

Tackling Congestion to Address Distributed Denial of Service: A Push-Forward Mechanism

Srinivasan Krishnamoorthy and Partha Dasgupta

Computer Science and Engineering Department
Arizona State University
Tempe AZ 85287

Abstract—Distributed Denial of Service attacks prevent legitimate users from accessing a target machine or the service a target machine provides. One common method of attack is overwhelming the target machine with a large volume of traffic. Thus, handling congestion indirectly leads to detection and recovery from Distributed Denial of Service attacks.

The Internet is an interconnected collection of Autonomous Systems. Every host on an Autonomous System connects to the Internet through an Access Router. Monitoring the rate of packets to and from a host, at the Access Router, helps in identifying Distributed Denial of Service attacks initiated at the host. Monitoring every Access Router leads to an effective Distributed Denial of Service prevention, but is infeasible. An alternative is a combination of Access Router monitoring and Intermediate Router monitoring with a novel Push-Forward mechanism that provides good defense within manageable deployment requirements. Push-Forward messages reduce the amount of traffic to monitor at the Intermediate Routers. Prototype testing and simulations of such a combination reveal good congestion detection and recovery time with very little performance overhead.

Index Terms—Network Security, Distributed Denial of Service.

I. INTRODUCTION

Denial of Service (DoS) attack is defined as an “explicit attempt by an attacker to prevent legitimate users of a service from using that service” [1]. Distributed Denial of Service (DDoS) attack is an extension to DoS where the attack originates from distributed sources. In DDoS attacks, one target machine is attacked from a large number of collaborating attack machines. DDoS attacks have become a prevalent networking issue that requires a distributed, coordinated detection and recovery mechanism. Detecting or preventing a DDoS attack is hard because the traffic looks normal; the attacking packets are no different from the normal packets.

A. Background

DDoS attacks can be classified as logic or brute-force attacks. Logic attacks exploit protocol vulnerabilities, and can be easily prevented by fixing the exploited bugs. Brute-force attacks use normal looking packets to overwhelm the target machine. Defending against brute-force attacks is tougher. There are many solutions to defend against these attacks. Based on their deployment location, defense mechanisms can be classified as Target-end, Intermediate network and Source-end solutions [2]. Target-end mechanism deployed near the victim can identify DDoS attacks easily, because they can view the aggregate attack traffic, but tracing back to source of the attacks is hard and the amount of traffic to analyze is large. Intermediate network solutions, deployed at the core network infrastructure, have trace back and deployment problems. Source-end solutions, deployed at the edges of the Internet (access networks), too have deployment problems; but tracing back to attack sources is easy and amount of traffic to analyze is minimal. We propose a solution to detect and recover from brute-force attacks that deploys at both the Source-end and Intermediate network.

B. Motivation

The Internet is a collection of Autonomous systems. Every host in an Autonomous System has to connect to the Internet through an Access Router, thus a natural location to deploy a Source-end solution. Since the traffic to be analyzed by the Access Router is less, a sophisticated mechanism can be deployed for Source-end solutions.

A survey conducted by Moore identifies 94% of the attack traffic as TCP based [3]. Thus traffic measurement between source and destination machines at the Access router provides a good indication of disproportionate flows. These flows can be congestion causing flows at the target machine and in turn can be attack traffic.

C. Related Work

There has been a lot of research in the area of detection, recovery, prevention and traceback of DDoS attacks. The mechanism in this paper describes a detection and recovery technique with the added advantage of easy traceback to the

sources of the attacks.

The classic way to prevent a network based attack is configuring a firewall [4] to filter incoming and outgoing traffic. Ingress / Egress filtering falls under the above mentioned methodology [5, 6]. In [7], Park and Lee propose a solution to DoS prevention using route-based filtering. Route-based distributed filtering uses routing information to determine if a packet is valid with respect to its source/destination addresses. Such a technique requires the routers to know the topology of the network and they fail to protect against more sophisticated attacks while restricting the network activities.

