Random Flow Network Modeling and Simulations for DDoS Attack Mitigation*

Jiejun Kong, Mansoor Mirza, James Shu, Christian Yoedhana, Mario Gerla, Songwu Lu
Computer Science Department, University of California, Los Angeles, CA 90095
{jkong,mansoor,shuj,yoedhana,gerla,slu}@cs.ucla.edu

Abstract—**Recent events show that distributed denial-of-service (DDoS) attack imposes great threat to availability of Internet services. In this paper, we study and evaluate DDoS attacks in a random flow network model, a novel and general approach to DDoS attack prevention and tolerance. The model can be used to evaluate the effectiveness of a DDoS countermeasure framework. Following the random flow network model and state-of-art Internet topology and traffic models, our simulation reveals the general relationship among several metrics derived from the model. Based on the simulation results, we suggest to build a more complete and effective DDoS countermeasure framework using complementary solutions to achieve DDoS attack detection, prevention, and tolerance at same time.**

I. Introduction

Denial-of-service (DoS) attack has become a great threat to Internet services. When attack is distributed over the Internet through carefully plotted coordination, the impact of distributed DoS (DDoS) attack can be proportionally severe and less tractable to DoS countermeasures. Though many DDoS countermeasures have been proposed recently, it is not clear that any one of them is able to stop Internet DDoS attacks in the foreseeable future.

When a panacea for an illness is missing, a practical solution could be combining complementary countermeasures as a monolithic defense system that features maximized effectiveness against the threat. Recently some leading designs have followed this direction to counter-attack DDoS sources [10]. Following the same principle, we seek to gain a more complete and comprehensive understanding of DDoS attack before designing a countermeasure framework. This demands general models to analyse DDoS attacks and countermeasures. In this paper a random flow network model is proposed to quantify the effectiveness of DDoS countermeasures. Besides, well-accepted Internet topology and traffic models are employed in our simulation to illustrate the relationships among several simple metrics derived from the flow network model. Based on the simulation evaluation, we propose a two-fold DDoS countermeasure framework to resist Internet DDoS attack with maximized effectiveness. The contribution of this paper is to offer a more general view of anti-DDoS strategies and to correlate the complementary DDoS countermeasures within one framework.

This paper is organized as follows. We begin in § II by describing the state-of-art DDoS countermeasures as well as network topology and traffic modeling of the Internet. § III presents the random flow network model for DDoS attack mitigation. § IV describes our implementation and evaluation of the random flow network model. And § V concludes the paper.

II. Background

A. Denial-of-service attacks and proposed countermeasures

Denial-of-service (DoS) attacks aim at disabling normal functioning of Internet servers by wasting network resources that could have been utilized to provide useful services to legitimate clients. Depending on attackers' strategy, the target resources may be file system space, process space, network bandwidth, or network connections. While system-oriented DoS attacks which exploit vulnerabilities in operating system and protocol stack implementations are more tractable by applying system security patches, congestion-based DoS attacks which exploit weaknesses inside the network design represent a more intricate threat.

Among various network-based DoS attacks, coordinated and distributed denial-of-service (DDoS) attack emerges to be the most intractable one. Though many DDoS countermeasures have been recently proposed in literatures, none of them has been widely deployed to protect the Internet. Among many proposed countermeasures, we classify the leading candidates into following categories.

Source traceback Source traceback does not directly address the DDoS problem. It serves as a reactive detection and deterrent tool against DoS sources. Typical traceback methods include packet-based marking, link testing, and verifiable logging.

Packet-based marking is normally comprised of two complementary components: a marking procedure executed by routers in the network and a path reconstruction procedure implemented by the victim. The routers augment IP packets with address marks en-route, then the victims can use information embedded in the IP packets to trace the attack back to the actual source. Compared to naive chaining of routers' marks, Probabilistic Packet Marking (PPM) [12] is an efficient traceback approach that features controlled IP packet overhead and minimal router support. However, attackers can exploit marking field spoofing to significantly increase the complexity of traceback [8]. It is non-trivial to minimize the delay and overhead in tracing back each of such "smart" attackers. During the immoderate delay, spoofed packets are allowed to exert their influence on server resources before being reactively controlled.

Instead of packet marking, an alternative method is to generate traceback information using separated IP control information such as link testing messages and verifiable logging messages. SPIE [13] is a processing intensive countermeasure that generates and stores hashed packet checksum on affordable routers. Hops on the attacking path can be discovered when the victim

* This work is funded by iMASH project, National Science Foundation award ANI-9986679.

