Empirical Study of Tolerating Denial-of-Service Attacks with a Proxy Network

Ju Wang, Xin Liu and Andrew A. Chien
Department of Computer Science and Engineering and
Center for Networked Systems
University of California, San Diego

jwang@cs.ucsd.edu, xinliu@cs.ucsd.edu, achien@ucsd.edu

Abstract

Proxy networks have been proposed to protect applications from Denial-of-Service (DoS) attacks. However because large-scale study in real networks is infeasible and most previous simulations have failed to capture detailed network behavior, the DoS resilience and performance implications of such use are not well understood in large networks. While post-mortems of actual large-scale attacks are useful, only limited dynamic behavior can be understood from these single instances.

Our work provides the first detailed and broad study of this problem in large-scale realistic networks. The key is that we use an online network simulator to simulate a realistic large-scale network (comparable to several large ISPs). We use a generic proxy network and deploy it in a large simulated network using typical real applications and DoS tools directly. We study detailed system dynamics under various attack scenarios and proxy network configurations.

Specific results are as follows. First, rather than incur a performance penalty, proxy networks can improve users' experienced performance. Second, proxy networks can effectively mitigate the impact of both spread and concentrated large-scale DoS attacks in large networks. Third, proxy networks provide scalable DoS-resilience – resilience can be scaled up to meet the size of the attack, enabling application performance to be protected. Resilience increases almost linearly with the size of a proxy network; that is the attack traffic a given proxy network can resist while preserving a particular level of application performance grows almost linearly with proxy network size. These results provide empirical evidence that proxy networks can be used to tolerate DoS attacks and quantitative guidelines for designing a proxy network to meet a resilience goal.

I. INTRODUCTION

Denial-of-Service (DoS) attacks are a continuing key threat to Internet applications. In such attacks, especially distributed DoS attacks, a set of attackers generates a huge amount of traffic, saturating the victim’s network and causing significant damage. Overlay networks have been proposed to protect applications against such DoS attacks [1-6]. These overlay networks are also known as proxy networks [6, 7]. The key idea is to hide the application behind the proxy network, using the proxy network to mediate all the traffic to the application and thereby preventing direct attacks on the application. Realistic study of these problems should involve large networks, real applications, and real attacks.

To date, studies of these approaches have been limited to theoretical analysis and small-scale experiments [1-6]. Thus, we still do not have answers to many key questions about the viability and properties of these proxy approaches. Specifically, with real complex network structures and protocol behavior, can proxy networks tolerate DoS attacks? If so, what are the key parameters to achieve effective and efficient resilience? If we use proxy networks, what are the performance implications for applications?

Our approach exploits the recent availability of a detailed large-scale online network simulator – MicroGrid [8, 9] – to study proxy networks with real applications and real DoS attacks. MicroGrid supports detailed packet-level simulation of large networks and use of unmodified applications. With MicroGrid, we are able to make detailed performance studies in large networks environment with complex, typical application packages and real attack software. Our studies include networks with up to 10,000 routers and 40 Autonomous Systems (ASes) with a physical extent comparable to the North American continent. We believe this is the first empirical study of proxy networks for DoS resilience at large-scale, using real attacks, and in a realistic environment.

Our experiments explore a range of network sizes, proxy network configurations, attack parameters, and application characteristics. The key results are summarized below:

- Rather than incurring a performance penalty, proxy networks can improve users’ experienced performance, reducing latency and increasing

1 The term overlay network refers to both structured Distributed Hash Tables (DHT) and unstructured overlays.
delivered bandwidth. The intuition that indirection reduces performance turns out to be incorrect, as the improved TCP performance more than compensates.

- Proxy networks can effectively mitigate the impact of both spread and concentrated large-scale DoS attacks in large network environment. Our experiments have shown that a 192-node proxy network with 64 edge proxies (each connected by a 100Mbps uplink), can successfully resist a range of large-scale distributed DoS attacks with up to 6.0Gbps aggregated traffic and different attack load distribution; most users (>90%) do not experience significant performance degradation under these attack scenarios.

- Proxy networks provide scalable DoS-resilience – resilience can be scaled up to meet the size of the attack, enabling application performance to be protected. Resilience increases almost linearly with the size of a proxy network; that is the attack traffic a given proxy network can resist while preserving a particular level of application performance grows almost linearly with proxy network size.

