BENCHMARKS FOR DDOS DEFENSE EVALUATION

Jelena Mirkovic, Erinc Arikan and Songjie Wei
University of Delaware
Newark, DE

Sonia Fahmy
Purdue University
West Lafayette, IN

Roshan Thomas
SPARTA, Inc.
Centreville, VA

Peter Reiher
University of California Los Angeles
Los Angeles, CA

***Abstract*— There is a critical need for a common evaluation methodology for distributed denial-of-service (DDoS) defenses, to enable their independent evaluation and comparison. We describe our work on developing this methodology, which consists of: (i) a benchmark suite defining the elements necessary to recreate DDoS attack scenarios in a testbed setting, (ii) a set of performance metrics that express a defense system's effectiveness, cost, and security, and (iii) a specification of a testing methodology that provides guidelines on using benchmarks and summarizing and interpreting performance measures.**

We identify three basic elements of a test scenario: (i) the attack, (ii) the legitimate traffic, and (iii) the network topology including services and resources. The attack dimension defines the attack type and features, while the legitimate traffic dimension defines the mix of the background traffic that interacts with the attack and may experience a denial-of-service effect. The topology/resource dimension describes the limitations of the victim network that the attack targets or interacts with. It captures the physical topology, and the diversity and locations of important network services. We apply two approaches to develop relevant and comprehensive test scenarios for our benchmark suite: (1) we use a set of automated tools to harvest typical attack, legitimate traffic, and topology samples from the Internet, and (2) we study the effect that select features of the attack, legitimate traffic and topology/resources have on the attack impact and the defense effectiveness, and use this knowledge to automatically generate a comprehensive testing strategy for a given defense.

I. Introduction

Distributed denial-of-service (DDoS) attacks are a serious threat for the Internet's stability and reliability. Attacks are typically launched from multiple coordinated machines, under an attacker's control (therefore the term "distributed"), and deny service to legitimate clients by consuming a critical resource. DDoS attacks usually target a single network or a host, although there have been incidents involving multiple targets [1]. Any critical resource may be exhausted, such as router buffer space, network bandwidth, a server's memory or CPU resources, etc. The resource is exhausted either by an attacker sending excessive traffic (memory, bandwidth) or by sending specifically crafted traffic that requires complicated processing (CPU, memory bus).

DDoS attacks have gained importance in recent years because the attackers are becoming more sophisticated and organized, and because several high-profile attacks targeted prominent Internet sites [2], [1]. Many defenses have been proposed against DDoS, both by the research and commercial communities. Some defenses focus on a specific type of attack, while others claim that they can stop all attacks. In such a diverse market, it is necessary to develop an objective, comprehensive and common evaluation methodology for DDoS defenses. This would facilitate comparison of competing products in a common setting, and an objective assessment of their performance claims, thus propelling the DDoS research towards a better understanding of the DDoS phenomena and a design of higher quality solutions.

In this paper, we describe our ongoing work on the development of a common evaluation methodology for DDoS defenses. This methodology consists of three components: (1) a *benchmark suite*, defining all the elements necessary to recreate a comprehensive set of DDoS attack scenarios in a testbed setting, (2) a set of *performance metrics* that express a defense system's effectiveness, cost and security and (3) a specification of a *testing methodology* that provides guidelines on using benchmarks and summarizing and interpreting performance measures.

Benchmarks are commonly used for testing systems in a controlled environment to predict their behavior in real deployment. The value of benchmarks lies in their ability to *faithfully* recreate: (1) *all* elements that may affect a system's performance, and (2) *typical* values, combinations and behavior of these elements that a system may encounter during its operation. Since DDoS attacks are adversarial, in addition to typical scenarios, we must provide atypical scenarios that challenge defenses in novel ways. This is necessary for a comprehensive benchmark suite that thoroughly evaluates defenses against current and future attacks.

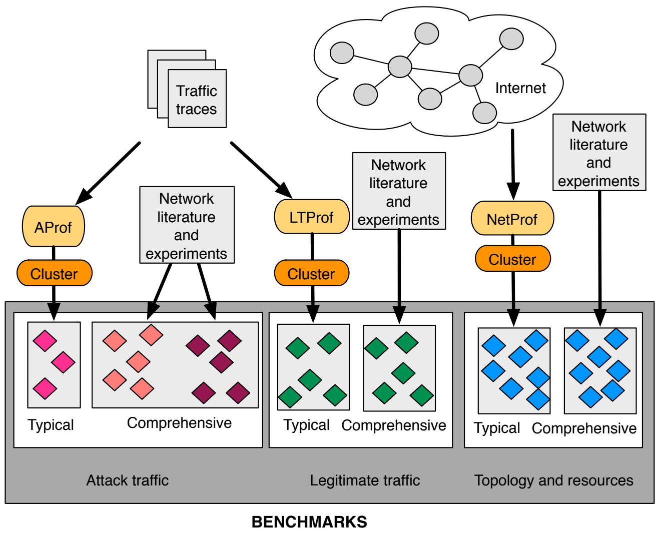


Fig. 1. Benchmark components and their generation

Realistic and comprehensive test scenarios are only one part of the evaluation methodology. Another critically important part is the specification of accurate and expressive performance metrics. These metrics must capture how damaging a given attack was and how well a defense neutralized this damage. In addition to measuring a defense's success in eliminating the DoS effect, the metrics must capture how quickly the defense responds and its deployment and operational cost. Finally, there may be security concerns associated with a defense (e.g., collaborative defenses face certain security risks that stand-alone systems do not). While it is difficult to measure a system's security quantitatively, the evaluation methodology should provide guidelines for describing and comparing risks inherent in a system's design.

