Power to the People:
Securing the Internet One Edge at a Time[*]

Soon Hin Khor[†]
Carnegie Mellon University
Information Networking
Institute and CyLab Japan
Kobe, Hyogo 650-0044, Japan
skhor@andrew.cmu.edu

Nicolas Christin
Carnegie Mellon University
Information Networking
Institute and CyLab Japan
Kobe, Hyogo 650-0044, Japan
nicolasc@andrew.cmu.edu

Tina Wong
Carnegie Mellon University
Information Networking
Institute
Pittsburgh, PA 15213, USA
tinaw@andrew.cmu.edu

Akihiro Nakao
University of Tokyo
Interdisciplinary Information
Studies
Tokyo 113-0033, Japan
nakao@iii.u-tokyo.ac.jp

ABSTRACT

Despite a plethora of research in the area, none of the mechanisms proposed so far for Denial-of-Service (DoS) mitigation has been widely deployed. We argue in this paper that these deployment difficulties are primarily due to economic inefficiency, rather than to technical shortcomings of the proposed DoS-resilient technologies.

We identify economic phenomena, *negative externality*—the benefit derived from adopting a technology depends on the action of others—and *economic incentive misalignment*—the party who suffers from an economic loss is different from the party who is in the best position to prevent that loss—as the main stumbling blocks of adoption. Our main contribution is a novel DoS mitigation architecture, *Burrows*, with an *economic incentive realignment* property. Burrows is obtained by re-factoring existing key DoS mitigation technologies, and can increase the "social welfare," i.e., economic benefit, of the entire Internet community—both infrastructure providers and the Internet users.

At the core of Burrows is a wide-area virtual private network, or secure overlay, carved out of the existing Internet. Entry points into the Burrows overlay are controlled by gateways, which in addition to providing connectivity, minimize negative externality flowing between Burrows and the Internet. To rectify the aforementioned economic incentive misalignment, the power to realize Burrows is put into the hands of the Internet users. In addition, Burrows supports incremental deployment: even with as few as two participants, Burrows provides an environment more secure than without it.

Categories and Subject Descriptors

C.2.0 [**Security and Protection**]: Denial of Service; C.2.1 [**Network Topology**]: Overlay Networks

General Terms

Security, Reliability, Economics

Keywords

Network Security, Denial of Service Attacks, Overlay Networks

1. INTRODUCTION

The phenomenal growth of the Internet owes much to the simplicity of its design principles, which allow to widely interconnect heterogeneous systems. The straightforwardness of the interconnection primitives, however, also proves to be the main security chink of the Internet. In particular, the Internet's original design principles do not provide any form of control for a server to dictate how much traffic it wants to receive and from whom. As a result, Internet hosts are vulnerable to Denial-of-Service (DoS) and Distributed-Denial-of-Service (DDoS) attacks, whose economic and social impact has grown to considerable proportions. In 2000, the Yankee Group estimated a loss of $1.2 billion among Yahoo, eBay, Buy.com and Amazon due to a coordinated DoS[1]. The annual infrastructure security report compiled by ArborNetworks [16] shows that DoS and Bots (platforms from which DoS can be launched) top the chart as the primary security concerns at 46% and 31% respectively out of 55 respondents.

Mechanisms to mitigate or prevent DoS attacks have accordingly received significant attention in the research community, resulting in considerable technological advances. However, due to its economic properties, the Internet remains largely vulnerable to DoS

[1]We will use the term DoS to refer to both DoS and DDoS.
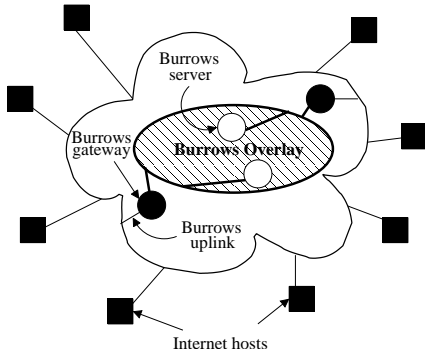
Figure 1: High-level architectural overview. Hosts participating in Burrows only connect with the rest of the Internet through dedicated Burrows gateways.

attacks. Indeed, because the Internet is a common property, it requires collaboration from multiple parties to fully secure it, an issue known as *externality*. Almost all DoS mitigating mechanisms thus far have not fully accounted for externality and have therefore faced an uphill battle in gaining acceptance, eventually resulting in a less than effective deployment.

