EFFECTIVENESS OF RATE-LIMITING IN MITIGATING FLOODING DOS ATTACKS

Jarmo V. E. Mölsä
Networking Laboratory
Helsinki University of Technology
P.O. Box 3000, FIN-02015 HUT
Finland
email: jarmo.molsa@hut.fi

ABSTRACT
This paper investigates the effectiveness of rate-limiting in mitigating TCP-based flooding Denial of Service (DoS) attacks. Rate-limiting is used as a DoS defense mechanism to discard a fraction of incoming attack packets. Part of legitimate traffic is, however, mis-detected as attack traffic. The main contribution of this paper is to find out how much a DoS attack can be rate-limited without any undue penalties for those legitimate TCP flows, which are mis-detected as attack traffic. The research methodology is based on analyzing the TCP throughput in a simulated network where packet-loss is one-way due to rate-limiting of incoming packets. Empirical measurements in a small network are used to verify the simulation results.

KEY WORDS
Internet security, Denial of Service, Rate-limiting, TCP throughput.

1 Introduction

Flooding Denial of Service (DoS) attacks are part of everyday life in the Internet [1]. These attacks try to overwhelm a victim with unnecessary data preventing authorized access to resources or delaying time-critical operations. Intrusion Detection Systems (IDS) [2] can be used to detect DoS attacks. Reliable detection, however, is not always possible [3][4][5]. A well-managed IDS is able to detect many real attack flows (true positives), but it will also mis-detect some legitimate flows as attack flows (false positives). Regardless of this the first reaction against detected DoS attacks must be automatic, because human intervention is slow and attack characteristics can change rapidly in a distributed attack. An automatic reaction mechanism must at the same time try to avoid damages from attack traffic and restrict damages to legitimate traffic.

There are two widely known reaction mechanisms against flooding DoS attacks: filtering and rate-limiting [6]. In filtering all incoming packets of a flow are discarded, and in rate-limiting an incoming packet in a flow is discarded with a certain probability. As DoS traffic cannot be easily distinguished from legitimate traffic [7], filtering (blocking) can cause more damage to legitimate user traffic than rate-limiting, because blocking will completely prevent availability of services to those users, whose traffic matches the characteristics of attack traffic. Even though rate-limiting is well-known and referenced in many papers, its effectiveness in mitigating flooding DoS attacks has not been analyzed.

The main contribution of this paper is to analyze the effectiveness of rate-limiting as an automatic reaction mechanism against flooding DoS attacks when the usability of legitimate connections must be preserved. The research methodology is based on analyzing the throughput of a legitimate TCP flow as a function of one-way packet-loss rate in a simulated network. Maximum packet-loss allowed by legitimate flows also defines, how much a DoS attack can be mitigated. The results from simulations are verified with empirical measurements in a small test network.

In this paper all flows are expected to be TCP-based, because the vast majority (83% of packets [8]) of existing traffic and most of DoS attacks (94% of all recognized attacks [1]) are TCP-based. Also, congestion control of TCP is sensitive to packet-loss. The effect of packet-loss on TCP throughput has been studied for example in [9] and [10], but these studies consider only the loss of TCP data segments. There does not seem to be any studies about TCP throughput when packet-loss is one-way, and either TCP data segments or TCP acknowledgements are lost.

The contents of this paper is the following. Section 2 describes some related work about TCP throughput and rate-limiting. The following section explains the expected application area of rate-limiting including the major limitations. Next section suggests a possible structure for a rate-limiting system. Section 5 gives the simulation results, and Sect. 6 gives the empirical results. The final section concludes the paper.