A testing methodology specifies how to choose the appropriate attack, legitimate traffic and topology elements for realistic and comprehensive defense testing. It also describes how to aggregate and interpret performance results obtained through a variety of tests.

Our work on the common evaluation methodology for DDoS defenses is in its early stage. While most of our planned tools and activities are ready, data collection and actual definition of test scenarios are just beginning. In this paper, we describe our current progress, present some preliminary results and define future directions for our defense benchmarking project. Section II provides a high-level overview of the benchmark suite, while Sections III, IV and V provide more details on each dimension of the test scenarios. We describe our proposed performance metrics in Section VI, and we provide a brief overview of the testing methodology in Section VII. Section VIII summarizes related work and Section IX provides a conclusion and future directions.

II. DDoS Defense Benchmarks

DDoS defense benchmarks must specify all elements of an attack scenario that influence its impact on a network's infrastructure and a defense's effectiveness. We consider these elements in three dimensions:

- *DDoS attack* — features describing a malicious packet mix arriving at the victim, and the nature, distribution and activities of machines involved in the attack. Attack features naturally determine an attack's impact, as they influence the attack's strength and the resources that are being targeted. The attack's strength and diversity also stress a defense's resource limits (memory, processing), and sophisticated attacks may manage to blend in with the legitimate traffic and avoid detection by some defense systems.
- *Legitimate traffic* — features describing a legitimate packet mix and the communication patterns in the target network. During the attack, legitimate and attack traffic compete for limited resources. The legitimate traffic's features determine how much it will be affected by this competition. For example, TCP connections respond to packet loss by reducing their sending rate, which makes them much less competitive than non-TCP traffic. High-rate TCP connections suffer less from intermittent congestion than low-rate ones, and long connections suffer more than short-lived ones.
- *Network topology and resources* — features describing the target network architecture. These features identify weak spots that may be targeted by a DDoS attack and include network topology and resource distribution. In addition to this, some defenses will perform better or worse, depending on the topology chosen for their evaluation. We need to understand this interaction, to design objective test scenarios. For example, defenses that share resources based on the traffic's path to the victim will perform best with star-like topologies, where attackers and legitimate users are located at different branches. This is because attack and legitimate traffic paths are then clearly distinct, leading to a perfect separation. However, such a scenario is not realistic and does not evaluate collateral damage to users who share a path with an attacker.

The basic benchmark suite will contain a collection of *typical* attack scenarios, specifying typical settings for all three benchmark dimensions. We harvest these settings from the Internet, using automated tools we developed for this project. The *AProf* tool collects attack samples from publicly available traffic traces. It uses a variety of detection criteria to discover attack traffic, and then separates it from the other trace traffic and extracts or infers relevant attack

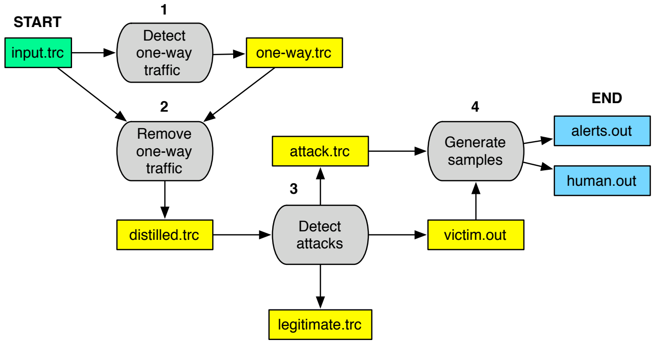


Fig. 2. Attack sample generation with AProf

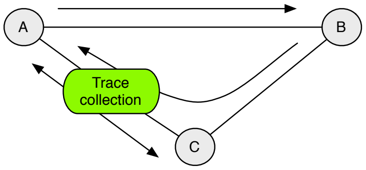


Fig. 3. An example of a traffic trace collection scenario that records one-way traffic between A and B, but two-way traffic between A and C

features. The *LTProf* tool collects legitimate traffic samples from public traces by creating a communication profile for each observed source IP, and grouping profiles by their similarity to expose typical communication patterns. The topology/resource samples are collected by the *NetProf* tool, which harvests router-level topology information from the Internet and uses the nmap tool to detect services within chosen networks. Samples are then clustered by feature similarity into a few distinct categories to be included in the benchmarks. The automated tools are designed to facilitate easy update of the benchmarks in the future as attack, background traffic and network design trends change.

The typical suite provides tests that recreate attack scenarios seen in *today's* networks. A defense can thus be quickly evaluated in a realistic setting but these tests are insufficient for comprehensive evaluation. For instance, they do not answer the question "How would this defense perform in a future network where peer-to-peer traffic was dominant?"

To facilitate in-depth understanding of a defense's capabilities, each dimension will also contain a *comprehensive* suite. Comprehensive tests define a set of traffic and topology features that influence an attack's impact or a defense's performance, and a range in which these features should be varied. Instead of performing exhaustive testing in this multi-dimensional space, our work focuses on understanding the interaction of each select feature with an attack and a defense, so that we can specify several relevant test values and prune all but necessary feature combinations. This is an ambitious goal, but it is necessary for comprehensive defense evaluation and our preliminary research indicates that many features are independent from each other and have only a few relevant test values (e.g., low, medium and high).