These results provide empirical evidence that proxy networks can be used to tolerate DoS attacks and quantitative guidelines for designing a proxy network to meet a resilience goal.

The remainder of the paper is organized as follows. Section II provides background on the DoS problem and the proxy network approach. Section III defines the problem and describes our approach. Section IV briefly describes the MicroGrid simulation environment which provides critical new capabilities. Experimental results and analysis are presented in Section V. Section VI discusses the implications of our studies and relates our work to previous work. Section VII summarizes the results and discusses directions for future work.

II. BACKGROUND

We briefly describe the applications of concern and the denial-of-service attacks that we study in this paper. Then we describe generic proxy network approach to protect applications from DoS attacks.

A. Internet Applications & Denial-of-Service Attacks

Figure 1 illustrates a typical Internet application deployment, such as an e-Commerce application. The application service runs on a cluster of servers. Users are distributed widely across the Internet, and access the application service via the Internet. As shown in Figure 2, in this application model the Internet is a communication layer used to convey a well-defined application-level protocol between the applications and their users. e-Commerce sites and search engines are good examples of such application services.

![Figure 1 Internet Application and DoS Attacks](image)

Because Internet applications must support access from widely dispersed users, Denial-of-Service (DoS) attacks are a major security threat to Internet applications. In a DoS attack, attackers consume resource which either the applications or accesses to the applications depend on, making the applications unavailable to their users.

![Figure 2 Application Model](image)

There are two classes of DoS attacks: infrastructure-level and application-level attacks. Infrastructure-level attacks directly attack the service infrastructure, such as the networks and the hosts of the application services, for example, by sending floods of network traffic to saturate the victim network. In contrast, application-level attacks exploit weakness in application-level protocols, for example, by overloading application services with abusive workload or by sending malicious requests which cause application services to crash.

Infrastructure-level DoS attacks only require the knowledge of applications’ network address, i.e. IP address. Meanwhile, application-level DoS attacks are tightly-coupled with application-level protocols and do not require applications’ IP addresses.

Distributed Denial-of-Service (DDoS) attacks are large-scale DoS attacks which employ a large number of attackers distributed across the network. There are two stages in such attacks. First, attackers build large zombie networks by compromising many Internet hosts and installing a zombie program on each. Second, attackers activate this large zombie network, directing them to “DoS” a victim. Both infrastructure and
application-level DoS attacks can be used in stage two. Automated DDoS toolkits, such as Trinoo, TFN2k and mstream [10-12], and worms, such as CodeRed [13, 14] provide automation, enabling large scale attacks to be easily constructed.

This paper focuses on infrastructure-level distributed DoS attacks.

B. Proxy Network Approach

Proxy networks have been proposed as a means to protect applications from DoS attacks [1-4]. Figure 3 illustrates a generic proxy network encompassing most of the proposed approaches [1-4]. Nodes of the proxy network run on a large resource pool of Internet hosts; proxies and applications form an overlay network by having logical connections among them. Applications are hidden behind the proxy network; all traffic to and from the applications comes through the proxy network. A small number of proxies, known as edge proxies, publish their IP addresses and users use them to communicate with the applications. Each proxy only knows the IP addresses of its neighbors. Different overlay schemes [1, 3, 5] employ different overlay topologies and routing algorithms, and they have slight differences in how they enforce application access through the proxy network.

In [6, 7], we studied the feasibility of location-hiding – hiding applications’ IP addresses – to prevent attackers from circumventing the proxy network defense. Once all application accesses are enforced through the proxy network, only application level traffic is delivered, thereby preventing direct infrastructure DoS attacks on the applications.

The proxy network is widely distributed and highly redundant, so theory speculates that it can shield applications from infrastructure DoS attacks.

III. PROBLEM DEFINITION AND APPROACH

A. Problem Definition

We have little understanding of the performance or effectiveness of proxy networks to provide DoS resilience in large-scale realistic networks. To date, studies of these problems have been limited to theoretical analysis and small-scale experiments. They cannot capture real complex network structures, real temporal and feedback behavior of network and application protocols, and detailed network dynamics, such as router queuing and individual packet drops. All these have important impact on system performance.