2 Related Work

TCP throughput has been studied in [9], which gives a relatively simple model for the bandwidth BW of a sustained TCP connection, when packet-loss probability p is relatively small:

$$BW = \frac{MSS}{RTT}\frac{C}{\sqrt{p}} \ , \qquad (1)$$

where MSS denotes the Maximum Segment Size, RTT denotes the Round-Trip Time, and C denotes a constant. This model assumes that TCP avoids retransmission timeouts and always has a sufficient receiver window size. According to the measurements in [9], a TCP connection can withstand a packet-loss rate between 1–10%, depending on the parameters. A more accurate model for the TCP throughput is derived in [10].

Both of these above-mentioned models expect that only TCP data segments are lost. One-way packet-loss, where either TCP data segments or TCP acknowledgements are lost, is not considered by these models.

Rate-limiting as an automatic reaction mechanism against flooding DoS attacks has been studied in [7], which specifies an infrastructure called the Cooperative Intrusion Traceback and Response Architecture (CITRA). The test in [7] demonstrated the suitability of rate-limiting in a test environment, where legitimate and attack traffic were completely distinguishable, i.e. only attack traffic was rate-limited. Legitimate traffic was passed through without bandwidth or packet-loss penalties.

There do not seem to be any papers analyzing the resistance of TCP flows against inadvertent rate-limiting, when a legitimate flow is mis-classified as DoS attack traffic. This is the goal of this paper.

3 Application Areas for Rate-Limiting

Effective mitigation of a DoS attack without any damage to legitimate traffic is difficult. As stated in [7], DoS traffic cannot be easily distinguished from legitimate traffic, because sophisticated DoS tools generate a packet stream that resembles legitimate traffic. An attacker can also intentionally choose such traffic that maximizes damage to legitimate traffic, i.e. an indirect DoS attack based on the side-effect of an automatic reaction mechanism. In this case a countermeasure intended to protect a system from DoS attacks can turn out to be the vehicle for carrying out the DoS attack itself, which in practice means that an automatic reaction mechanism causes more damage than the attack traffic itself. Even in the case of true attacks (true positives) it may be difficult to build an efficient and reliable filter (identification information based on packet data) that matches only the detected attack traffic. Selection of an automatic reaction mechanism is thus a trade-off between the effective mitigation of DoS attacks and the damage to legitimate traffic.

To minimize damage to legitimate traffic and to make it more difficult to turn a countermeasure into an attack mechanism, the use of rate-limiting is preferred here instead of blocking as an automatic early-reaction mechanism against DoS attacks.

The key parameter R in rate-limiting is the proportion of packets being discarded, i.e. $(1 - R) \times 100\%$ of the identified attack packets are passed through a router. The value of R has to be chosen so that legitimate traffic can withstand the packet-loss and that real attack traffic is reasonably dampened. Especially TCP-connections easily close, if R is too high.

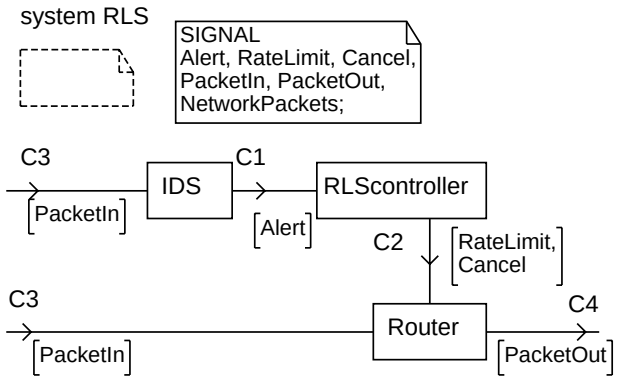

Figure 1. The SDL system diagram for a Rate-Limiting System (RLS) consisting of an IDS, an RLS controller, and a router with QoS-support.

The major application area of rate-limiting is a degradative (non-destructive) flooding attack, where a victim is overloaded with incoming packets. The attack traffic is expected to be TCP-based in this paper. Degradative flooding DoS attacks consume network bandwidth, processing power, disk space etc. Destructive flooding DoS attacks, which cause a permanent DoS condition e.g. by filling disks or crashing several target hosts, can possibly also be delayed enough, so that human intervention has enough time to prevent a total DoS condition. Rate-limiting can also slow down worm propagation, which is important in restricting the effect of fast spreading worms [11] [12].