0-7803-7802-4/03/$17.00 © 2003 IEEE

presents a verifiable sample packet to the upstream routers. The traceback mechanism incurs overheads in the form of control message processing, storage, and communication.

Filtering "malicious" packet flows In some cases the "malicious" packet flows can be identified on clearly-defined metrics, e.g., obviously wrong source address or other obvious errors in packet header. Such packet flows can be proactively filtered at routers.

In ingress filtering [5], routers check a packet for its source IP address, and block packets that come from an address beyond the routers' possible ingress address range. This requires a router to accumulate sufficient knowledge to distinguish between legitimate and illegitimate addresses, thus it is most feasible in customer networks or at the border of Internet Service Providers (ISP) where address ownership is relatively unambiguous.

DPF [10] explores the power-law of Internet topology in source address validation. It can be distributed in Internet core routers to proactively stop packet flows with obviously wrong source address, and meanwhile to reactively trace back the attacking sources. Empirical experiments show that DPF can efficiently identify spoofed IP addresses outside the autonomous system (AS[1]) where the attackers reside. In particular for 1997–1999 Internet AS topology, enabling DPF on 18.9% vertex cover of Internet AS topology can effectively stop 88% traffic with spoofed source address. However, though the vertex cover ratio is relatively stable according to Internet AS topology study, the membership of the vertex cover may vary over time. Besides, computing appropriate filtering tables based on existing inter-domain routing protocols (e.g. BGP) is a non-trivial problem. Inconsistent filter table will lead DPF to filter regular traffic.

In SAVE [9], symmetry between the destination forwarding and source validation is explored to realize a protocol similar to Internet routing, but along the reverse direction for maintaining an incoming tree of authenticated sources. The protocol enables SAVE routers to filter "malicious" packet flows.

Packet flow filtering is an effective means to counterattack DDoS flows. However, since filtering is rendered per-flow, routers must possess sufficient power to process a large number of flows simultaneously. Though every operation used by filtering could be efficient, due to the empirical experience obtained from per-flow based Internet IntServ QoS design, such design raises the scalability concern in Internet core routers which could already be heavily loaded.

Rate control Unfortunately, not all "malicious" behaviors are identifiable. In many cases, there is no clear boundary between DDoS attack and insufficient service availability. Countermeasures based on rate control seek to enforce fairness in bandwidth allocation, thus minimize the damages caused by DDoS attacks.

In Pushback [6], rate control is defined on the granularity of *aggregate*, which is a subset of packet flows with identifiable specific property. Pushback routers can effectively regulate any network aggregate if the corresponding property is associated with an attack. Intuitively, the attacking aggregates can be identified by the victim and then pushed back to upstream pushback routers. Router throttle [15] employs a similar strategy, but defines more specific algorithms to ensure fairness among packet flows.

Unlike filtering, rate control is more friendly to regular traffic and incurs less management cost by reusing existing QoS mechanisms. However, its effectiveness depends on the aggressiveness of attacking flows and is thus reduced when the number of attacking flows is large.

In summary, each DDoS countermeasure has its strength and weakness. It is not clear that any of the countermeasures will be the "silver bullet" that can stop DDoS attacks immediately and efficiently. When an obvious answer is unavailable, an acceptable one could be combining the strength of all effective solutions and let them compensate each other's weakness. This requires a general model to illustrate the shared features of all countermeasures. Following this demand, we will demonstrate our efforts in § III.

B. Internet modeling

Internet topology approximation Despite the apparent randomness of the Internet, there are some surprisingly simple power laws of the Internet topology recently discovered by researchers [4]. The empirically derived power laws suggest that the random and hierarchical graph models that have been used to generate Internet-like topologies may not comprehend the inherent structure of the Internet topology at AS level. Barab*á*si and Albert [2] use dynamic graph models to show that power law graphs can be constructed by applying incremental growth on a random graph and enforcing the new nodes to follow a preference to connect to existing ones that are already well connected. They state that this dynamic graph model is applicable to the Internet AS topology modeling, and hence explains why Internet AS topology exhibits the power-law constraints.

Based on large amount of empirical AS data collected from the Internet, literatures [3][7] have revisited these theoretic analysis and further implement several Internet topology generators. Extensive data analysis revealed several inconsistencies between the real Internet topology and results obtained in previous research. Though using a simple model to approximate Internet topology is still an open question, research has offered us several topology generators that match very well with the real Internet (e.g., Inet, Brite, etc.).