Thus, we still do not have answers to many key questions about the viability and properties of these proxy approaches.

• With real complex network structures and protocol behavior, can proxy networks tolerate DoS attacks? In particular, in large realistic networks, under various attack scenarios, how much can proxy networks mitigate the impact of DoS attacks on users’ experienced performance? What are the key parameters to achieve effective and efficient resilience? How does this capability scale up when proxy networks grow in size?

• What are the basic performance implications of proxy networks? How do they affect users’ experienced performance for real applications in large-scale realistic networks?

B. Approach

Our approach is to use newly available simulation tools to new studies that are significantly more realistic in several key dimensions, including:

- Detailed network dynamics, such as router queuing and individual packet drops.
- Real temporal and feedback behavior of network and application protocols and their interaction with other network traffic.
- Emergent properties of large-scale realistic networks, such as topology, latency and bandwidth distribution.

Because DoS attacks exercise extreme points of network behavior, correct modeling of such detail is important for realistic studies. In this context, we study the performance and DoS resilience of the generic proxy network approach. Details of our approach include:

• use of a large-scale, high-fidelity packet-level online network simulator – MicroGrid (see section IV.B) – to simulate large-scale realistic network environment, which include up to 10,000 routers and 40 ASes comparable to the size of large ISPs.
• a real proxy network implementation and real applications deployed together in the MicroGrid virtual environment.

• a large zombie network of 100 zombies and a real distributed DoS toolkit to generate attack traffic. It supports controlled experiments with various attack scenarios.

• a tree proxy network topology, rooted at the application with edge proxies at the leaves providing user access. The number of edge proxies is the width of the tree, and the number of hops from root to leaves is the height. For a localized application implementation, the tree corresponds to subset of links that would be exercised in all proxy networks.

• systematic study of a range of attacks, proxy network configurations, application, and resilience strategies.

We systematically study users’ experienced performance using a range of proxy network topologies to understand the basic performance impacts of proxy networks; then we generate a range of attack scenarios with different attack magnitude and distribution, and systematically study their impact on users’ experienced performance with proxy networks of different sizes to understand proxy networks’ DoS-resilience capabilities and scalability.

IV. EXPERIMENTAL ENVIRONMENT

We describe the key software components used in the empirical study, MicroGrid simulation environment, and the resources used in the experiments.

A. Software Environment

There are four key software components used in the experiments: a generic proxy network implementation, apache web server [15] as the application, a web testing tool “siege”[16] to simulate user access, and a DDoS attack tool “Trinoo”[10].

1) Proxy Network Implementation

The generic proxy network is composed of proxy nodes. It can be configured to support any topology and extended to support any routing algorithm. Proxies have unique identifiers, and act as routers. Each pair of neighboring proxies maintains a TCP connection. When a proxy starts, it connects to its neighbors according to the given topology information and some bootstrap location information. Messages can be routed inside the proxy network using any routing algorithm.

The proxy network supports all TCP applications transparently. We use the DNS scheme used by content delivery networks [17] to direct user access to proxies.

Figure 4 Proxy Network Prototype

As shown in Figure 4, edge proxies listen to user connection requests, and encode application traffic into messages which are routed via the proxy network to the application. At the exit of the proxy network, application proxies (proxies that directly connect to the application) decode the messages, establish new connections to the application if necessary, and send the data to the application. The TCP connections among proxies are persistent and shared among users.

2) Application Service

We use Apache web server as a representative application front-end. Since we focus on the network impact of DoS attacks, specific details of the application logic at the back-end are not critical. Here we use Apache server to serve files of different sizes as a representative scenario.

3) User Simulator

We use siege – a web test toolkit – to generate user requests. Siege can generate web requests based on a list of URLs and measure the response time for each of the requests. This allows us to simulate user access and collect statistics which characterize user experienced performance.

4) DDoS Attack Toolkit

Trinoo [10] is a DDoS attack toolkit generally available on the Internet. It includes a daemon and a master program. A typical trinoo network consists of a collection of compromised Internet hosts running the trinoo daemon program. The master program is used to control this trinoo network to make DDoS attacks. Given a list of IP addresses, trinoo daemons send UDP packets to the targets at the given start time. In its original form, the trinoo daemon repeatedly sends UDP packets at full speed. To support controlled experiments, we changed trinoo daemon allowing its sending rate to be adjusted.