There are two classes of DoS attacks against which rate-limiting is not an effective defense mechanism. First, logic DoS attacks [1] having a single target cannot be rate-limited, because even one packet can do harm e.g. by crashing, infecting, or compromising a host. Second, very high-bandwidth flooding attacks would require rate-limiting with a high value of R, which in practice will approach complete blocking with R =1.

4 A Suggested Structure and Requirements for a Rate-Limiting System

This section describes a possible structure and requirements for a Rate-Limiting System (RLS). An RLS is an early-reaction system, which automatically reacts to detected DoS attacks.

The Specification and Description Language (SDL) system diagram for an RLS is depicted in Fig. 1. An RLS consists of an IDS, a control mechanism (RLS controller), a distribution mechanism, and Quality of Service (QoS) support in RLS-compatible routers. One or more IDSes can reside either in access network links or in end-hosts. IDSes send their DoS alerts to an RLS controller, which creates a filter matching the detected attack traffic. The filter may

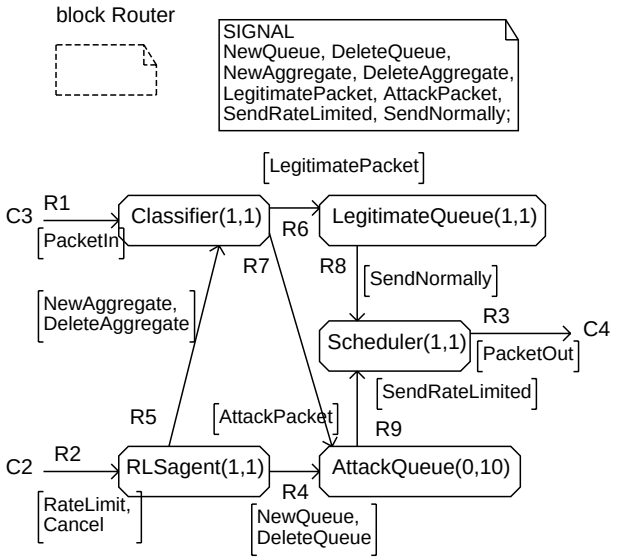

Figure 2. The SDL block diagram for a router in an RLS.

match both legitimate and real attack traffic due to problems in reliable detection and creation of exact filters. The filter data has to be distributed to upward routers nearer the attack source, e.g. by using the Pushback messages [13].

The SDL block diagram for a router in an RLS (RLS router) is shown in Fig. 2. The main parts of an RLS router are an RLS agent, a classifier, a legitimate queue, attack queues, and a scheduler. An RLS agent controls rate-limiting in a router by using standard QoS building blocks, i.e. a classifier, different queues, and a scheduler. An RLS agent receives messages from an RLS controller. These received messages include the filter data identifying the attack traffic. An RLS agent first creates a new queue for the attack aggregate and then installs the received filter in the classifier. All packets traversing an attack queue are discarded with a probability of R. All legitimate packets go through the legitimate queue without any added packet-loss. The scheduler will transmit all legitimate packets and those attack packets that survive the rate-limiting. The maximum number of attack aggregates has to be limited to prevent routers from being overwhelmed with the number of queues to be handled.

4.1 Requirements for Actual Rate-Limiting in Routers

The RLS requires a basic QoS support from routers, which are expected to classify incoming packets, direct them in different queues, use Active Queue Management (AQM) [14], and finally schedule packets to be sent to an outgoing link.

Each attack aggregate is directed to its own queue according to the filter data (like IP addresses, port numbers, and protocol numbers) describing the main characteristics of a specific aggregate of attack traffic. Because the objective is to discard packets from an attack aggregate with a probability of R, an AQM mechanism can be used. The scheduler must give each queue a fair share of bandwidth, because attack queues may also contain legitimate packets. If transmission capacity (link bandwidth) is not the bottleneck, then the scheduler is not a critical point, and there is freedom in selecting a scheduler algorithm and setting its parameters.