Flooding Detection System (FDS) [8], installed on leaf-routers for DDoS detection can classify the TCP control packets (SYN/ACK/FIN/RST) and the TCP data packets. By design, SYN and FIN packets are normally paired. But during a SYN attack, the number of SYN packets is very high compared to FIN packets in the leaf-routers to which the attack machine is connected. This approach can only detect SYN flooding.

WADeS [9] detects a DDoS attack using Wavelet methods. It captures the high bandwidth flows followed by computation of wavelet variance on the traffic, because the attack traffic would produce changes in the wavelet variance. It also combines the thresholds to enable attack detection.

ICMP Traceback [10] relies on ICMP traceback messages that are router-generated. These messages are used by the victim to reconstruct the attack path. Since the messages are generated with a low probability, messages could be dropped due to attack traffic congestion. This makes it hard for the victim to reconstruct the attack path.

There are additional IP based traceback mechanisms [11, 12, 13] that are used for DDoS attack traceback.

CenterTrack [14] is an overlay network, which tracks certain packets through IP tunnels and uses that traffic information to decide if there is an ongoing attack. They require special tracking routers and they are deployed within an ISP's network which limits their traceback to just the local network.

"Pushback" [15] introduces additional functionalities at routers to monitor the traffic at the routers. They are deployed near the target machines. Once an attack is detected, the attack traffic is identified and rate limited. This information is then sent upstream and upgraded routers act upon the pushback message. The disadvantage stems from the fact that the deployment is near the target and by the time the attack is detected the network is already flooded with attack traffic.

D-WARD is a Source-end solution to detect and recover from DDoS attacks. D-WARD proposes a pure Source-end monitoring system that detects and throttles outgoing attack traffic from a network that has deployed this system.

D-WARD selectively imposes rate limiting on offending traffic with a self-regulating reverse feedback system [16]. This solution requires heavy user involvement. It suffers from the deployment issue because it is a pure Source-end solution and it doesn't handle legitimate congestion problems.

MULTOPS is another Source-end solution that uses a data structure to keep track of the traffic pattern at a source router [17]. This data structure, Multi-Level Tree for Online Packet Statistics (MULTOPS), can be used with other detection mechanisms to provide improved results. MULTOPS is not a complete defense mechanism in itself; it can be used to complement our solution.

II. REQUIREMENTS AND GOALS

A. Requirements

Our research problem can be stated as "to address DDoS using congestion control and recovery mechanisms." Some of the features that we wanted our solution to exhibit are: a lightweight router based solution; a solution that does not require changes to end-user machines; a solution that is deployed incrementally at the Source end and Intermediate network and a solution that will contribute very little overhead at the Intermediate routers.

B. Goals

Our goals to provide an effective DDoS solution can be summarized as: a) Identify a victim's unavailability due to a congestion causing attack traffic, near the source of the congestion using Access Routers; b) Rate limit such congestion causing traffic at the Access Routers; c) Alert downstream intermediate routers about the congestion causing traffic using the Push-Forward mechanism and d) Rate-limit such traffic at the intermediate downstream routers to handle distributed congestion causing sources.

III. DESIGN

The above listed goals are realized using three main mechanisms in our solution. They are the a) Access and Intermediate Router Monitoring; b) Rate Limiting and c) Push-Forward mechanism.

A. Access and Intermediate Router Monitoring

For the Access Router monitoring, the router on a source (access) network is upgraded with some additional functionality to determine congestion at the victim. The idea is to measure the traffic to and from various destination machines and the source machines. We use this measurement to identify the unavailability of the destination machines and classify such machines as suspected victims. This classification is fairly simple if the traffic is bi-directional, since there is a defined ratio between the to and fro traffic

between the source and the destination. If the ratio is not maintained then we can classify the destination as a victim and the flow as congestion causing traffic. If the traffic is unidirectional, then we use a history mechanism to classify the flow as congestion causing traffic.