B. MicroGrid Simulation Toolkit

The MicroGrid [8, 9] is an integrated online packet-level simulator that provides modeling of virtual network environments. Using the MicroGrid, users can...
configure an arbitrary virtual network, deploy it to a cluster, and then execute their unmodified applications directly in that virtual network. Three key capabilities of MicroGrid make it crucial to our study.

- The ability to simulate large networks at high fidelity even at high levels of traffic. MicroGrid has demonstrated good scalability in realistic large-scale simulations of networks with 20,000 routers (comparable to a large Tier-1 ISP network like AT&T) [18].

- Support for realistic topology, routing and a full network protocol stack. MicroGrid is integrated with a topology generator mABrite[19], which can create realistic Internet-like network topologies and set up BGP routing policies automatically based on realistic Internet AS relationships. It supports Internet routing protocols such as BGP [20] and OSPF [21]. It also supports networking protocols, such as IP, UDP, TCP [22] and ICMP [23].

- Support for direct execution of unmodified applications.

These capabilities of MicroGrid allow us to study the properties of the proxy network and detailed behavior of the system in a large-scale network environment with realistic settings, running real applications and real attacks. These capabilities are markedly greater than testbeds such as PlanetLab [24] or small scale simulators such as NS2 [25], where the scale, intensity and range of attack scenarios that can be studied are limited.

C. Simulation Setup

1) Simulated Network

As shown in Figure 5, the proxy network, apache server, siege programs and trinoo attackers are deployed in the MicroGrid simulated network environment. The mABrite topology generator is used to generate Internet-like Power-Law network topologies [19, 26]. We use two virtual networks in our experiments. One includes 1000 routers and 20 ASes, and the other includes 10,000 routers and 40 ASes, which is comparable to the size of a large ISP network. Both networks span a geographic area of 5000 miles by 5000 miles, which is roughly the size of the North American continent. This physical extent determines link latencies. OSPF routing is used inside ASes, and BGP4 is used for inter-AS routing.

2) Physical Resources

Our experiments use two clusters. The MicroGrid simulator runs on a 16-node dual 2.4GHz Xeon Linux cluster with 1GB main memory on each machine, connected by a 1Gbps Ethernet switch. Other software components run on a 24-node dual 450MHz PII Linux cluster with 1GB main memory on each machine, connected by a 100Mbps Ethernet switch. These two clusters are connected with a 1Gbps link.

V. EXPERIMENTS AND RESULTS

We study the performance implications and DoS resilience of proxy networks, and address the problems stated in Section III.

A. Proxy Network Performance

To understand the basic performance implications of the proxy network approach, we compare the user-observed service performance for direct application
access and via proxy network. Users choose edge proxies based on proximity.

The proxy network is deployed as follows. Edge proxies are distributed uniformly across the simulated network. Application proxies (see Section IV.A) are placed on hosts physically close to the application. The remaining proxies are distributed evenly between edge proxies and application proxies. This heuristic tries to align a proxy network structure with underlying network to avoid long detours in overlay routes.

Figure 6 shows the results in the simulated network with 20ASes and 1000 routers (described in Section IV.C) for a tree-topology 192-node proxy network. 64 nodes are edge proxies. The X-axis is the response time for a user to download a given size file (1.5KB, 100KB or 1MB). We plot measured performance for direct access and proxy network. The Y-axis is Cumulative Density Function (CDF) of user-observed response time over the user population. Hence a curve close to the Y-axis implies more users experiencing good performance.

While one might expect proxies to degrade performance, the proxy network improves performance. For small requests (e.g. 1.5KB), the 50-percentile response time is reduced by half, and for medium size requests (e.g. 100KB), the improvement is even more significant, and so is the case of large files (e.g. 1MB). There are three main reasons for these phenomena:

1. Proxy network improves connection set up time. As described in Section IV.A (see Figure 7), there are established TCP connections among proxies. For each virtual connection between a user and the application, instead of establishing a long TCP connection from the user to the application, two shorter TCP connections are established: one from the user to the edge proxy and one from the corresponding application proxy to the application. Both of them have small RTT (round trip time), because application proxies are close to the application, and users choose edge proxies based on proximity. Since the TCP handshake [22] takes 1.5 RTT, the connection setup cost can be reduced by one RTT between the user and the application\(^2\). This effect is prominent for small requests (e.g. 1.5KB) as shown in Figure 6.