The use of an AQM mechanism for rate-limiting is preferred here instead of a bandwidth-allocating scheduler, because an AQM mechanism can discard fairly accurately a certain proportion of packets traversing a queue. An AQM mechanism does not even need to know the bandwidth of an attack aggregate. A scheduler, on the contrary, needs a reliable estimate of the bandwidth of the attack aggregate, which is not feasible considering the properties of real, fast varying DoS attacks. A scheduler can limit the bandwidth allocated to a queue, but this kind of rate-limiting may not even mitigate an attack at all, if the bandwidth of an attack aggregate falls below the initially allocated bandwidth. The AQM mechanism chosen should share the bandwidth as fairly as possible, so that the non-responsive attack traffic does not steal bandwidth from the responsive legitimate traffic also belonging to the attack aggregate.

5 Simulation Results

The effect of one-way packet-loss on TCP throughput was investigated by simulating a transmission of a large file with the File Transfer Protocol (FTP). The simulated network included one RLS router to mitigate flooding DoS attacks against the server. The ns-2 network simulator was used in these simulations.

5.1 The Setup of the Simulator

The topology of the simulated network is shown in Fig. 3. The legitimate FTP traffic is sent between the FTP client and the FTP server which are attached to the Client router and the Server router, respectively. The RLS router in the middle implements the rate-limiting and the related one-way packet-loss as an ns-2 error model, which uniformly discards a specific fraction (R) of packets being sent to the Server router. The underlying TCP for FTP applications is of type Reno (TCP/Reno) with a packet size of 1460 bytes. The links between the Client router and the Server router have a bandwidth of 2 Mbps and a delay of 10 ms. The Attack traffic router is connected to the RLS router through a 500 kbps link with a delay of 20 ms.

A Distributed DoS (DDoS) attack is simulated in the network with a group of 50 DDoS sources. Each DDoS source sends a large file with the FTP protocol to the FTP server. Attack traffic is thus sent over the TCP protocol (TCP/Reno). These DDoS sources are able to create at most 500 kbps of background traffic due the link bandwidth at the Attack traffic router.

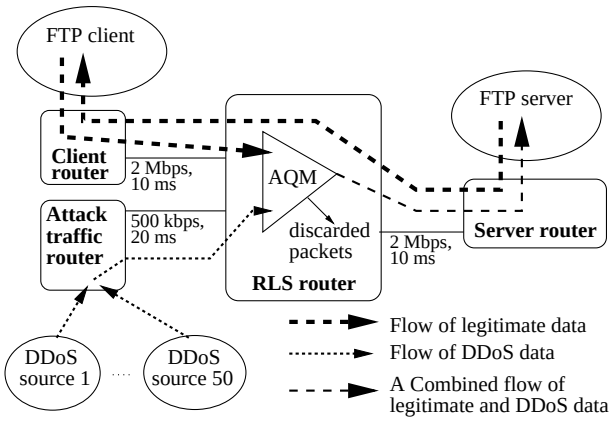

Figure 3. The topology of the simulated network. The dotted lines indicate the flow of data. The AQM in the RLS router discards a specific fraction of packets being sent to the FTP server. No packets are discarded by RLS in the reverse direction.

The flow of data packets is shown with dotted lines in Fig. 3 (the flow of TCP acknowledgements from the FTP server to DDoS sources is not shown in this figure). The FTP client either downloads a large file from the FTP server or uploads a large file to the FTP server. Both legitimate and DDoS FTP packets being forwarded to the Server router are discarded with probability of R at the RLS router by an AQM mechanism. The reverse direction for FTP traffic does not encounter any packet-loss by the RLS.