Internet traffic approximation Poisson process has been used to model telecommunication traffics for many years. At the beginning of Internet era, people assumed that data traffic generated by individual network nodes is Poisson distributed and so does the aggregated traffic. However, a number of studies have shown that the data traffic in both local-area and wide-area IP networks clearly differs from Poisson process [11]. Empirical measurements suggest that IP data traffic is not memoryless. Instead, presence of self-similarity is widely identified in data traffic of many types of Internet applications. The real IP

[1] AS is defined by routing. Routers inside an AS run an interior routing protocol. The Border Gateway Protocol (BGP) is the *de facto* standard inter-AS routing protocol of today's Internet.

traffic is better modeled by Pareto model with density function $P(X) = \frac{\alpha b^{\alpha}}{x^{\alpha+1}}, x \geq b$, which has a heavy tail that is not shown in Poisson process.

Based on these works, we can create an Internet-like random network following the topological and flow generation properties that have been widely reckoned so far.

III. Problem Analysis

A. Convert DDoS attack to flow network analysis

As a branch of graph theory, flow network theory studies material flows in a system which is modeled as directed graph [1]. Each directed edge in a flow network can be thought of as a conduit for the material. Each conduit has a stated capacity, given as a maximum rate at which the material can flow through the conduit. Vertices are conduit junctions that follow "flow conservation". That is, other than the *source* and the *sink*, the rate at which material enters a vertex must equal the rate at which it leaves the vertex.

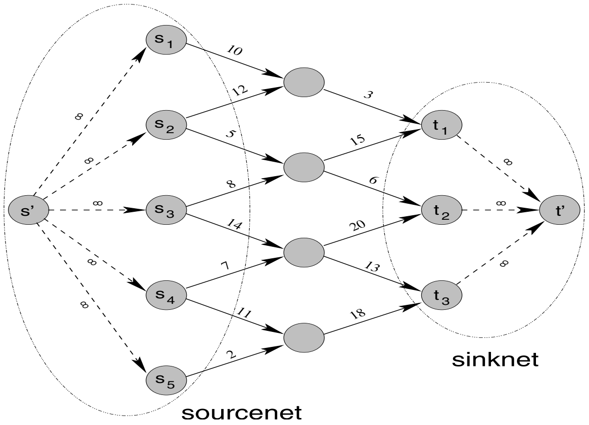

Fig. 1. A single-source single-sink representation of any flow network

Flow network features simple representation. Any multiple-source multiple-sink flow network can be converted to a single-source single-sink flow network using the method depicted in Figure 1. We can add a supersource s' and an edge with infinity capacity from s' to each of the multiple sources, meanwhile add a supersink t' and an edge with infinite capacity from each of the multiple sinks to t'. Thus flow network analysis can be applied assuming the single source s' and the single sink t'. For the ease of representation, we name the subgraph formed by the multiple sinks and sources as *sinknet* and *sourcenet*, respectively.

Some combinatorial problems in flow network analysis can easily be cast as maximum-flow problems. The question raised by maximum-flow problem is: What is the greatest rate at which material can be shipped from the source s' to the sink t' without violating any capacity constraints? It is solved by efficient algorithms like Ford-Fulkerson method and preflow-push method.

DDoS attack can also be reduced to maximum-flow problem. In DDoS scenarios, the distributed attackers are represented by multiple sources greedily pumping flows towards the sinks representing the victims. All other nodes are intermediate nodes forwarding data packets between the attackers and victims. This simple framework explains general intuitions on DDoS attacks. The max-flow min-cut theorem, one of the most important theorem in flow network theory, justifies the success of DDoS attackers' strategy. The theorem states that the maximum achievable flow value from the source s' to the sink t' is equal to the capacity of certain cut in the flow network. Therefore, by adopting a strategy similar to Ford-Fulkerson method, the DDoS attackers are able to fill up the capacity of a minimum cut between the source and the sink, thus successfully partition the network.

B. Problem modeling

To mitigate DDoS attacks with minimum cost, we are interested in the following general problem that has not been studied in literatures:

> Given a flow network $G = (V, E)$, a non-negative **source cost function** $f : V \rightarrow \mathbb{Z}^+$, a non-negative **sink cost function** $g : V \rightarrow \mathbb{Z}^+$, and a non-negative **partition penalty function** $h : V \times V \rightarrow \mathbb{Z}^+$, if the current set of sources and sinks is S and T, respectively, then there exists a min-cut $(X, \overline{X})$ between S and T.