2. The TCP connections among proxies are persistent, and in most cases the TCP congestion windows for those connections have already been fully opened by previous data transfers and other users' traffic. Thus, they no longer suffer a slow start phase [22] to grow the congestion window. For medium size requests (e.g. 100KB shown in Figure 6), this effect is most prominent.

3. A series of shorter TCP connections can also improve throughput and robustness as studied in Logistic Networking [27]. Here we give a brief explanation, and details can be found in [27]. The throughput can be improved because the TCP throughput is roughly TCP send buffer size divided by RTT, and the connections among proxies have shorter RTTs comparing to the RTT between the user and the application. The throughput effect can be seen for large requests (e.g. 1MB shown in Figure 6).

Figure 7 Direct Access vs. Proxy Network

![Figure 7 Direct Access vs. Proxy Network](image)

1. Proxy network improves connection set up time. As described in Section IV.A (see Figure 7), there are established TCP connections among proxies. For each virtual connection between a user and the application, instead of establishing a long TCP connection from the user to the application, two shorter TCP connections are established: one from the user to the edge proxy and one from the corresponding application proxy to the application. Both of them have small RTT (round trip time), because application proxies are close to the application, and users choose edge proxies based on proximity. Since the TCP handshake [22] takes 1.5 RTT, the connection setup cost can be reduced by one RTT between the user and the application\(^2\). This effect

\(^2\) Only a one-way trip is needed from the edge proxy to the application proxy, instead of a full hand-shake. In fact, once the user gets connected to the edge proxy, it can start sending data. This can be overlapped with the connection setup at the application proxy side.
B. DoS-Resilience of Proxy Networks

To explore the DoS-resilience capability of proxy networks, we study user-experienced performance under a range of attack scenarios with or without proxy networks. We use the same proxy network, which contains 192 proxies (64 edge proxies), in the simulated network with 20 ASes and 1000 routers. In addition, we constructed a DDoS network, which contains 100 Trinoo daemons randomly distributed in the network.

Our first experiment explores whether a proxy network can really protect an application from DoS attacks. Our second experiment studies the DoS-resilience capability of the proxy network under two large-scale attack scenarios: spreading DoS attacks where attack load is distributed evenly on all the edge proxies, and concentrated DoS attacks where attack load is concentrated on a subset of edge proxies to saturate their incoming links. Our final experiment studies the scalability of proxy networks with respect to DoS-resilience, by varying the size and width of proxy networks.

1) Can a proxy network protect applications?

We compare the impact of a DoS attack cast on the application and the proxy network. In our experimental setting, the application service is connected by a 250Mbps link, and each edge proxy is connected by a 100 Mbps link. Figure 9 shows the CDF for user-observed service response time of 100KB request size in scenarios with or without a proxy network. The results show that a 250Mbps attack on the application significantly increases service response time (about 10x) and the application becomes unusable. However, when a proxy network is used, the attack has no observable impact on the user experienced performance. The reason is straightforward. By having a collection of edge proxies to dilute the impact of attack, a proxy network has a greater capacity than the application, thereby not as easily being saturated.

2) How large an attack can a proxy network resist?

To investigate how well a proxy network can tolerate DoS attacks, we launch both spreading and concentrate DoS attacks on the proxy network described in Section V.A, which has 64 edge proxies and 192 proxies in total. Each of the edge proxy has a 100Mbps uplink. In both cases, we vary the aggregated attack magnitude from 3.2Gbps to 6.4Gbps. In this experiment, users do not switch proxies during attacks.

In the case of spreading DoS attacks, Figure 10 shows that when attack magnitude is no more than 6.0Gbps (recall that the aggregated uplink capacity for all the edge proxies is 6.4Gbps), more than 95% of the users observe no significant performance degradation -- the DoS attack has been successfully tolerated. The reason for this is that the edge proxies successfully dilute attack, and even under heavy attack loads, most of the edge proxies still have sufficient capacity left to serve user requests. Figure 10 also shows that when attack load reaches 6.4Gbps, all the edge proxies are saturated, significant performance degradation occurs for all users.