The Intermediate Routers handle traffic from a set of Access Routers. Thus the volume of traffic handled by Intermediate Routers is huge. Analyzing each and every packet that flows through the router puts a lot of overhead on the Intermediate Routers. Hence, Intermediate Routers analyze only traffic flows that are suspect. Information about suspected flows is received from Access Routers that have identified congestion causing traffic at their access networks. This information is propagated to the Intermediate Routers using Push-Forward messages.

B. Rate Limiting

The congestion causing traffic that is identified by the monitoring mechanism is rate limited. Rate limiting basically drops packets based on a predefined formula. The rate limiting is similar at both the Access and the Intermediate routers. Information about both dropped and forwarded packets are logged so that this information can be used as a feedback mechanism to decide if a flow is to be rate limited in the next interval.

C. Push-Forward mechanism

We have introduced a novel mechanism to handle distributed congestion causing sources. This mechanism addresses the deployment issues of a Source-end solution and reduces the overhead at the Intermediate routers.

Once an Access Router identifies potential attack traffic, it alerts the downstream routers using a Push-Forward message. This message alerts the downstream routers of specific suspect flows that have to be monitored. This takes care of flows that might originate from an access network that did not deploy our mechanism. Thus distributed attack sources or flows are caught at the Intermediate Router, which has upgraded.

Only Access Routers generate the Push-Forward message. The Push-Forward message is addressed to the suspected victim. Route prevalence in a network path studied by Paxson in [18], gives us the confidence that the Push-Forward message will follow the same path as that of attack traffic.

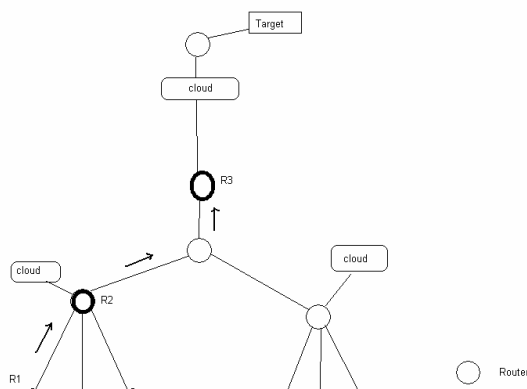


Fig. 1. System visualization.

The Push-Forward mechanism helps keep our design simple because the Access Routers need not know which downstream Intermediate Routers have upgraded. Push-Forward messages initiated monitoring only on upgraded Intermediate Router.

Fig. 1. shows the system visualization. The circles at the bottom level represent the Access Routers. Other routers are the Intermediate Routers. Darkened circles represent the upgraded routers. A1, A2, and A3 are the distributed attack sources. A1 is caught at the Access Router R1 because R1 has our upgraded mechanism. A1 generates the Push-Forward message to alert and hence initiate monitoring at the Intermediate Routers R2 and R3. The Push-Forward message is represented by the arrow marks. As a consequence of the Push-Forward message, A2 is caught at R2 and A3 is caught at R3.

IV. ARCHITECTURE

The four main modules to handle the various functionalities of the solution are detailed below.

A. Collection Module

This module logs traffic measurement statistics. The logged details are the destination IP, source IP, the protocol, the size of the packet, etc. Since the packet headers are already used for forwarding look-up, gleanig and logging the same information from the packets do not add substantial overhead.

B. Statistics Module

The Statistics module periodically parses through the log generated by the Collection module. The Statistics module classifies flows that are disproportionate as congestion causing flows. This decision uses two different methods based on the traffic type. If the traffic is bi-directional, the module looks for the ratio between the requests and responses between the source and the destination. If this ratio is not maintained, then it classifies the flow as congestion causing. If the traffic is unidirectional, then the module compares the flow statistics with the history. This history is generated periodically and is used as a benchmark that unidirectional flows have to adhere to.

