QoSoDoS: If You Can't Beat Them Join Them

Moti Geva and Amir Herzberg
Department of Computer Science, Bar Ilan Universiry, Ramat-Gan 59200, Israel

Introduction

The Internet is a Best-Effort delivery network, that is, it provides no guarantee for message delivery and no bounds on delay. If the network is congested routers drop arbitrary packets instead of transmitting them to their destinations. An example for malicious exploitation of Best-Effort behavior is DoS attacks which are commonly directed at this router behavior. ***Flooding DoS attacks*** are carried out by an attacker who congests a victim host by flooding it with futile messages, hence making routers along the way drop most of the legitimate traffic before it reaches the victim server. In many cases, DoS attacks and especially network flooding attacks are carried out using multiple zombie hosts and are referred to as ***Distributed DoS (DDoS) attacks***. These kind of attacks are usually considered hard to confront.

QoSoDoS takes a novel approach to assure ***QoS over DoS-prone networks*** (i.e. Best-Effort networks), which can coexist along with most current solutions. Instead of identifying and preventing the attack itself, QoSoDoS assures QoS by applying an automatic repeat request (ARQ) paradigm in which it assures, in a high probability, the successful delivery of the message.

Motivation and Rationale

There are many types of mission critical services which require high quality of service in terms of delivery, but are tolerant to a large delay and jitter, such services can be observed in many batch, offline and background transaction, as well as in financial and other contractual obligations in which time limits are within the order of minutes, hours or even days.

We assume that both QoS and Best-Effort models have hidden probabilities in which QoS assure delivery in high probability, denoted P_Q, and Best-Effort in low probability (DoS network), denoted P_D.

$$0 \lesssim P_D \ll 1 \text{ and } 0 \ll P_Q \lesssim 1$$

To make a DoS network assure QoS in a higher probability we need to retransmit each packet more than once. Each sent packet increases the probability for packet successful delivery. Generally, if n packets were sent and the delivery probability for each single packet is P_D, then the probability P_Q that some packet was accepted by the destination host is $P_Q = (1-P_D)^n$, hence,

$$n = \left\lceil \log_{(1-P_D)}(1-P_Q) \right\rceil$$

From DoS To QoS

In order to describe a DoS network based ***Leaky Bucket model*** we assume that our DoS network assures QoS, in a ***Latency-Rate Server*** (LR-server) service model, in which we get a worst case latency L_D and worst case rate R_D, with low delivery probability P_D, i.e. high probability for packet loss. Note that $R_D \le Bandwidth$. Table 1 compares DoS network's and QoSoDoS's parameters. P_D, P_Q, R_D, L_D, R_Q, B_Q and L_Q are configurable. By using packet retransmission, with a pre-defined deadline, it is possible to ***assure QoS over DoS networks*** while assuming that the DoS network can assure a low probability for packet delivery. We believe that this assumption is reasonable and can be considered valid for the Internet.

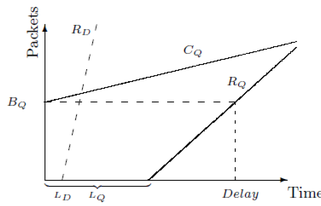

Figure 1: QoSoDoS delay analysis. L_D and R_D (dashed) are the DoS network's service latency and rate respectively, which are assured in low probability (P_D). L_Q and R_Q are the latency rate assured parameters in high probability (P_Q). Packets are transmitted in an average rate of at least $R_Q = \frac{R_D}{\alpha \cdot n}$, where $\alpha \ge 1$ is a transmission rate relaxation parameter, and $n \gg 1$ is the number of required packet retransmissions to assure packet delivery in probability P_Q. B_Q and $C_Q \le R_Q$ are the assured leaky bucket parameters.

Parameter	DoS	QoSoDoS
Latency	L_D	$L_Q \ge L_D$
Rate(R)	R_D	$C_Q \le R_Q = \frac{R_D}{\alpha \times n},\ \alpha \ge 1$
Burst(B)	B_D	B_Q
Delay	$L_D + \frac{B_D}{R_D}$	$L_Q + \frac{B_Q}{R_Q}$
Probability	$0 \lesssim P_D \ll 1$	$0 \ll P_Q \lesssim 1$

Table 1: Comparison between DoS network (Best-Effort) and QoSoDoS parameters. The table exhibit bounds for each parameter (see figure 1). Latency and Rate are the LR-Server service parameters and Rate, Burst and Delay are the Leaky Bucket parameters. Probability is the probability for a packet to be successfully delivered to its destination. $n = \lceil \log_{(1-P_D)}(1-P_Q) \rceil$ is the number of retransmissions required to assure successful delivery in probability P_Q while transmitting packet in probability P_D.