Figure 9 DoS Resilience of Proxy Network

More interesting, we can see large performance degradation for a small fraction of users (<5%) when the attack magnitude is 6.0Gbps. It is due to the correlation among proxies and users (see Figure 11). Two edge proxies A and B share an uplink of OC3 (155Mbps). Before attack traffic saturates both proxies' local links (100Mbps), the shared OC3 link gets congested first. Therefore, users who use these two proxies and users who are in the same network as these proxies will be affected. This effect limits the effectiveness of proxy networks.

Figure 10 Resilience to Spreading DoS Attack

Figure 11 Correlation among Proxies and Users
Figure 12 shows the case of concentrate attacks, where attack load is concentrated on a subset of proxies. In this case, attack traffic saturates part of the proxy network and a significant percentage of users are affected due to congestion and packet loss. This effect is more prominent when attack load is higher than the proxies’ capacity (e.g. 4.0Gbps attack on 32 proxies).

We observe that parts of the proxy network are not under direct DoS attacks, therefore if users can switch to edge proxies not being attacked, the performance can be potentially improved.

We repeat the concentrate DoS attack experiment, and let users switch to the closest proxy not being saturated. Figure 13 shows the CDF of user-observed performance. Compared with Figure 12, the performance has been significantly improved. For comparison, Figure 13 also plots the baseline case where users directly access the application without attack traffic. It shows that even under high attack load (e.g. 6.0Gbps) the proxy network can still maintain slightly better performance than direct application access without attacks for most users. Therefore proxy networks can effectively resist DoS attacks.

To understand the performance gap between the attack cases and the non-attack case, we measure the users’ experienced performance without attack, while using the set of edge proxies they switch to during attacks (shown in Figure 14). For most users, this curve closely follows the attack cases, showing that the performance gap is due to switching edge proxies rather than congestion caused by attack traffic. Additionally, a small number of users are greatly affected by the attack due to the limitation of the underlying network discussed in Figure 11.

Finally, we explore how varying the size (width) of the proxy network affects DoS resilience. This is an important scaling property of the proxy network, showing how effective we can resist larger scale DoS attacks by building larger proxy networks.

The goal of our experiment is to evaluate the amount of attack load proxy networks can withstand for a range of proxy network widths. It is hard to directly measure the maximum attack load a proxy network can tolerate. Instead, we set the attack magnitude to be 95% of the proxy network’s capacity, and measure the user-observed performance. We define the capacity of a proxy network to be the sum of the link capacity of its edge proxies. For example, if the proxy network has 16 edge proxies and each edge proxy has a 100 Mbps uplink, then its capacity is 1.6Gbps and the aggregated attack magnitude is 1.52Gbps.

Proxy network scaling results are shown in Figure 15. The X-axis is the number of edge proxies in the proxy network (they all have height 3), and the Y-axis is the user-experienced service response time for a
certain percentile of users. We can see that for up to 95 percent users, the curves stay horizontal and less than 2 seconds (recall from Figure 6 that the 95 percentile performance for direct application access without attacks is 2 seconds). If we define 95% users not being affected by DoS attacks as successful DoS resilience, then the amount of attack traffic can be tolerated grows linearly with the size of the proxy network.

Summarizing our experiments, we first explored the user-experienced performance using a range of proxy networks in two simulated large realistic network environments. We find that proxy networks can in fact improve the performance for TCP-based applications. Then we conducted a series of experiments investigating the user-experienced performance under different attack scenarios using a range of proxy networks. We find that proxy networks provide good resilience to spreading DoS attacks. And, if proxy switching is allowed, concentrate attacks can be tolerated – most users can retain good performance. In exploring the properties of large proxy networks, we find that by growing the size (width) of the proxy network, the magnitude of DoS attacks it can tolerate grows almost linearly. Therefore, in realistic large network environments, proxy networks can have great performance potential and scalable DoS-resilience capability. It is a promising approach to DoS defense.

