Reproducing the Results of "Distributed Denial of Service Attacks"

Ashkan Aghdai

May 1, 2016

Abstract

Here we reproduce Figures 4 and 5 of Distributed Denial of Service Attacks paper. The objective of this experiment is to show how effective Distributed Denial of Service (DDOS) attacks are under two different queueing schemes: Drop Tail and Random Early Detection. It shows whether or not legitimate users can obtain desired service while a DDOS attack is in process. Results are verified using a GENI testbed.

1 Introduction

A Denial of Service (DoS) attack is defined as an attempt to make a machine or a resource unavailable to its legitimate users [2]. When we have more than one source attacing the target, it is called a Distributed DoS (DDoS) [5]. Typically, attackers try to take control of a lot of unique IP addresses (Zombie Machines) and use them to exhaust a resource either on the target machine, or the network that connects legitimate users to the target machine/s.

In this experiment, we perform a simple DDoS attack. As shown in Figure 3, a client tries to reach a service through a single hop network with a router connecting it to the server. Assuming that an attacker takes control of a set of zombie machines connected to the same router, we simulate how he/she can make the service unavailable to the client by targeting the router and overwhelming its queue. As an outcome, packets will be droped while the attack is in progress and the client will experience a considerable drop in its bandwidth. Therefore the key metric in this experiment is the rate at which client communicates with the server while the attack is in progress. A rate of zero indicates no connectivity and a rate drop indicates service degradation.
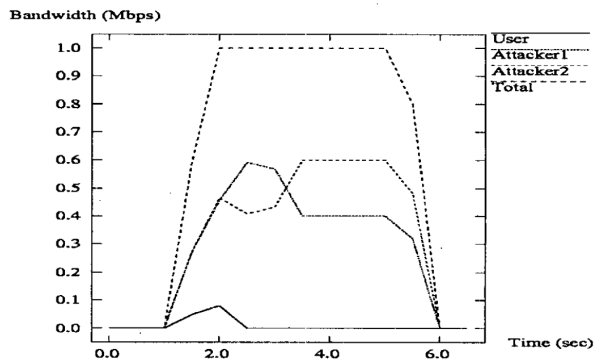

Figure 1: User's bandwidth during the attack using a DropTail queue [3], Figure 4 of [5].
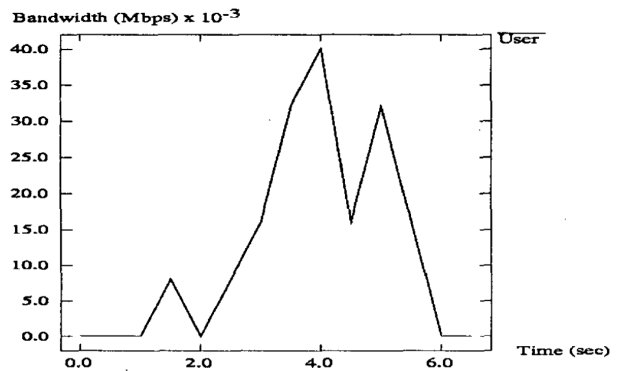

Figure 2: User's bandwidth during the attack using a RED queue [4], Figure 5 of [5].
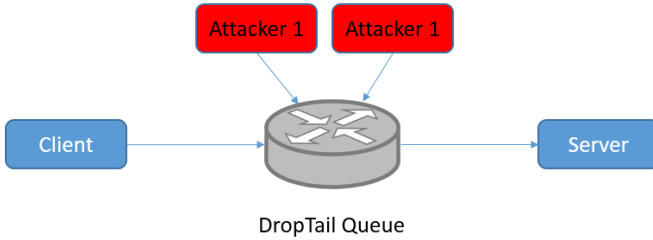
Figure 3: A simple DDoS attack.
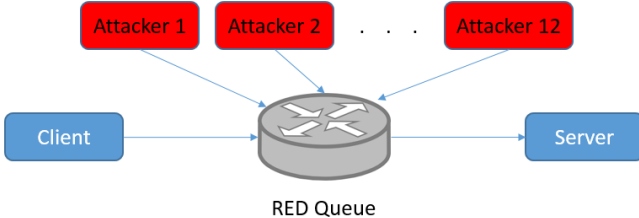

Figure 4: Architecture for DropTail queue.