Figure 1 illustrates the benchmark components and their generation using automated tools, network literature and experiments in DETER testbed [3] for security experimentation. Once finalized, the benchmarks will be integrated with the DETER traffic and topology generators, to facilitate easy use by researchers, DDoS defense vendors and their customers.

III. Attack Traffic

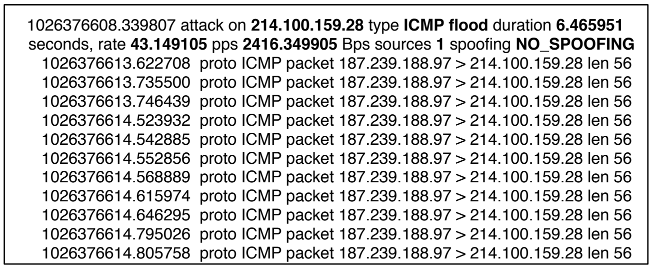
The attack traffic dimension specifies the attack scenarios observed in today's incidents, and hypothetical scenarios designed by security researchers, that may become popular in the future.

A. *Typical attack scenarios*

Typical attack scenarios are obtained by building a set of automated tools that harvest attack information from public traffic traces, stored in libpcap format. These tools form the *AProf* toolkit. They detect attacks in the trace, separate legitimate traffic from attack traffic destined to the attack's target, and create attack samples that describe important attack features such as strength, type, number of sources, etc. Traffic separation does not need to be perfect, as long as the majority of the attack traffic is correctly identified. This ensures that the attack trace will be dominated by attack packets, and that misidentified legitimate traffic will not taint the sampling process. Finally, attack samples are clustered to yield representative attack categories.

Attack sample generation is performed in these four steps, shown in Figure 2:

1) *One-way traffic removal.* One-way traffic is collected if there is an asymmetric route between two hosts and the trace collection occurs only on one part of this route, as illustrated in Figure 3. Because many applications generate two-way traffic (TCP-based applications, ICMP echo traffic, DNS traffic), some of our attack detection tests use the absence of the reverse traffic as an indication that the destination may be overwhelmed by a DDoS attack. One-way traffic, if left in the trace, would naturally trigger a lot of false positives.
   We identify one-way traffic by recognizing one-way TCP traffic, and performing some legitimacy tests on this traffic to ensure that it is not part of the attack. Each TCP connection is recorded and initially flagged as one-way and legitimate. The one-way flag is reset if we observe reverse traffic. The connection



```
1026376608.339807 attack on 214.100.159.28 type ICMP flood duration 6.465951
seconds, rate 43.149105 pps 2416.349905 Bps sources 1 spoofing NO_SPOOFING
   1026376613.622708  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376613.735500  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376613.746439  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376614.523932  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376614.542885  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376614.552856  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376614.568889  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376614.615974  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376614.646295  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376614.795026  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
   1026376614.805758  proto ICMP packet 187.239.188.97 > 214.100.159.28 len 56
```

Fig. 4. A snippet from human.out

is continuously tested for legitimacy by checking if its sequence or acknowledgment numbers increase monotonically. Each failed test adds some amount of suspicion points. Connections that collect a sufficient number of suspicion points have their legitimate flag reset. When the connection is terminated (we see a TCP FIN or RST packet or no packets are exchanged during a specified interval), its IP information is written to the one-way.trc file, if its one-way and legitimate flags were set. In the second pass, we remove from the original trace all packets between pairs identified in one-way.trc, producing the refined trace distilled.trc.

2) *Attack detection* is performed by collecting traffic information from the distilled.trc at two granularities: for each connection (traffic between two IP addresses and two port numbers) and for each destination IP address observed in a trace. Each direction of a connection will generate one connection and one destination record. A packet is identified as malicious or legitimate using the detection criteria associated with: (1) this packet's header, (2) this packet's connection and (3) the features of an attack, which may be detected on the packet's destination. We currently perform the following checks to identify attack traffic:

   - We identify attacks that use aggressive TCP implementations (such as Naptha attack [4]), bypassing the TCP stack, or fabricate junk TCP packets using raw sockets, by checking for a high sent-to-received TCP packet ratio on a destination record, or for mismatched sequence numbers on a TCP connection. If an attack is detected, TCP packets going to the attack's target will all be identified as attack traffic.
   - We identify TCP SYN attacks by checking for a high SYN-to-SYNACK packet ratio on a destination record. All TCP SYN packets going to the target will be flagged as attack traffic.
   - We identify TCP no-flag attacks by checking for the presence of TCP packets with no flags set. Only no-flag TCP packets will be flagged as attack traffic.
   - Some UDP applications require responses from the destination of the UDP traffic (e.g., DNS). The absence of these responses is measured through high sent-to-received UDP packet ratio on a given destination record, and used to identify UDP attacks. In case of one-way UDP traffic (such as media traffic), we will identify an attack if there is no accompanying TCP connection between a given source and destination pair, and there has been a sudden increase in UDP traffic to this destination. In case of an attack, all UDP traffic will be flagged as attack traffic.
   - High-rate ICMP attacks using ICMP echo packets are detected by checking for high echo-to-reply ICMP packet ratio on a destination record. All ICMP packets to this destination will be flagged as attack traffic.
   - We detect known-malicious traffic carrying invalid protocol numbers, same source and destination IP address, or private IP addresses. All packets meeting this detection criteria will be flagged as attack.
   - We check for packet fragmentation rates that are higher than expected for Internet traffic (0.25% [5]), and we identify all fragmented traffic in this case as part of an attack.