5.2 The Effect of One-Way Packet-Loss on TCP Throughput

The simulations consisted of the transmission of a very large file for 100 000 seconds. The amount of data transmitted during this time was calculated from the final TCP acknowledgement received by the sender.

Figure 4 shows the simulation results for file upload and download tests when no background DDoS traffic was present. Simulation results in Fig. 5 show the results of the file transfer tests during a DDoS attack.

The x-axis of these figures shows the packet discard probability R. The y-axis shows the average throughput during the whole 100 000 second simulation as bits per second (bps). The solid thick line indicates the throughput of file downloading, and the dotted thick line indicates the throughput of file uploading. The thin dotted line indicates the theoretical TCP throughput according to Eq. (1) (MSS=1460 bytes, RTT=40 ms, and C=0.45). Even though the theoretical curve is shown for the whole x-axis range, it is valid only with relatively small values of R.

These simulation results indicate that for file upload the one-way packet discard probability R must be below 0.1 for TCP to have a reasonable average throughput. File download, however, is able to withstand a packet discard

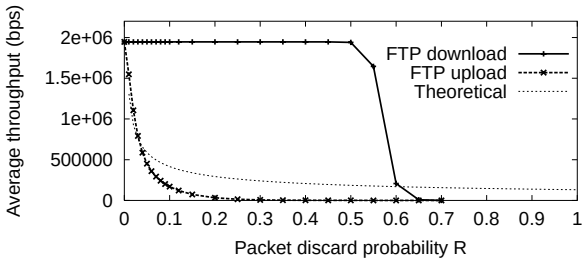

Figure 4. The average TCP throughput in the simulator. No background traffic was present.

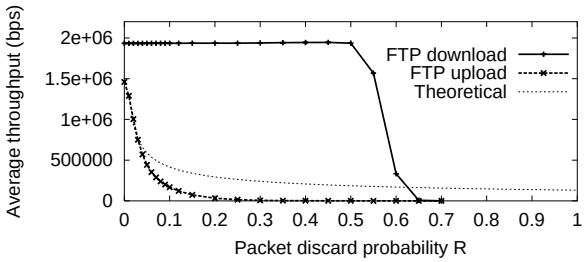

Figure 5. The average TCP throughput in the simulator. A flooding DDoS attack was in the background.

probability up to 0.5 before the average throughput starts to decline seriously.

The effect of the background DDoS attack is visible only in the throughput of file upload. When uploading a file the bandwidth of the network link from the RLS router to the Server router is shared with the DDoS attack traffic. Two competing types of traffic will share the bandwidth of this link, and less bandwidth is available for legitimate file uploading during a DDoS attack. File downloading is being sent in the reverse direction on this network link, and the DDoS attack does not consume the bandwidth of the link in this direction. Changing the TCP-based DDoS attack traffic to UDP-based (50 Pareto On/Off traffic sources) did not have any visible effect on these results. The local connection from the RLS router to the FTP server is assumed to provide the full bandwidth for both directions at the same time (e.g. by separate wires).

The shape of the upload throughput curve matches reasonably well with the theoretical curve. On the other hand, the shape of the download throughput curve differs quite much from the theoretical curve. This difference comes from the type of packets discarded by the RLS router, which applies rate-limiting only to the traffic being forwarded to the Server router. When downloading data from a server, only TCP acknowledgements experience increased packet-loss, but none of the TCP data segments suffer from forced packet-loss at the RLS router. The loss of an acknowledgement does not necessarily require a retransmission, because acknowledgements are incremental. Successive acknowledgements can recover information in

earlier lost acknowledgements. The loss of a TCP data segment, however, cannot be restored without a retransmission either through a fast retransmission (duplicate acknowledgements) or a timeout. File downloading is thus able to withstand a relatively high proportion of lost acknowledgements, because successive acknowledgements make it unnecessary to retransmit packets.