Figure 5: Architecture for RED queue.

2 Experiment Design

Similar to the original experiment, the client establishes a constant bit rate (CBR) UDP stream to the server. To verify Figure 4, we show that only 2 zombie machines, as in the figure 4, sending CBR traffic are sufficient to disrupt the service when router implements DropTail queueing policy. Figure 5 shows how queues with Random Early Detection (RED) policy perform under a DDOS attack. However, in this scenario we need more zombie machines, as in figure 5, to disrupt the service. Results from the original paper show that unlike DropTail, the client rate will be decreased rather than going down to zero. It is still considered to be a successful attack as the client experience noticeable service degradation.

Since the authors have not specified the parameters, setting the proper parameters is the toughest part for this project especially for the second part where we experiment with RED, since configuring RED is known to be difficult when it comes to setting proper parameters [4].

2.1 Choosing the Right Parameters

For the DropTail queue, we know that the queue rate should be smaller than sum of the clients and two attackers' rate. Therefore for NS2 simulation I am sticking to CBR rates 0.1Mbps for client and 0.6Mbps for zombie machines using 500B packets all of which to be aggregated on a queue with rate 1Mbps. Queue size will also be set to the bandwidth delay product (assuming 1ms RTT): $1Mbps * 1ms = 1Kb = 125B$. For GENI testbed, we can use tc, where we stick to parameters from lab2 with limited buffer size according to the average RTT of given VMs using bandwidth-delay product (i.e. around 125B per milli second of RTT).

For RED experiment, we have to set 5 parameters: buffer space, min_{th}, max_{th}, max_p, and w_q. RED paper suggests max_{th} to be set at least twice as min_{th}, author of RED in answer to another question [1] suggests a value of 3 times as a rule of the thumb. We will stick to the same ratio

Parameter	DropTail Simulation	RED Simulation
Number of Attackers	2	12
Total Attack Bandwidth	1.2Mbps	9Mbps
Queue Size	125 B/msRTT	125 B/msRTT
Client's Desired Rate	0.1Mbps	0.1Mbps

Table 1: Related parameters.

in this experiment. She also suggests a value of 0.1 for max_p. min_{th}'s default value in NS2 is 5 packets, for this experiment we set it to 2500B, 5 times our packet size. w_q should be at least 0.001, NS2 uses 0.002 by default.

To summarize, since parameters are not specified by DDoS paper, my strategy for obtaining the results from RED is to stick to rule of the thumb values suggested by the author as well as NS2 defaults. If I fail to reproduce results using these values, I will report how and why I changed them to obtain results closer to those of the original paper.

Table 1 summarizes the parameters for the two experiments.

The workload on this experiment, both at the client and zombie machines, consists of synthetic traffic of CBR UDP with packets of 500 bytes. For the testbed we will produce this traffic using D-ITG.

To conclude the experiment we will analyze the data by computing average client bandwidth and packet drops against variable total rate of attack. For each factor we plot the mean value of five replications and indicate the range of the mean across replications.

References

[1] http://www.icir.org/floyd/REDparameters.txt. Accessed: 2016-03-01.

[2] Steven M Bellovin. Security problems in the tcp/ip protocol suite. *ACM SIGCOMM Computer Communication Review*, 19(2):32–48, 1989.

[3] Dimitri P Bertsekas, Robert G Gallager, and Pierre Humblet. *Data networks*, volume 2. Prentice-Hall International New Jersey, 1992.

[4] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *Networking, IEEE/ACM Transactions on*, 1(4):397–413, 1993.

[5] Felix Lau, Stuart H Rubin, Michael H Smith, and Ljiljana Trajković. Distributed denial of service attacks. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 3, pages 2275–2280. IEEE, 2000.