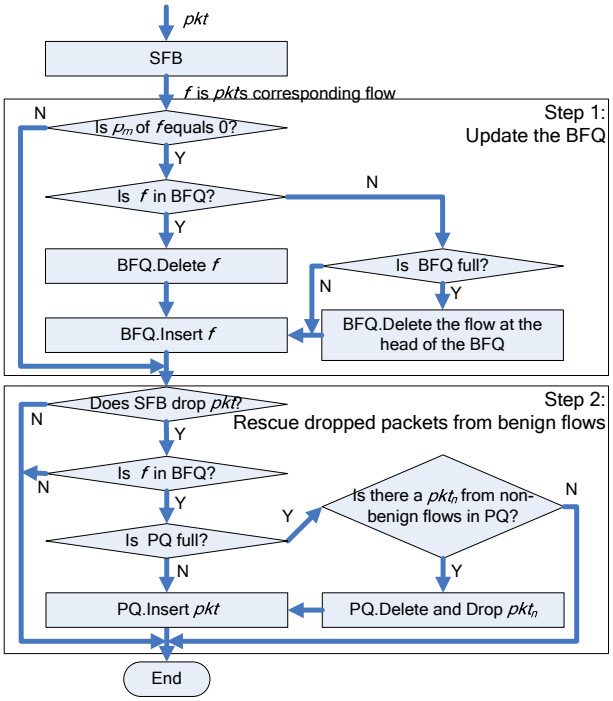

Fig. 2. Flow diagram of Resilient SFB (RSFB)

The observation which drives RSFB is that spoofing DDoS attacks quickly drive p_m to a high value (bigger than 0) for all the flows that SFB maintains in Bloom Filters. Recall that we only insert those flows whose p_m equals 0 into the BFQ. When a spoofing DDoS attack is attacking, there is hardly any new flow whose p_m equals 0 and then can be inserted into the BFQ. The above facts help us to isolate the BFQ from spoofing DDoS attack flows. Thus our algorithm is effective in rescuing benign flows and can significantly improve the performance of TCP when the router is under a spoofing DDoS attack.

III. Performance Evaluation

In this section, we use NS-2 simulator [4] to conduct a set of simulations to evaluate the performance of the proposed RSFB algorithm in the presence of spoofing DDoS attacks. Several other AQM algorithms include RED [1], RED-PD [3], SFB [2] (with code provided by [5]), and DropTail are used in the comparison.

Fig. 3 shows the experimental topology. The queue size of the bottleneck link is 50 packets. AQM algorithms are used on the bottleneck queue, and other queues use DropTail. A TCP (*Newreno*) based FTP flow with packet size of 1000 bytes is generated from each user (User 1 to User 30). Spoofing DDoS traffic is generated from Attacker 1 to Attacker 20 by sending UDP packets with packet size of 50 bytes and source address spoofed. The size of the BFQ in RSFB is empirically set as 50 flows. And the other parameters of the AQM algorithms are all NS-2 default values. Let R_a denotes the attack traffic rate of each attacker. We vary R_a from 0 Mpbs to 0.5 Mbps to perform a set of experiments.

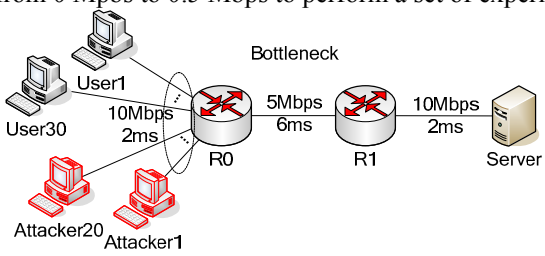

Fig. 3. Experimental topology

The experimental results are shown in Fig. 4. The results show that the RSFB algorithm is highly robust and can fully preserve the TCP throughput in the presence of spoofing DDoS attacks. The results also confirm that the existing AQM algorithms, including the fairness-aimed ones (SFB and RED-PD), are notably vulnerable to spoofing DDoS attacks. Especially, the RED-PD algorithm failed to finish the simulations when R_a>0.25, because it can not handle so much flows generated by spoofing DDoS attacks.

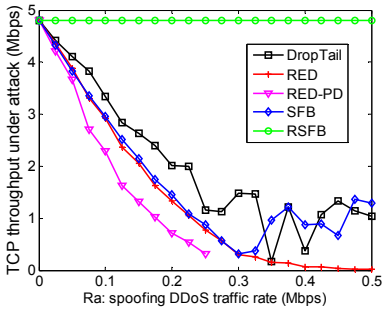

Fig. 4. TCP throughput under spoofing DDoS attacks

IV. Conclusions

We have proposed a Resilient SFB (RSFB) against spoofing DDoS attacks in this paper. Simulations and analysis show that the RSFB algorithm (i) is highly robust, and (ii) can fully preserve the TCP throughput under spoofing DDoS attacks.

References

[1] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397-413, 1993.

[2] F. Wu-Chang, D. D. Kandlur, D. Saha, and K. G. Shin, Stochastic fair blue: a queue management algorithm for enforcing fairness, in *Proceedings of IEEE INFOCOM*, 2001.

[3] R. Mahajan, S. Floyd, and D. Wetherall, Controlling high-bandwidth flows at the congested router, in *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2001.

[4] S. McCanne and S. Floyd, The Network Simulator - ns-2, in *http://www.isi.edu/nsnam/ns/*, 2008.

[5] Contributed Code - Nsnam, in *http://nsnam.isi.edu/nsnam/index.php/Contributed_Code*, 2009.