Extrapolating Anderson's observation [4] on *economic incentive misalignment* into the Internet domain, we find another reason why the Internet has remained insecure despite much research. The party in the best position to enforce security (infrastructure providers) is different from the party who will mostly suffer if there is a lack of security (end users as well as content providers). Since most DoS mitigation mechanisms were designed for infrastructure providers who do not have incentives to deploy them [5], those mechanisms have mostly remained as research artifacts.

Considering adequate DoS mitigation mechanisms already exist, we posit that it is crucial to identify key available mechanisms and re-factor them to include an economic incentive realignment mechanism. Our core contribution is to propose *Burrows*, an overlay network that (1) minimizes negative externality, i.e., enables an end-user to be secure without being affected by the (in)action of those who are indifferent to Internet security, and (2) rectifies economic incentive misalignment, i.e., empowers end-users to deploy this mechanism without relying on infrastructure providers.

A high-level overview of Burrows is given in Figure 1. End-users install servers that require security protection in the Burrows overlay. These servers, represented by white circles in Figure 1. are called Burrows servers, and are only connected to the Burrows overlay network. All communications between systems outside the overlay (i.e., Internet hosts, depicted by black boxes in the figure) and Burrows servers are controlled by Burrows gateways, denoted by black circles in the figure.

By completely shielding Burrows servers from the rest of the Internet, the Burrows overlay and gateways not only protect participating servers from direct attacks launched from Internet hosts, but also minimize the negative externality flowing from the Internet to the Burrows servers. Employing a peer-to-peer model, where each end-user who protects her server using Burrows is required to contribute a Burrows gateway, rectifies the economic incentive misalignment by empowering end-users to build Burrows without assistance from the infrastructure providers.

The rest of the paper elaborates on the Burrows architecture as follows. Section 2 reviews the current attempts to mitigate DoS. Section 3 defines our design objectives, by identifying key DoS

mitigation properties both from security and from economic point of views. Section 4 follows up with the details of the Burrows architecture and Section 5 conducts its feasibility study. Section 6 discusses how Burrows can be beneficial to both infrastructure providers and end-users. Sections 7 and 8 conclude the paper by describing future work and summarizing our contributions.

2. RELATED WORK

The vulnerability of the Internet to DoS attacks has driven a large number of research efforts on DoS mitigation. DoS mitigation mechanisms can be roughly categorized based on "where they are deployed" and "whether collaboration is required" [7]. Mechanisms designed without collaboration and deployed at an end-point often leave the end-point's uplink unprotected, while collaborative mechanisms can provide both end-point and uplink protection, but require that economic incentives of all the participants be aligned. This section discusses some of the proposals most closely related to our own.

Replication of content to multiple nodes, as in Coral [10], Akamai [13], and CoDeeN [38], increases availability, thereby reducing the effectiveness of DoS attacks. Although such proposals are elegant, they require existing applications to be rewritten to fit into content replication models and have limited protocol support (usually HTTP and/or streaming content). Replicating contents of databases and email servers will require a different model because of the content sensitivity and the session-oriented nature of those applications.

A more general-purpose solution, which can support a wide range of protocols, is to use a Resilient Overlay Network (RON) [3]. RON consists of end-systems running software routers which constantly evaluate the path metrics among themselves to achieve better resilience cooperatively. Although RON primarily aims to failover network link failures, it is arguably able to detect and route around DoS hotspots. However, RON does not address the situation where the end-points themselves become the target of DoS.