   Each packet is classified as legitimate or attack as soon as it is read from the trace, using the attack detection criteria described above. If more than one attack is detected on a given target, we apply precedence rules that give priority to high-confidence alerts (e.g., TCP no-flag attack) over low-confidence alerts (high TCP packet-to-ack ratio). Packets that pass all detection steps without raising an alarm are considered legitimate. We store attack packets in attack.trc and we store legitimate packets in legitimate.trc. When a new attack is detected, attack type and victim IP are written to a file called victim.out.

3) *Attack sample generation.* Attack samples are generated using the attack.trc file by first pairing each attack's trace with the information from victim.out, and then extracting the attack information such as spoofing type, number of sources, attack packet and byte rates, duration and dynamics from the attack trace and compiling them into an attack alert. This step produces two output files: human.out, with the alert and traffic information in a human readable format (a snippet is shown in Figure 4), and alerts.out with the alerts only.

Although it is too early to offer conclusions about typical attack scenarios, our preliminary public trace analysis indicates that an overwhelming majority of attacks are TCP SYN attacks. Each attack machine is participating at a very low rate (2-5 packets per second), presumably to stay under the radar of network monitors deployed at the source, and attacks range in duration from several minutes to several hours.

B. Comprehensive attack scenarios

We are applying three approaches to build comprehensive attack scenarios: (1) We categorize defense approaches and identify attacks that are particularly harmful to certain categories, e.g., an attack that slowly increases its sending rate could trick defenses that build a baseline of normal network traffic over time, (2) We use the network literature and experiments to identify attacks that target critical network services, such as routing or DNS, and invoke overall network collapse, and (3) We investigate the link between the attack features (rate, packet mix, dynamics, etc.) and the attack impact, for a given test setting (network, traffic and defense) to identify relevant features and their test values.

IV. Legitimate Traffic

The legitimate traffic dimension of the benchmarks consists of host models that describe a host's sending behavior. Our final goal is to use these models to drive traffic generation during testing. For the typical suite, we build host models by first creating host profiles from public traffic traces, and then clustering these profiles based on their feature similarity to generate representative models. This process is automated via the *LTProf* tool we developed. For the comprehensive suite, we utilize the networking literature and our own tests to investigate how legitimate traffic features determine an attack's impact and how they interact with various defense systems. In the remainder of this section, we describe in more detail our work on the typical legitimate traffic suite.

We extract features for host profiles from packet header information, which is available in public traffic traces. Each host is identified by its IP address. Selected features include open services on a host, TTL values in a host's packets, and average number of connections and their rate and duration. We also profile several of the most recent TCP and UDP communications and use the Dice similarity of these communications as one of the host's features. This feature reflects the diversity of all the communications initiated by a host. We only build profiles for those hosts that are frequently appearing in the traces, providing sufficient information for profile-building, and hosts that actively initiate communications with other hosts.

After host profiles are built, we cluster them using their feature similarity to derive typical host models. We use agglomerative algorithms for profile clustering: each host is initially placed in a separate cluster, and the algorithm iteratively merges similar clusters until some stop criteria are met. Currently, we use a selected value for minimal intra-cluster distance as a stop criterion. The distance measure is based on the Dice coefficient, with a centroid representing each cluster.

Our preliminary results for legitimate traffic models were obtained by profiling the Auckland-VIII packet trace from NLANR-PMA traffic archive. This trace was captured in December 2003 at the link between the University of Auckland and the rest of the Internet. After filtering out hosts that are not frequent and active, we have 62,187 host profiles left for clustering. Unfortunately, since the data is random-anonymized, we could not identify inside vs. outside hosts. Thus, the resulting models characterize both the incoming and the outgoing traffic of the University of Auckland's network.

We first identify four distinct host categories, based on some observed features: (1) *NAT boxes* have very diverse TTL values that cannot be attributed to routing changes, (2) *scanners* only generate scan traffic, (3) *servers* have some well-known service port open; we differentiate between DNS, SMTP, Web and other servers, and (4) *clients* have no open ports and initiate a consistent volume of daily communications with others. We then apply clustering within each host category. Clustering process generates several compact and large clusters in each category, that contain the majority of hosts. The features of these clusters also indicate meaningful grouping. For example, a large group of SMTP (mail) servers also provides DNS service. In practice, DNS service is necessary for sending and forwarding of e-mail messages, so it makes sense to co-locate it with the SMTP service on the same host. The clustering result confirms this conventional wisdom. Table I shows the number of all and dominant host clusters. We omit the description of each dominant cluster, for brevity.