C. Drop Module

Drop module rate limits the flows that are identified as congestion causing by the Statistics module. The Drop module checks and matches each packet with its set of non-conforming flows. If the packet belongs to one of the flows then the module decides whether to drop it based on a predefined formula for rate limiting.

D. Push-Forward Message Generation Module

This module is responsible for generating the Push-Forward messages from the Access Routers. Access Routers alert the Intermediate Routers with the information contained in the Push-Forward message. The message contains information about suspect flows, like the destination IP, source IP and the Protocol of the suspected flow. This message is a simple UDP packet addressed to the target machine with the IP Router Alert option set. The Router Alert option forces every downstream router to deep inspect the packet. Upgraded Intermediate Routers take note of the information contained in the packet and other normal Intermediate Routers just forward the packet without any effect. Thus the Push-Forward messages initiate the monitoring and rate limiting of suspected flows at the upgraded Intermediate Routers.

V. IMPLEMENTATION

Our design prototype was implemented on a Linux router and few main implementation aspects are discussed below.

A. Access Router Implementation

The Collection module implemented as a Linux Kernel module, residing in the IP layer, logs various information about every packet that flows through the router. The module collects packets in intervals of 5 seconds and stops every cycle to refresh its Drop table. Thus the amount of traffic monitored is nearly 95%. The Statistics module is a user level process that periodically parses through the log and generates statistics of every flow. The parsing happens in $O(n)$ time, 'n' being the number of packets logged in the previous interval. This statistics is stored in a hash and contain information like the number of requests seen in the previous monitoring interval, the number of replies, the number of packets dropped due to rate limiting, cumulative size of packets, etc. If the flow is bi-directional, the following logarithmic formula is used to decide if the flow is a congestion causing flow.

$$Allow_Requests = Replies * (1 + 1 / \log (Replies))$$

Any flow whose $(Requests + Drops) > Allow_Requests$ is classified for rate limiting. Here Requests, Replies and Drops denote the corresponding number of requests, replies, and drops seen for that flow in the previous monitoring interval.

Information about the identified flows is updated to a Drop table, which is hashed for easy lookup and is used by the Drop module for rate limiting. The flows are identified using the destination IP, source IP and the protocol. The Drop module, which is a LKM, decides to either accept or drop a packet belonging to an identified congestion causing flow, using the following policy:

Packet Count < Min – Forwarded.

Min < Packet Count < Max – Dropped randomly.

Packet Count > Max – Dropped.

where, Packet Count – Current No. of requests in this cycle; Min – No. of replies seen in previous cycle; Max – Allow_requests computed for this cycle.

Push-Forward message generation module sends the information about rate-limited flows to downstream routers using a simple UDP packet, whose Router Alert (RA) option is set and is addressed to the target machine. The reserved field of the RA option contains the protocol information of the suspected flow.

B. Intermediate Router Implementation

As per our design, under normal circumstances, the Intermediate Router does not do any monitoring. This reduces the overhead of monitoring normal traffic at the router. The Intermediate Router starts its own monitoring and rate limiting activities only after it receives a Push-Forward message from one of the Access Routers upstream. For this purpose, the Intermediate Router maintains a Monitor table that contains information of flows that have to be monitored, which is updated by the Push-Forward messages. Only flows that match one of the entries in the Monitor table are monitored. Then the Intermediate Router works autonomously similar to that of an Access Router but do not generate Push-Forward messages.

VI. RESULTS

Our prototype was tested in Intel Pentium 4 desktops running Linux kernel 2.4.20 at 2GHz. We use five machines, one as the Access Router, one as the Intermediate Router and the other three as host machines. We have four functional tests, three to test the functionalities of the Access Router and one to test the Intermediate Router and the Push-Forward mechanism.