DDoS Countermeasure Problem (DDC): How can a DDoS countermeasure framework decrease S to a subset $S' \subseteq S$ and increase T to a superset $T' \supseteq T$, so that the following cost C is minimized under the new min-cut $(X', \overline{X'})$?

$$\begin{aligned} C &= \sum_{i \in (S-S')} f(i) + \sum_{i \in (T'-T)} g(i) \\ &\quad + \left(\sum_{j \in X', k \in \overline{X'}} h(j,k) - \sum_{j \in X, k \in \overline{X}} h(j,k) \right) \end{aligned}$$

Intuitively, the cost functions model the damages caused by DDoS attacks as well as the investments and overheads incurred by realizing DDoS countermeasures. $\sum_{i \in S} f(i)$ quantifies the complexity in eliminating the attack sources, $\sum_{i \in T} g(i)$ quantifies the complexity in establishing the victim sinks, and the sum of penalty cost $\sum_{j \in X, k \in \overline{X}} h(j,k)$ quantifies the communication loss or service loss between any two nodes in different partitions. Effectiveness of a DDoS countermeasure framework is measured by the value of overall cost C, which must be less than 0 for a qualified solution. For example, in an anti-DDoS solution relying only on ingress filtering, the sink cost measured by g is 0. If a limited number of edge routers could not significantly change the penalty cost measured by h, then the overhead cost measured by f may increase the overall cost C to be positive[2]. Hence the validity of the DDoS countermeasure is challenged. Despite the simplicity in DDC's expression, it is hard to obtain a theoretic answer to this integer programming problem due to many variates in the system.

C. Necessity of network simulation

In many cases, network simulation helps people to understand the behavior of a complex system based on a set of implementable metrics. The state-of-art progress made in Internet modeling offers us an opportunity to study the answer of

[2] In the real world, all arguments are quantifiable and recordable by the routers and victims at network-wide scale. Thus the overall cost C is in general computable "post-mortem".

DDC problem in some special random flow networks. In particular, (i) we are more interested in Internet-like random flow networks. The topological settings in our target random flow network should follow the newly discovered power laws and other associated topological constraints; (ii) We are more interested in flows following Internet traffic models. For the current Internet, the Pareto distribution model is an acceptable approximation for data traffic, and session arrivals of some applications are well-modeled as Poisson process. (iii) We are more interested in the partition penalty cost function h because it is determined by the network settings. On the contrary, f and g are likely determined by out-of-band means.

IV. Implementation and evaluation

We use ns2 simulator (www.isi.edu/nsnam/ns) to study the DDC problem in random flow networks following Internet-like topology and traffic model. Since function f and g are non-negative, source/sink cost monotonously increases with source/sink size. We measure how the source size and sink size affect network partition penalty cost. When flows originated from DDoS attackers and regular users are mixed together, the penalty cost function h is approximated by throughput deterioration, which is the congestion caused by the attack flows at the min-cut. In all simulations, we focus on how the application goodput (application data delivered) is affected by variations in source size and sink size.

A. Simulation implementation

We use Web service provisioning as the example application in our implementation. A network of routers is randomly generated and client hosts are randomly connected to routers. The sinknet is established as a random subset of routers running ns2 HTTP server. Each client host runs an ns2 HTTP client which randomly requests service from a nearby HTTP server.

In each simulation scenario, we pre-define n HTTP clients in the flow network. (i) The number of regular clients is defined by a percentage of the n HTTP clients. Any valid percentage value is supported in our implementation, and we choose 60% in all simulation scenarios. (ii) The remaining HTTP clients are potential DDoS attackers. We let them gradually join the network and finally there are $40\%{\cdot}n$ HTTP clients working as DDoS attackers in the network. During the entire simulation span, we measure HTTP goodput for the $60\%{\cdot}n$ regular clients.

To curtail the complexity to a tractable scale, we simplify the simulation scenario in terms of quantity of network nodes and application clients. Though the number of routers and client hosts on the Internet is potentially very large, the ns2 simulator is not scalable to a comparable size on the hardware we have. Therefore we implement a fully-adjustable ns2 program that can simulate a random flow network with user-configurable settings. The number of network nodes, client hosts, HTTP servers and HTTP clients can be selected as any number from 1 to maximum integral value supported by ns2.