TABLE I

Legitimate host categories

Host category	Hosts	All clusters	Top clusters
DNS servers	44%	62	Top 6 clusters contain 96% of hosts
SMTP servers	6.4%	65	Top 8 clusters contain 88% of hosts
Web servers	4.4%	85	Top 6 clusters contain 74% of hosts
Other servers	3.2%		
Clients	28%	27	Top 6 clusters contain 90% of hosts
NAT boxes	9%	94	Top 7 clusters contain 67% of hosts
Scanners	5%	9	Top 5 clusters contain 99% of hosts

V. Topology and Resources

We believe that it is imperative to have representative topologies for DDoS defense testing both at the Internet level, to test distributed defenses that span multiple autonomous systems, and at the enterprise level, to test localized defenses that protect a single network.

To reproduce topologies containing multiple autonomous systems (ASes) at the router level, we are developing a tool, *NetTopology*, which is similar to *RocketFuel* [6]. NetTopology relies on invoking traceroute commands from different servers [7], performing alias resolution, and inferring several routing (e.g., Open Shortest Path First (OSPF) routing weights) and geographical (e.g., location) properties.

To generate topologies that can be used on a testbed like DETER, we have developed two additional tool suites: (i) *RocketFuel-to-ns*, which converts topologies generated by the NetTopology tool or RocketFuel to DETER-compliant configuration scripts, and (ii) *RouterConfig*, a tool that takes a topology as input and produces router (software or hardware) BGP and OSPF configuration scripts, according to the router relationships in the specified topology. Configuring routers running the BGP protocol poses a significant challenge, since Internet Service Providers use complex BGP policies for traffic engineering. We utilize the work by Gao et al. [8], [9] to infer AS relationships and use that information to generate configuration files for BGP routers. Jointly, the NetTopology, RocketFuel-to-ns and RouterConfig tools form the *NetProf* toolkit. They enable us to automatically configure the DETER testbed with a set of realistic topologies and routing environments.

A major challenge in reproducing realistic Internet-scale topologies in a testbed setting is the scale-down of a topology of several thousands or even millions of nodes to a few hundred nodes (which is the number of nodes available on a testbed like DETER [3]), while retaining relevant topology characteristics. In our RocketFuel-to-ns tool, we allow a user to specify a set of Autonomous Systems, or to perform a breadth-first traversal of the topology graph from a specified point, with specified degree and number-of-nodes bounds. This enables the user to select portions of very large topologies containing only tens of nodes up to a few hundred nodes, and use them for testbed experimentation. The RouterConfig tool works both on (a) topologies based on real Internet data, and on (b) topologies generated from the GT-ITM topology generator [10]. We selected GT-ITM since it generates representative topologies, even when the number of nodes in the topology is small [11]. One major focus of our future research lies in defining how to accurately scale down DDoS experiments, including the topology dimension.

Another challenge in defining realistic topologies lies in assigning realistic link delays and link bandwidths, because such data, especially within an enterprise network, is not public, and it is sometimes impossible to infer. Tools such as [12], [13], [14], [15] have been proposed to measure *end-to-end* bottleneck link capacity, available bandwidth, and loss characteristics. Standard tools such as ping and traceroute can produce end-to-end delay or *link delay* information, if their probe packets are not dropped by firewalls. Identifying *link bandwidths* is perhaps the most challenging problem. Therefore, we use information about typical link speeds (optical links, Ethernet, T1/T3, DSL, cable modem, dial up, etc.) published by the Annual Bandwidth Report [16], to assign link bandwidths in our benchmark topologies.

For localized defense testing, it is critical to characterize enterprise topologies and identify services running in an enterprise network, in order to accurately represent them in our benchmarks. Towards this goal, we analyzed enterprise network design methodologies typically used in the commercial marketplace to design and deploy scalable, cost-efficient production networks. An example of this is Cisco's classic three-layer model of hierarchical network design that is part of Cisco's Enterprise Composite Network Model [17], [18]. This consists of the topmost core layer which provides Internet access and ISP connectivity choices, and a middle distribution layer that connects the core to the access layer and serves to provide policy-based connectivity to the campus as well as hide the complexity of the access layer from the core. Finally, the bottom access layer addresses the design of the intricate details of how individual buildings, rooms and work groups are provided network access and typically involves the layout of switches and hubs.

Our analysis of the above commercial network design methodologies leads us to conclude that there exist at least six major aspects (decisions) that impact enterprise network design: (1) the edge connectivity design that determines whether the enterprise is multi-homed or single-homed; (2) network addressing and naming and in particular if internal campus addresses are private or public and routable, and if such addresses are obtained as part of ISP address block assignments; (3) the design of subnet and virtual local area networks (VLANs); (4) the degree of redundancy required at the distribution layer; (5) load sharing requirements across enterprise links and servers, and (6) the placement and demands of security services such as virtual private networks (VPNs) and firewall services.

Our long-term goal is to study through experiments how network topology properties impact the manner in which various DDoS attacks play out in real enterprise networks and how they impact the efficacy of various DDoS defenses. Interesting questions to consider include: (1) How does IP address assignment affect traffic filtering choices and the efficacy of filter-based DDoS mitigation? (2) Can redundant and backup links and routes mitigate certain DDoS floods? (3) Can dynamic load balancing across redundant links mitigate flooding attacks? (4) Can load balancing across a server farm mitigate DDoS attacks on servers? (5) Can asymmetric routes reduce the feasibility and efficiency of DDoS trace-back or that of a collaborative defense scheme? (6) Do subnet and VLAN structures provide containment against attacks that use broadcast, multicast and amplification features? and (7) What makes a DDoS attack effective on multiple geographically dispersed campuses? What type of defense mitigation would work best and where should it be placed?