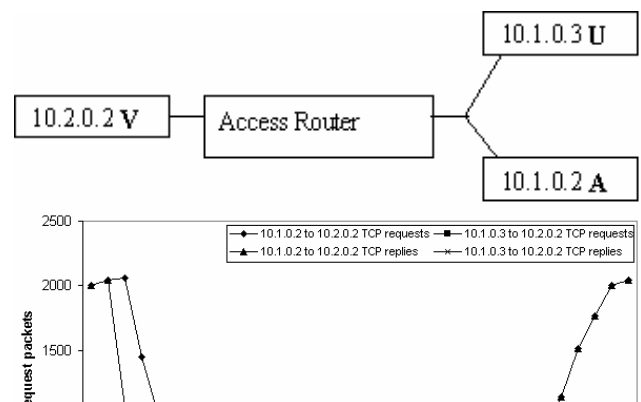


Fig. 2. Target Unavailability Test.

A. Target Unavailability Test

This test establishes the ability of an Access Router to identify victims whose services are down. Though the connection to the victim is available, the victim does not respond to the traffic from the source machines. This test also shows that the Access Router rate limits only the deviating flows. Fig. 2. shows that the flows from machine A to victim V is rate limited once the Access Router identifies a problem at V at interval 4. Thus we see a drop in traffic to a predefined threshold. The other traffic from A to another normal destination D remains unaffected. We also see that once V comes out of the congestion, the traffic is brought back to the normal level in a controlled fashion.

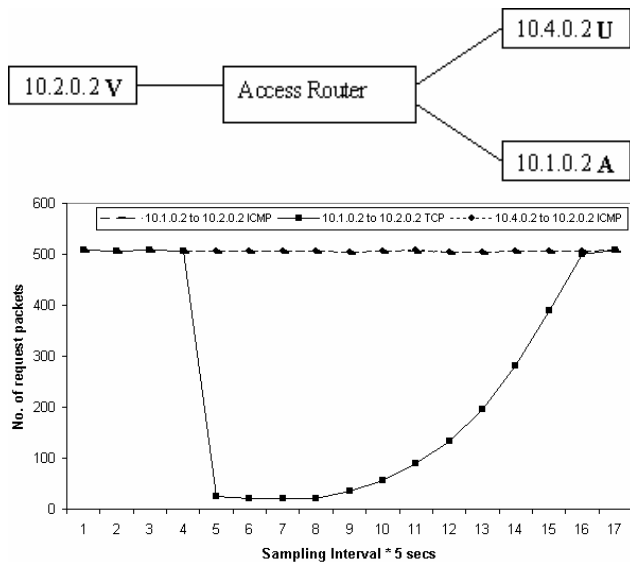


Fig. 3. Service Unavailability Test.

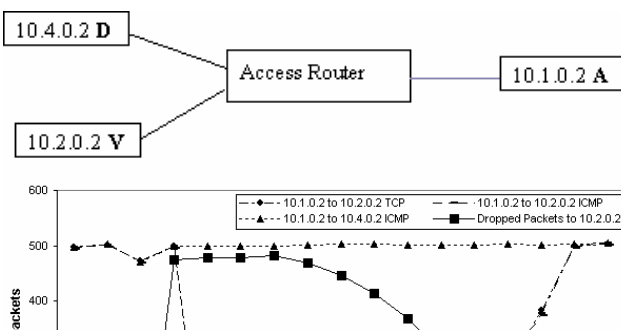


Fig. 4. Fair-share Rate Limit Test.

B. Service Unavailability Test

Here we test the ability of the Access Router to identify a specific service that is unavailable at the victim. Thus the Access Router rate limits only the flow whose service is unavailable and does not affect a flow using another service at the same machine. Fig. 3. shows that there are two flows from A to victim V and one flow from another user U to V. Since only services using TCP goes unavailable at V, only the TCP traffic is rate limited by the Access Router.

C. Fair-share Rate Limit Test

This test is used to establish the ability of the Access Router to identify flows that carry heavy traffic and rate limit flows proportionally, when there are multiple similar flows to the victim. The chart in Fig. 4 shows two traffic flows. Both are from A to a victim V. But one flow produces four times the traffic of another flow. Once the Access Router detects a problem at V, rate limiting begins and the heavy traffic flow is rate limited more than the normal flow, hence exhibiting a feedback property between the victim and the Access Router.