The theoretical model expects that only TCP data segments are lost with a certain probability. If only TCP acknowledgement packets are lost, the actual throughput curve has thus higher values than the theoretical curve.

5.3 Suitability of Rate-Limiting as a DoS Attack Mitigation Mechanism

The simulation results show that the effect of one-way packet-loss on TCP throughput is application-dependent. File downloading tolerates rate-limiting better than file uploading to a server protected with an RLS.

According to these results rate-limiting is a useful automatic reaction mechanism against flooding DoS attacks. Rate-limiting can mitigate an incoming DoS attack up to 50%, but still provide a reasonable service quality for those legitimate users mis-detected as attackers. This can be achieved even when attack and legitimate traffic cannot be distinguished at all.

The effect of random packet-loss inherent in real networks was not included in the simulator. Regardless of this, these simulation results show that rate-limiting disturbs information downloading much less than information uploading.

6 Empirical Results

A small test system was implemented to verify the simulation results. Simulation software does not include all real-life effects, like processor load. The goal is to see whether theory and practice match reasonably together.

The test network consists of three Linux hosts (kernel version 2.4.20). One host acts as an FTP client and another host acts as an FTP server. The third host acts as an RLS router between the server and the client. The RLS router implements a simple AQM mechanism, which discards packets always with a probability of R.

6.1 FTP Download and Upload Tests

The empirical tests consisted of downloading or uploading a 450 kB file from/to the server with FTP. The throughput indicated by the FTP client was recorded after each file transfer. A test was run approximately 15 times for each value of R. There was no background traffic during these empirical tests.

The RLS router was initialized with a specific value of R before any FTP throughput tests were run. The classifier in the RLS router was initialized so that the legitimate FTP traffic is treated as attack traffic. This made it possible to study the effect of rate-limiting on legitimate FTP traffic.

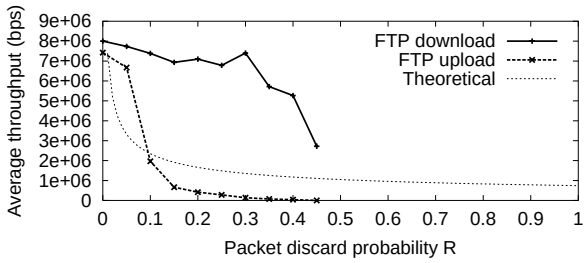

Figure 6. The average TCP throughput measured from the test system. No background traffic was present.

The empirical results for the FTP tests are shown in Fig. 6. The thick solid line indicates the measured average TCP throughput for an FTP download as a function of packet discard probability R. The thick dotted line indicates the measured average TCP throughput for an FTP upload. The thin dotted line indicates the theoretical TCP throughput as indicated by the Eq. (1) (MSS=1448 bytes, RTT=3.9 ms, and C=0.25).

As can be seen from this figure, the empirical results are approximately the same as the simulation results. FTP download tolerates rate-limiting much better than FTP upload. The maximum value of R is, however, lower than in the simulations. According to these empirical results FTP download tolerates a packet discard probability $R = 40\%$. Even though these empirical tests were rather short, they support well the simulation results.

6.2 Web Browsing Tests

To see whether the file transmission results are applicable to interactive traffic, web browsing was tried shortly in the test system.

Web browsing resembles file downloading because most of the data is transmitted on an HTTP connection from a web server. This direction does not suffer from forced packet-loss in the test system. Only request and acknowledgement packets experience increased packet-loss.

The first negative effects can be perceived around the packet-loss rate of $R = 0.3$, but the quality of web browsing remains acceptable up to the packet-loss rate of $R = 0.55$. These results are highly subjective, but indicate the common properties between web browsing and file downloading.

Rate-limiting seems to be a suitable DoS defense mechanism for mitigating flooding DoS attacks against WWW servers. Web browsing is an important application type, because many e-commerce sites are accessed only by web browsers. Also, well-known web sites have been a target for many published DoS attacks [15].