Relaxing Transmission Rate

To prevent QoSoDoS from becoming a source for DoS by itself, QoSoDoS prefers transmitting packets using TCP. It does so by initially starting transmission using TCP. In addition QoSoDoS refrains from transmitting at the channel's maximal rate, R_D. Instead, QoSoDoS assures QoS rate of $R_D/(\alpha*n)$, $\alpha>1$. This allows a more relaxed transmission as described in figure 2 as well as resuming TCP after the DoS attack is over.

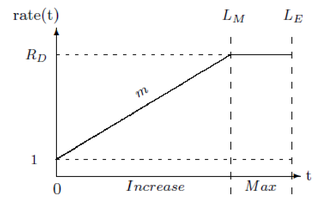

Figure 2: QoSoDoS single message transmission rate over time. The above example describes a linear rate increase heuristic. Different heuristics, such as exponential increase, are also applicable. 0 and L_M are the beginning of increase and maximal rate periods respectively, and L_E is the deadline, where $L_E = \frac{1}{R_Q}$ (see figure 1), by which all n retransmissions of a packet must end as described in section 2.3. m is the slope.

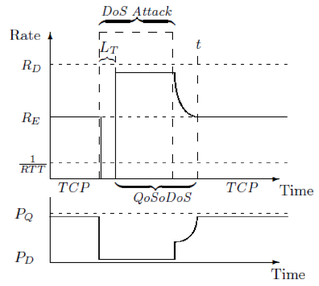

Figure 5: QoSoDoS Average rate vs. time (macro view). Top figure represents rate vs. time and the bottom figure is single message effective delivery probability (P_E). In the top figure, the solid line represent the average rate by which legitimate clients transmit packets. R_E is the effective average rate by which TCP would transmit packets under congestion control (w/o attack). At the beginning of the DoS attack TCP practically stops transmitting data. L_T is QoSoDoS's TCP-timeout interval, which, when expires, results in the abortion of TCP and transmission rate increase by QoSoDoS. When the DoS attack is over, QoSoDoS starts a recovery period. At t QoSoDoS concludes it can resume TCP.

Experimental Results

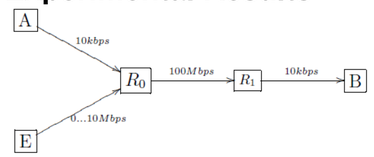

Figure 9: Experiment setup. A is legitimate host trying to be serviced by B. E is a flooding DoS attacker. Each host A, E, and B resides in a different subnet, namely 10.0.0.1/24, 10.0.1.1/24 and 10.0.2.1/24 respectively. R_0 is a router with a 100Mbps rate and is connected to R_1 which is a shaping router with 10Kbps. Both A's and E's transmission rate is shaped to simulate various network congestions. We used a small rate for A, so we could simulate large attacks ratios (up to 0.001), with assurance that the Ethernet doesn't have any affect on the results.

	E		P_D (A/(E+A))		$Rate$	$P_{E'}$
10	Mbps	1/1000	(0.1%)		5.7bps	0.087%
1	Mbps	1/100	(1%)		36.7 bps	0.5%
100	Kbps	1/10	(10%)		552.6 bps	17%
10	Kbps	1/1	(50%)		4730 bps	40.7%
1	Kbps	10/1	(91%)		9912 bps	-
0.1	Kbps	100/1	(99%)		10006 bps	-

Table 3: A/E vs. Delivery rate. A transmission rate is contant 10Kbps while E's changes. The rate is the average packets per seconds. A/E ratios of 0.1% to 10% are pure QoSoDoS results, in which TCP was not resumed. In the one-to-one ratio (50%) 94.6% of the result is made by QoSoDoS while 5.4% is made by resumed TCP. The two last results of ratios 91% and 99% are a purely result of TCP.

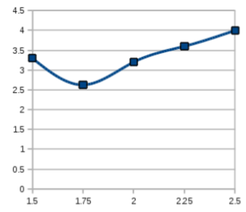

Figure 10: TCP resuming time vs. α. In the above example, the attacker, E, repeatadly oscillates between a 10 seconds period of transmitting at 10Mbps and a 10 seconds period of 5Kbps. In $\alpha \le 1.5$, TCP was not resumed. The minimal value for α was found when $\alpha = 1.75$ which took approximately 2.6 seconds to resume TCP. In $\alpha > 1.75$, the average transmission rate of the QoSoDoS client is lower, and therefore takes longer time period to gather k ACK packets and get $P'_E \ge \theta$.

Citations

[1] Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, volume 2050 of *Lecture Notes in Computer Science*. Springer, 2001.

[2] Jelena Mirkovic and Peter L. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. Computer Communication Review, 34(2):39–53, 2004.

[3] Jelena Mirkovic and Peter L. Reiher. D-ward: A source-end defense against flooding denial-of-service attacks. IEEE Trans. Dependable Sec. Comput., 2(3):216–232, 2005.