VI. RELATED WORK

Our primary focus is the capabilities of proxy networks used for DoS defense. The most related studies are those exploring the use of overlay networks to resist DoS attacks. Secure Overlay Services (SOS) [1] protects applications against flooding DoS attacks by installing filters around applications and only allowing traffic from secret "servlets". SOS uses Chord [28] to implement communication between users and the secret servlets without revealing the IP addresses of the servlets. Mayday [4] generalizes the SOS architecture and analyzes the implications of choosing different filtering techniques and overlay routing mechanisms. Internet Indirection Infrastructure (i3) [3, 5] also uses Chord overlay to protect applications from direct DoS attacks. SOS, Mayday and i3 can all be viewed as specific instances of our generic proxy network. Each of these efforts has involved some evaluation by theoretical analysis or small-scale experiments.

The primary distinctions of our work are:

First, our studies are based on detailed network behavior and explore large scale network structures, attacks, and proxy networks with real application, attack, and protocol software. The primary reason we are able to undertake these studies is the novel capabilities of MicroGrid.

Second, our work differs in focus. Each of these other efforts focuses on their specific proposed solution, exploiting its structure and characteristics for analysis. Therefore the evaluation of one often applies only to that particular instance of proxy networks. In contrast, our work focuses on the fundamental capabilities and limitations of proxy networks in general.

Another class of related research is on the performance and static resilience of overlay networks in general. [29] studied these issues from a graph theoretic perspective, and [30] takes an empirical approach to study the overlay network performance. There are three key differences between our work and their studies. First, our work studies the impact of DoS attacks, which affects network dynamics and the performance of real applications, which is not their focus. Second, our work studies performance of real applications, taking into account dynamic behavior of network protocols such as TCP, while their work only considers RTT. Third, our work focuses on performance between users and the application, while they study performance between any pair of overlay nodes.

A third class of related work is the simulation studies on Internet worms and their impact on BGP [31, 32]. They focus on worm propagation and its impact on the network, particularly on the behavior and vulnerabilities of BGP, which are not our focus. These studies are complementary to ours.

VII. CONCLUSIONS AND FUTURE WORK

To understand the performance implications and DoS-resilience capability of proxy networks in large realistic networks, we use a detailed large-scale online network simulator – MicroGrid [8, 9] – to study proxy networks with real applications and real DoS attacks. With MicroGrid, we are able to make detailed performance studies in large networks environment with complex, typical application packages and real attack software. Our studies include networks with up to 10,000 routers and 40 Autonomous Systems (ASes) with a physical extent comparable to the North American continent. We believe this is the first empirical study of proxy networks for DoS resilience at large-scale, using real attacks, and in a realistic environment.

Our experiments explore a range of network sizes, proxy network configurations, attack parameters, and application characteristics. The key results are summarized below:
• Rather than incurring a performance penalty, proxy networks can improve users’ experienced performance, reducing latency and increasing delivered bandwidth. The intuition that redirection reduces performance turns out to be incorrect, as the improved TCP performance more than compensates.

• Proxy networks can effectively mitigate the impact of both spread and concentrated large-scale DoS attacks in large network environment. Our experiments have shown that a 192-node proxy network with 64 edge proxies (each connected by a 100Mbps uplink), can successfully resist a range of large-scale distributed DoS attacks with up to 6.0Gbps aggregated traffic and different attack load distribution; most users (>90%) do not experience significant performance degradation under these attack scenarios.

• Proxy networks provide scalable DoS-resilience – resilience can be scaled up to meet the size of the attack, enabling application performance to be protected. Resilience increases almost linearly with the size of a proxy network; that is the attack traffic a given proxy network can resist while preserving a particular level of application performance grows almost linearly with proxy network size.

These results provide empirical evidence that proxy networks can be used to tolerate DoS attacks and quantitative guidelines for designing a proxy network to meet a resilience goal.

There are several directions for future work. First, we can study proxy networks with topologies which have multiple paths from each edge proxy to the application, in order to understand the benefit of multi-path on performance and DoS-tolerance. Second, multiple applications can share the same proxy network. We can study the correlated impact of DoS attacks on multiple applications. Third, further study is necessary to understand the impact of proxy deployment and understand the network topology on user-observed service performance.

REFERENCES
17. Akamai, Akamai Technology Overview.
25. The ns Manual (formerly ns Notes and Documentation). K. Fall and K. Varadhan, Editors, UC Berkeley. LBL, USC/ISI, and Xerox PARC.