7 Conclusion

Flooding DoS attacks are part of everyday life in the Internet. As it is difficult to distinguish DoS traffic from legitimate traffic, a defense mechanism should have little or no negative effects on legitimate traffic. Rate-limiting is this kind of a defense mechanism, and it can mitigate any kind of flooding DoS attacks, like TCP-, UDP-, or ICMP-floodings. Even though rate-limiting is a widely referenced defense mechanism against DoS attacks, its effectiveness has not been analyzed. This paper used both simulations and empirical tests to evaluate effectiveness of rate-limiting in mitigating TCP-based flooding DoS attacks.

TCP throughput was analyzed in a simulated network which included a rate-limiting feature to mitigate flooding DoS attacks against a server. File uploading to the server was sensitive to rate-limiting, and tolerated a packet-loss rate of less than 10%. File downloading from the server, on the other hand, tolerated one-way packet-loss much better than file uploading. Downloading was able to tolerate a one-way packet-loss rate up to 50%. File downloading and web browsing are examples of applications, which seem to tolerate well the extra packet-loss from rate-limiting.

According to these results the effectiveness of rate-limiting is limited to decreasing the intensity of a TCP-based flooding DoS attack by up to 50%, when legitimate users mainly download data. This should be seen as a useful result, because attack mitigation is possible even when legitimate and attack traffic cannot be distinguisted at all. Rate-limiting can thus be used as a fast, automatic reaction mechanism to mitigate an attack without any undue penalties for legitimate traffic.

References

[1] D. Moore, G. M. Voelker, and S. Savage, Inferring Internet denial-of-service activity, *Proceedings of the 10th USENIX Security Symposium*, Washington, D.C., 2001.

[2] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, Network intrusion detection, *IEEE Network, 8*(3), 1994, 26–41.

[3] T. H. Ptacek and T. N. Newsham, *Insertion, evasion, and denial of service: eluding network intrusion detection*. Secure Networks, Inc., 1998.

[4] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation, *Proceedings of the DARPA Information Survivability Conference and Exposition*, 2000.

[5] P. Mueller and G. Shipley, Dragon claws its way to the top, *Network Computing*, August 20 2001, 45–67.

[6] A. Householder, A. Manion, L. Pesante, G. M. Weaver, and R. Thomas, *Managing the threat of denial-of-service attacks*. CERT Coordination Center, 2001.

[7] D. Sterne, K. Djahandari, B. Wilson, B. Babson, D. Schnackenberg, H. Holliday, and T. Reid, Autonomic response to distributed denial of service attacks, *Proceedings of Recent Advances in Intrusion Detection, 4th International Symposium*, Davis, California, 2001, 134–149.

[8] S. McCreary and K. C. Claffy, Trends in wide area IP traffic patterns - a view from ames internet exchange, *Proceedings of the ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, 2000.

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

[10] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, Modeling TCP throughput: A simple model and its empirical validation, *Proceedings of the ACM SIGCOMM conference*, Vancouver, Canada, 1998.

[11] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, Internet quarantine: Requirements for containing self-propagating code, *Proceedings of the IEEE Infocom*, 2003.

[12] M. M. Williamson, *Throttling viruses: Restricting propagation to defeat malicious mobile code*, HP laboratories, Bristol, Tech. Rep. HPL-2002-172, 2002.

[13] S. Floyd, S. Bellovin, J. Ioannidis, K. Kompella, R. Mahajan, and V. Paxson, *Pushback messages for controlling aggregates in the network*, 2001, internet draft draft-floyd-pushback-messages-00.txt, work in progress.

[14] M.-K. Chan and M. Hamdi, An active queue management scheme based on a capture-recapture model, *IEEE J. Select. Areas Commun., 21*(4), 2003, 572–583.

[15] L. Garber, Denial-of-service attacks rip the Internet, *IEEE Computer, 33*(4), 2000, 12–17.