Simulating Internet topology We use the Inet software package (topology.eecs.umich.edu) to generate Internet-like random flow networks. Given the percentage of degree-1 nodes and a random seed, the Inet package can generate a random graph conforming to power laws observed in Internet AS topology. The number of nodes in the result graph is adjustable to be any positive integer.

Simulating Internet data traffic Research has discovered that Pareto model with shape parameter $\alpha{\in}[1..2]$ is a relatively accurate representation of wide-area Internet traffic. In our ns2 implementation, regular HTTP clients follow Pareto model with shape parameter 1.4 in HTTP data transmission. On the contrary, the DDoS attackers disregard any congestion control scheme and pump as much traffic as possible to overwhelm the HTTP servers. This behavior is implemented by a traffic model with uniform distribution at the rate of available bandwidth.

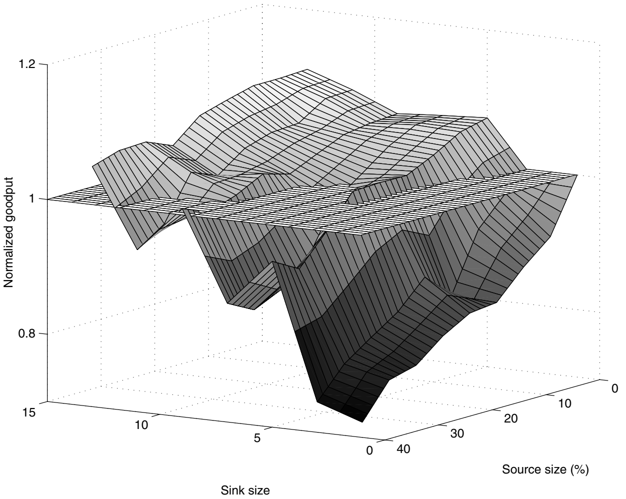

Fig. 2. Goodput vs. source/sink size in an Internet-like random flow network

B. Evaluation

We have run simulations on various network scales. The result shown in Figure 2 is obtained from a network with 20 routers and $n = 100$ HTTP clients/hosts. Among the clients $60\%{\cdot}n = 60$ are regular users, and the remaining are potential DDoS attackers. When number of attacker is 0, we treat the related penalty cost as 0 and measure the corresponding goodput λ. Other HTTP goodput values are normalized on λ. Figure 2 shows the zero-penalty-cost surface $\lambda = 1$ and how the size of sourcenet and sinknet affect the overall HTTP goodput measured for the $60\%{\cdot}n$ regular users. The right edge shows that the goodput gradually increases to normal when the number of DDoS attackers decreases from $40\%{\cdot}n$ towards 0. The left edge shows that the goodput increases to normal when the victim sink size increases. In particular, the goodput increment is faster when sink size is small, and slower when sink size is large. The middle part of the diagram is obtained from simulations varying on both attacking sources and victim sinks.

We have run simulations on multiple network scales smaller than the one described above. However, all the results show the same general structure which demonstrates how changes in sourcenet and sinknet affect the quality of application services in an Internet-like power-law random flow network.

The simulation results offer a general view of the DDC problem. Increasing goodput towards the base λ corresponds to decreasing penalty cost h towards 0. To achieve this goal, either the attack sources must be counter-attacked with cost measured by f, or the victim sinks must be expanded with cost measured by g. *While the interpretation justifies the anti-source strategy used in DDoS detection and prevention schemes, it also reveals the potentiality to build tolerance-based pro-sink DDoS countermeasures*: If either anti-source or pro-sink schemes, but not both, is used, then there are less chances to find the minimal-cost solution to DDC problem on the either of two edges instead of the entire surface.

In summary, the simulation results demonstrate the effectiveness of both anti-source schemes (e.g., source traceback, filtering packet flows with wrong source address) and pro-sink schemes. Examples of the latter case include server replications/proxies, content delivery networks, and pushback routers when network congestion between the routers and the victim is negligible.

C. Applicability to Internet-scale random flow networks

Due to the immense size and complexity of the Internet, simulating DDoS attacks in an Internet-scale random flow network is nearly infeasible. It is possible that some properties hold for small-scale random flow networks may be invalidated by the increasing system scale. Here we discuss the possibilities of applying our simulation results to the Internet in the real world.