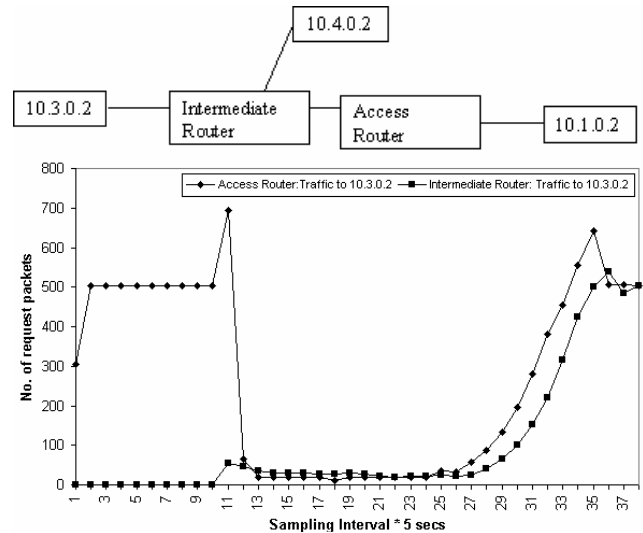


Fig. 5. Intermediate Router and Push-Forward Test.

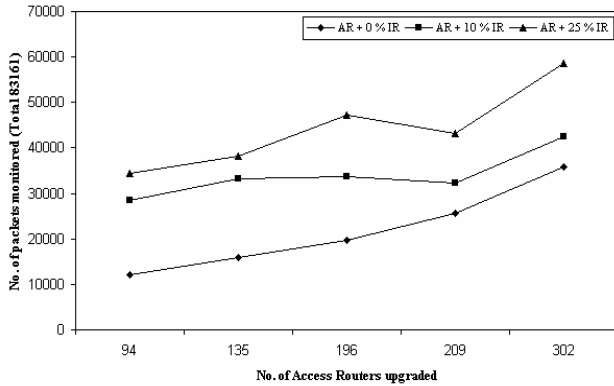


Fig 6. Simulation.

D. Intermediate Router and Push-Forward Test

This test proves that a suspect flow not caught at its own non-upgraded Access Router, is identified and rate limited at the Intermediate Router, because some other upgraded Access Router has alerted the Intermediate Router with a Push-Forward message. The chart in Fig 5 shows the traffic being monitored and rate limited at the Access and Intermediate Routers. Once the Access Router identified a deviating traffic, it rate limits such a traffic and alerts the Intermediate Router. Now the Intermediate Router begins its own monitoring and identifies similar deviating flows flowing through it and rate limits those flows. The monitoring, rate limiting and recovery at the Access Routers and Intermediate Routers are independent and autonomous.

E. Simulation

We have conducted large-scale tests and cost-benefit analysis using Network Simulator 2. We construct an Internet like network with the basic characteristics of the Internet using Inet software [19]. In one of our tests, we have an 800-node network, 695 of them Access Routers and 105 of them Intermediate Routers. We randomly select 100 Access Routers as having attackers under them. The attack machines generate variable number of packets to a selected target machine, maximum being 2000 packets per second. As a result we monitored 83,161 packets per second at the selected target machine. The chart in Fig. 6 shows the amount of traffic monitored by the Access and Intermediate Routers for varying degrees of deployment of our solution. The percentage of traffic monitored gives us an idea of the

effectiveness of this solution to analyze and rate-limit those traffic. The simulation chart shows the amount of traffic covered in the case of a pure Source-End solution (the bottom line in the chart) and a combined deployment. This shows that our solution of using Intermediate Routers along with a Source-End solution provides good traffic coverage.