VI. Performance Metrics

To evaluate DDoS defenses, we must define an effectiveness metric that speaks to the heart of the problem —- *do these defenses remove the denial-of-service effect*. Several metrics have been used in network literature for this purpose, including the percentage of attack traffic dropped and the amount of legitimate traffic delivered, legitimate traffic's goodput, delay and loss rate. These metrics fail to capture the most important aspect of a DDoS defense, which is whether legitimate service continues at a user-acceptable level during the attack. Even if all the attack traffic is dropped, a defense that does not ensure delivery and prompt service for legitimate traffic does not remove the DoS effect, and the attack still succeeds.

A better metric is a percentage of legitimate packets delivered during the attack. If this percentage is high, arguably service continues with little interruption. However, this metric does not capture the fact that loss of particular packets (e.g., TCP SYN or media control packets), even small numbers of them, can cripple some services, and it does not measure delay and its variation that can seriously degrade interactive traffic.

We propose a metric that speaks to the heart of the problem: did the legitimate clients receive acceptable service or not? This metric requires considering traffic at the application level and defining quality of service needs of each application. Specifically, some applications have strict delay, loss and jitter requirements and will be impaired if any of these are not met. In the QoS literature, these applications are known as *intolerant real-time* applications. Other real-time applications have somewhat relaxed delay and loss requirements, and are known as *tolerant real-time* applications. Finally, there are applications that conduct their transactions without human attendance and can endure significant loss and delay as long as their overall duration is not significantly prolonged. These applications are classified in the QoS literature as *elastic*. In our metrics definition, we preserve and extend known QoS classifications, and define QoS thresholds for each application category, mostly borrowing from [19].

We measure the overall denial-of-service impact in the following manner. We extract transaction data from the traffic traces captured at the legitimate sender during the experiment. A *transaction* is defined as a high-level task that a user wanted to perform, such as viewing a Web page, downloading a file, conducting a telnet session or having a VoIP conversation. Each transaction is categorized by its application, and we determine if it experienced a DoS effect by evaluating if the application's QoS requirements were met. Transactions that do not meet QoS requirements are labeled as "failed" and the rest are labeled as "succeeded". The DoS impact measure expresses the percentage of transactions, in each application category, that have failed. An effective defense should minimize DoS impact by reducing the percentage of failed transactions.

The advantage of the above-proposed metric lies in its intuitive nature — by directly measuring the denial-of-service impact, we can objectively ascertain how effective a defense is in neutralizing this impact. The proposed metric requires (1) determining which applications are most important, both by their popularity among Internet traffic and the implications for the rest of the network traffic if these applications are interrupted, and (2) determining acceptable thresholds for each application that, when exceeded, indicate a denial-of-service. Both tasks are very challenging since the proposed applications and thresholds must be acceptable to the majority of research and commercial actors, to make the DoS impact metric widely used.

In addition to measuring a defense's effectiveness, the defense performance metric must also capture the delay in detecting and responding to the attack, the deployment and operational cost and the defense's security against insider and outsider threats. Each of these performance criteria poses unique challenges in defining objective measurement approaches. Yet, these challenges must be explored and overcome to develop a common evaluation platform for DDoS defenses.

VII. Measurement Methodology

Any set of benchmarks needs continuous update as attack trends and network connectivity and usage patterns change. Our automated AProf, LTProf and NetProf tools can be used to generate new typical benchmark suites from future traffic traces.

The benchmark suite will contain a myriad of test scenarios, and our proposed metrics will produce several performance measures for a given defense in each scenario.

A measurement methodology will provide guidelines on aggregating results of multiple measurements into one or a few meaningful numbers. While these numbers cannot capture all the aspects of a defense's performance, they should offer quick, concise and intuitive information on how well this defense handles attacks and how it compares to its competitors. We expect that the definition of aggregation guidelines will be a challenging and controversial task, and we plan to undertake it after our benchmark suite and performance metrics work are completed.

VIII. Related Work

The value of benchmarks for objective and independent evaluation has long been recognized in many science and engineering fields [20], [21], [22]. Recently, the Internet Research Task Force (IRTF) has chartered the Transport Modeling Research Group (TMRG) to standardize the evaluation of transport protocols by developing a common testing methodology, including a benchmark suite of tests [23].

In the computer security field, the Center for Internet Security has developed benchmarks for evaluation of operating system security [24] and large security bodies maintain security checklists of known vulnerabilities that can be used by software developers to test the security of their code [25], [26], [27]. While the existing work on security benchmarks is to be commended, much remains to be done to define rigorous, clear and representative tests for various security threats. This is especially difficult in the denial-of-service field as there are many ways to deny service and many variants of attacks, while the impact of a given attack on a target network further depends on various network characteristics including its traffic and resources.

There is a significant body of work in the Quality of Service (QoS) field that is relevant to our definition of transaction success for DDoS defense performance metrics. Internet traffic has traditionally been classified according to the application generating it. A representative list of applications includes video, voice, image and data in conversational, messaging, distribution and retrieval modes [28]. These applications are either inelastic (real time) which require end-to-end delay bounds, or elastic, which can wait for data to arrive. Real time applications are further subdivided into those that are intolerant to delay, and those that are more tolerant, called delay adaptive.