Tangmunarunkit et al. [14] observed correlations between Internet AS size and Internet AS topology (in particular the AS out-degree). The tentative correlations imply the necessity of building intra-AS topological model to achieve better approximation of the Internet topology at router level. This feature is not considered in the Inet software used by our simulation. Missing the intra-AS model decreases the applicability of our results to the Internet. To mend this drawback, a simple intra-AS model is needed to achieve better approximation of Internet topology while keeping simulation complexity at current level.

Despite the non-trivial difference in scale and topological complexity between the Internet and the small-scale networks simulated, the authors believe the randomness-based design is critical in obtaining meaningful results that characterize the dynamic equilibrium of the system. In all simulation scenarios, network topology is randomly generated according to the state-of-art Internet topology research. Selection of flow sources and sinks, as well as the associated traffic models, are also based on random models capturing the general structure of Internet traffic. If the system behavior is specific to the small-scale flow networks we have simulated, our results will obviously be valid only in these limited contexts. However, in absence of substantial changes in the underlying random network structure, it is reasonable to expect that the general remarks will continue to hold as the network scale increases.

V. Conclusion and future work

DDoS attack has become an intractable threat to many Internet services. As none of the current anti-DoS scheme is expected to be the "silver bullet" that can stop this threat immediately and efficiently, we believe a feasible answer is combining the strength of effective schemes to lessen the intractability of the pressing threat. In this paper, we propose a general flow network model and study complementary means to resist DDoS attack. We employ extensive network simulations to illustrate the general relationship among several metrics derived from the flow network model. The simulations confirm the validity of intrusion detection and prevention based schemes. Meanwhile they also demonstrate the effectiveness of intrusion tolerance based countermeasures. We propose a two-fold Internet DDoS countermeasure framework that features maximized effectiveness rooted from complementary schemes.

Intrusion tolerance based anti-DDoS scheme is a novel approach that has been explored only recently. It has a lot of potential for future research that includes (i) a more complete flow network analysis of Internet traffic engineering, (ii) analysis to find the best places to deploy DDoS-tolerant proxies and pushback routers, (iii) an intra-AS topology model to approximate Internet topology more precisely at router level, and (iv) obtaining more hardware support to run simulations at larger scales. Before these open issues are explored, the applicability of our design to Internet-scale random flow networks is still an open question. Nevertheless, in absence of substantial changes in the underlying random network structure, it is reasonable to expect that the general remarks obtained from our experiments will continue to hold as the network scale increases.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] A. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, pages 509–512, Oct. 1999.

[3] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. The Origin of Power Laws in Internet Topologies Revisited. In *INFOCOM*, 2002.

[4] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-law Relationships of the Internet Topology. In *SIGCOMM*, pages 251–262, 1999.

[5] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial-of-service Attacks which employ IP Source Address Spoofing. `http://www.ietf.org/rfc/rfc2827.txt`, 2000.

[6] J. Ioannidis and S. M. Bellovin. Implementing Pushback: Router-Based Defense Against DDoS Attacks. In *ISOC Network and Distributed System Security Symposium (NDSS)*, 2002.

[7] C. Jin, Q. Chen, and S. Jamin. Inet: Internet Topology Generator. Technical Report CSE-TR-443-00, Department of EECS, University of Michigan, 2000.

[8] H. Lee and K. Park. On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack. In *IEEE INFOCOM*, pages 338–347, 2001.

[9] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang. SAVE: Source Address Validity Enforcement Protocol. In *INFOCOM*, 2002.

[10] K. Park and H. Lee. On the Effectiveness of Route-based Packet Filtering for Distributed DoS Attack Prevention in Power-law Internets. In *ACM SIGCOMM*, pages 15–26, 2001.

[11] V. Paxson and S. Floyd. Wide-area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transaction on Networking*, 3(3):226–244, 1995.

[12] S. Savage, D. Wetherall, A. R. Karlin, and T. Anderson. Practical network support for IP traceback. In *ACM SIGCOMM*, pages 295–306, 2000.

[13] A. C. Snoeren. Hash-based IP traceback. In *ACM SIGCOMM*, pages 3–14, 2001.

[14] H. Tangmunarunkit, J. Doyle, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Does Size Determine Degree in AS Topology? *ACM Computer Communication Review*, Oct. 2001.

[15] K. Y. Yau, C. S. Lui, and F. Liang. Defending Against Distributed Denial-of-service Attacks with Max-min Fair Server-centric Router Throttles. In *IEEE International Workshop on Quality of Service (IWQoS)*, 2002.