VII. DISCUSSION

To keep our design lightweight, we have made some design decisions that introduce some problems, but which are relatively easy to handle. Some of the problems that we are aware of are Push-Forward message spoofing, granularity of flow monitoring and the history mechanism. The Push-Forward messages can be authenticated using certificates between routers. Our design is flexible enough to handle different granularities of flow monitoring; we can monitor at the connection port level. The history mechanism has to be updated regularly to reflect current traffic conditions. Attackers can misuse this update stage. This misuse can be avoided by sampling the traffic pattern at random intervals. We also assume that the access networks have already deployed defenses against IP address spoofing.

VIII. CONCLUSION

We have presented a viable lightweight router based DDoS detection and recovery mechanism. This solution can be incrementally deployed at the Access and Intermediate Routers. The Push-Forward mechanism in a novel way of reducing the overhead incurred at the Intermediate Router. Simulations show that this solution is better than pure Source-end solutions.

ACKNOWLEDGMENT

We sincerely thank Dr. Amiya Bhattacharya, Assistant Professor in the Department of Computer Science at New Mexico State University, for his valuable inputs.

This work was supported in part by DARPA, under agreement numbers F30602-99-1-0517 and N66001-00-1-8920.

REFERENCES

- [1] CERT Coordination Center. Denial of service attacks. http://www.cert.org/tech_tips/denial_of_service.html.
- [2] Mirkovic, J., J. Martin, and P. Reiher. A taxonomy of DDoS attacks and DDoS defense mechanisms. University of California at Los Angeles. http://lever.cs.ucla.edu/ddos/uc_la_tech_report_020018.pdf.
- [3] Moore, D., G. M. Voelker, and S. Savage. 2001. Inferring internet denial-of-service activity. Proceedings of 10th USENIX Security Symposium.
- [4] Lyu, M., L. Lau. 2000. Firewall security: Policies, testing and performance evaluation. COMPSAC 2000.

- [5] Ferguson, P. 2000. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827. <http://sunsite.cnlab-switch.ch/ftp/doc/standard/rfc/28xx/2827>.
- [6] Sans Institute. Egress filtering v 0.2. <http://www.sans.org/y2k/egress.htm>.
- [7] Park, K., and H. Lee. 2001. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. Proceedings of ACM SIGCOMM 2001.
- [8] Wang, H., D. Zhang, and K. G. Shin. 2002. Detecting SYN flooding attacks. Proceedings of INFOCOM.
- [9] Ramanathan, A. 2002. WADeS: A tool for distributed denial of service attack detection. TAMU-ECE-2002-02, Master of Science Thesis.
- [10] Bellovin, S.M. 2000. ICMP traceback messages. Internet Draft: Draft-bellovin-itrace-00.txt.
- [11] Kent, and W. T. Strayer. 2001. Hash-based IP traceback. Proceedings of ACM SIGCOMM 2001.
- [12] Song, D. X., and A. Perring. 2001. Advanced and authenticated marking schemes for IP traceback. Proceedings of the 2001 IEEE Infocom Conference.
- [13] Dean, D., M. Franklin, and A. Stubblefield. 2001. An algebraic approach to IP traceback. Proceedings of NDSS '01.
- [14] Stone, R. 2000. CenterTrack: An IP overlay network for tracking DoS floods. 9th USENIX Security Symposium.
- [15] Ioannidis, J., and S. M. Bellovin. 2002. Implementing pushback: Router-based defense against DDoS attacks. Proceedings of NDSS'02.
- [16] Mirkovic, J., G. Prier, and P. Reiher. 2003. Source-End DDoS defense. Second IEEE International Symposium on Network Computing and Applications.
- [17] Gil, T.M., and M. Poletto. 2001. MULTOPS: a data-structure for bandwidth attack detection. 10th USENIX Security Symposium.
- [18] Paxson, V. 1997. End-to-End routing behavior in the internet. IEEE/ACM Transactions on Networking 5, vol. 5 (5).
- [19] Inet Topology Generator. <http://topology.eecs.umich.edu/inet/>.