The Internet integrated services framework mapped the three application types (delay intolerant, delay adaptive and elastic) onto three service categories: the guaranteed service for delay intolerant applications, the controlled load service for delay adaptive applications, and the currently available best effort service for elastic applications. The guaranteed service gives firm bounds on the throughput and delay, while the controlled load service tries to approximate the performance of an unloaded packet network [29]. Similarly, the differentiated services (DiffServ) framework standardized a number of Per-Hop Behaviors (PHBs) employed in the core routers, including a PHB, expedited forwarding (EF), and a PHB group, assured forwarding (AF) [30], [31]. In the early 1990s, Asynchronous transfer mode (ATM) networks were designed to provide six service categories: Constant Bit Rate (CBR), real-time Variable Bit Rate (rt-VBR), non real-time Variable Bit Rate (nrt-VBR), Available Bit Rate (ABR), Guaranteed Frame Rate (GFR) and Unspecified Bit Rate (UBR) [32]. For example, the network attempts to deliver cells of the rt-VBR class within fixed bounds of cell transfer delay (max-CTD) and peak-to-peak cell delay variation (peak-to-peak CDV). The traffic management specifications [32] defined methods to measure such quantities so that users can ensure they are receiving the service they had paid for.

Selecting representative benchmark topologies with realistic routing parameters and realistic resources and services is an extremely challenging problem [33]. Internet topology characterization has been the subject of significant research for over a decade [10], [34], [35], [36], [37], [38]. Several researchers have examined Internet connectivity data at both the Autonomous System (AS) level and at the router level, and characterized the topologies according to a number of key metrics. One of the earliest and most popular topology generators was GT-ITM [10], which used a hierarchical structure of transit and stub domains. Later work examined the degree distribution of nodes, especially at the Autonomous System level, and characterized this distribution as what is typically referred to as "the power law phenomenon" [34], [35], [36]. Clustering characteristics of the nodes were also examined, and the term "the small world phenomenon" [39], [37], [40] was used to denote preference to local connectivity. Recent work [41] uses joint degree distributions to capture different topology metrics such as assortativity, clustering, rich club connectivity, distance, spectrum, coreness, and betweenness.

Several Internet researchers have attempted to characterize Internet denial-of-service activity [42], [43]. Compared to our work on attack benchmarks, they used more limited observation approaches and a single traffic trace. Moreover, both of these studies were performed several years ago, and attacks have evolved since then.

Finally, there is a significant body of work on traffic modeling [44], [45], [46] but there is a lack of unifying studies that observe communication patterns across different networks and the interaction of this traffic with denial-of-service attacks. Our work aims to fill this research space.

IX. Conclusions and Future Work

While we have performed substantial work to define good benchmarking procedures for evaluating DDoS defenses, much work remains. The major technical challenges lie in the following four directions: (1) collecting sufficient trace and topology data to generate typical test suites, (2) understanding the interaction between the traffic, topology and resources and designing comprehensive yet manageable test sets, (3) determining a success criteria for each application, and (4) defining a meaningful and concise result aggregation strategy. The value of any benchmark lies in its wide acceptance and use. The main social challenge we must overcome lies in having all three components of our common evaluation methodology widely accepted by research and commercial communities.

Our existing methods have some limitations. Many of the inputs to the benchmark definitions rely on trace analysis. While such methods have the virtue of deriving information from real network events, they have the clear disadvantage that we can only analyze what the traces show us. Only a limited number of traces are currently publicly available, and they do not necessarily cover the range of important points on the Internet. Further, they are not necessarily complete (they capture a large volume of one-way traffic, and they will miss packets that were dropped during an attack before they reached the trace point), and the anonymization typically used in traces hides some information that would be useful in benchmark definition (such as the NLANR-PMA trace, which does not allow us to distinguish nodes inside their network from nodes outside it). We hope to overcome these limitations through cooperation with a DHS-funded PREDICT project [47], which collects traffic traces from major ISPs and enables trusted researcher access to this data.

Designing benchmarks for DDoS defenses is sure to be an ongoing process, both because of these sorts of shortcomings in existing methods and because both attacks and defenses will evolve in ways that are not properly captured by the benchmarks we and others initially define. However, there are currently no good methods for independent evaluation of DDoS defenses, and our existing work shows that defining even imperfect benchmarks requires substantial effort and creativity. The benchmarks described in this paper represent a large improvement in the state of the art for DDoS defense evaluation and a significant first step towards a common evaluation methodology. With input from research and commercial communities, we expect to further refine this methodology.

References

[1] Ryan Naraine. Massive DDoS attack hit DNS root servers. http://www.internetnews.com/dev-news/article.php/1486981.

[2] Ann Harrison. Cyberassaults hit Buy.com, eBay, CNN, and Amazon.com. Computerworld, February 9, 2000 http://www.computerworld.com/news/2000/story/0,11280,43010,00.html.

[3] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, and S. Schwab. Experiences With DETER: A Testbed for Security Research. In *2nd IEEE Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities (TridentCom 2006)*, March 2006.

[4] BindView Corporation. *The Naptha Dos Vulnerabilty*, November 2000. Available at http://www.bindview.com/Support/RAZOR/Advisories/2000/adv_NAPTHA.cfm.

[5] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. In *In Proceedings of ACM SIGCOMM 2000*, August 2000.

[6] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with RocketFuel. In *Proceedings of ACM SIGCOMM*, 2002.

[7] Traceroute.org. Traceroute tool, 2006. http://www.traceroute.org.

[8] L. Gao. On inferring autonomous system relationships in the Internet. In *Proc. IEEE Global Internet Symposium*, November 2000.

[9] F. Wang and L. Gao. On inferring and characterizing Internet routing policies. In *Proc. Internet Measurement Conference (Miami, FL)*, October 2003.

[10] E. Zegura, K. Calvert, and S. Bhattacharjee. How to Model an Internetwork. In *Proc. of IEEE INFOCOM*, volume 2, pages 594 –602, March 1996.

[11] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: Degree-based vs. structural. In *Proceedings of ACM SIGCOMM*, 2002.

[12] K. Lai and M. Baker. Nettimer: A Tool for Measuring Bottleneck Link Bandwidth. In *Proc. of USENIX Symposium on Internet Technologies and Systems*, March 2001.

[13] C. Dovrolis and P. Ramanathan. Packet dispersion techniques and capacity estimation. *IEEE/ACM Transactions on Networking*, December 2004.

[14] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *Proceedings of ACM IMC*, October 2003.

[15] R. Mahajan, N. Spring, David Wetherall, and Thomas Anderson. User-level internet path diagnosis. In *Proceedings of ACM SOSP*, October 2003.

[16] Websiteoptimization.com. *The Bandwidth Report*. http://www.websiteoptimization.com/bw/.

[17] Priscilla Oppenheimer. *Top-Down Network Design*. CISCO Press, 1999.

[18] Russ White, Alvaro Retana, and Don Slice. *Optimal Routing Design*. CISCO Press, 2005.

[19] Nortel Networks. *QoS Performance requirements for UMTS*. The 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/ftp/tsg_sa/WG1_Serv/TSGS1_03-HCourt/Docs/Docs/s1-99362.pdf.

[20] Project 2061. Benchmarks On-Line. http://www.project2061.org/publications/bsl/online/bolintro.htm.

[21] Standard Performance Evaluation Corporation. SPEC Web Page. http://www.spec.org.

[22] Transaction Processing Performance Council. TPC Web Page. http://www.tpc.org/.

[23] IRTF TMRG group. The Transport Modeling Research Group's Web Page. http://www.icir.org/tmrg/.

[24] The Center for Internet Security. CIS Standards Web Page. http://www.cisecurity.org/.

[25] CERT. CERT UNIX Security Checklist v2.0. http://www.cert.org/tech_tips/usc20_full.html.

[26] Microsoft Corporation. Security Tools. http://www.microsoft.com/technet/security/tools/default.mspx.

[27] SANS. The SANS Top 20 Internet Security Vulnerabilities. http://www.sans.org/top20/.

[28] M. W. Garrett. Service architecture for ATM: from applications to scheduling. *IEEE Network*, 10(3):6–14, May/June 1996.

[29] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633, June 1994. http://www.ietf.org/rfc/rfc1633.txt.

[30] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597, June 1999. http://www.ietf.org/rfc/rfc2597.txt.

[31] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC 2598, June 1999. http://www.ietf.org/rfc/rfc2598.txt.

[32] The ATM Forum. The ATM Forum Traffic Management Specification Version 4.0. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps, April 1996.

[33] K. Anagnostakis, M. Greenwald, and R. Ryger. On the Sensitivity of Network Simulation to Topology. In *Proc. of MASCOTS*, 2002.

[34] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proc. of ACM SIGCOMM*, pages 251–262, August 1999.

[35] T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. In *Proc. of IEEE INFOCOM*, June 2002.

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

[37] S. Jin and A. Bestavros. Small-world Characteristics of Internet Topologies and Multicast Scaling. In *Proc. of IEEE/ACM MASCOTS*, 2003.

[38] J. Winick and S. Jamin. Inet-3.0: Internet Topology Generator. Technical Report UM-CSE-TR-456-02, Univ. of Michigan, 2002.

[39] D. Watts and S. Strogatz. Collective Dynamics of Small-world Networks. *Nature*, 363:202–204, 1998.

[40] A. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, 1999.

[41] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K. Claffy, and A. Vahdat. The internet AS-level topology: Three data sources and one definitive metric. Technical report, University of California, San Deigo, 2005. Short version appears in ACM CCR, January 2006.

[42] D Moore, G Voelker, and S Savage. Inferring Internet Denial-of-Service Activity. Proceedings of the 2001 USENIX Security Symposium, 2001.

[43] Kun chan Lan, Alefiya Hussain, and Debojyoti Dutta. The Effect of Malicious Traffic on the Network. In *Passive and Active Measurement Workshop (PAM)*, April 2003.

[44] Bell Labs. Bell Labs Internet Traffic Research. http://stat.bell-labs.com/InternetTraffic/index.html.

[45] ICSI Center for Internet Research. Traffic Generators for Internet Traffic. http://www.icir.org/models/trafficgenerators.html.

[46] Hei Xiaojun. Self-Similar Traffic Modelling in the Internet. http://www.ee.ust.hk/~heixj/publication/comp660f/comp660f.html.

[47] RTI International. Protected Repository for the Defense of Infrastructure against Cyber Threats. http://www.predict.org/.