Secure Overlay Services (SOS) [14], Mayday [2], and WebSOS [32] all seek to protect end-points from DoS attack by filtering traffic deep inside the Internet infrastructure where bandwidth is abundant. The receiving end-point can choose which nodes to receive traffic from and the type of filtering mechanism to employ. Since filtering nodes belong to infrastructure owners while receiving end-points are owned by end-users, cooperation between infrastructure owners and end-users becomes a necessity. However, the lack of economic incentive for the infrastructure providers to provide such filtering capabilities may lead to a deployment impasse.

XenoService [42] recognizes the economic problem arising from proposals that involve multiple parties and attempts to also deliver a standardized replication mechanism for all types of servers. The authors propose to replicate applications by running them on multiple servers, which various applications can share to increase availability and mitigate DoS attacks. Replication of interactive servers is however acknowledged to be a difficult problem.

There are currently DDoS mitigation service available in the market for a premium, as documented in the annual Managed Security Service Providers (MSSP) survey carried out by `ISP-planet.com` [12]. However, the economic incentive issues arising from negative externality is evidenced by the scarcity of MSSPs that offer DoS mitigation service. In fact, only one out of the fifteen respondents provided the service, and not surprisingly that MSSP is a backbone ISP which owns many Internet nodes. Indeed, noncollaborative (e.g., proprietary) DoS mitigation architectures can only be effective if the service provider that deploys them owns a large number of nodes, and even in such a case, remain at the

mercy of an attacker going around the controlled nodes. Hence, DoS mitigation architectures need to be incentive-compatible to ensure collaboration between competing entities. Stated differently, a platform to enable even non backbone ISP MSSPs to deploy DoS mitigation solutions using shared nodes over a wider area is a necessity for competition to exist and push DoS technologies towards maturity. Failing that, there is a non-negligible risk that "secured" networks end up being disconnected from the rest of the Internet for the sake of security, returning network connectivity to the undesirable state in which it was before the advent of the Internet, that is, a collection of disconnected networks with limited interoperability.

Finally, recent work argues that further progress in the innovations on the Internet has plateaued and will be stifled without a clean slate design [5, 22, 21]. A complete re-design of the Internet may resolve its shortcomings including the lack of security, but such a radical change needs accumulation of careful thoughts and experiences with incremental technologies. History suggests that make-shift technologies like Network Address Translation (NAT) and Wi-Fi Protected Access (WPA) provide valuable deployment feedback to iteratively drive standards such as IPv6 and 802.11i to maturity. We believe that Burrows can likewise influence the efforts to design a future Internet infrastructure.

3. DESIGN OBJECTIVES

In this section, we discuss the objectives a DoS mitigation architecture with economic incentive realignment should strive to fulfill. To that effect, we first define the scope of the DoS threat, then identify the *security properties* required to defend against the threat and describe the *economic properties* necessary to stimulate adoption of the design. While some of the properties may have been addressed in past research, their deployability has not necessarily been fully discussed. Thus, we elect to revisit them and consider issues in their deployment.

3.1 DoS Attack Definition

A denial-of-service attack is characterized by an explicit attempt to prevent the legitimate use of a service [6]. In this paper, we focus the scope of our proposal on defending against DoS attacks that can be launched from commonly utilized DoS tools, e.g., Shaft, Trinoo and Tribe Flood Network [9]. The range of DoS attacks include UDP, ICMP echo, TCP SYN and Smurf attack [18]. This set of DoS attacks cover between 56% and 74% of all DoS attacks experienced [16], which suggests it is an adequate benchmark for evaluation.

3.2 Security Properties

We next discuss security properties, especially *what* needs to be protected and the *type* of protection required.

3.2.1 Protected Entities

Server Protection   One common type of DoS attack simply consists of flooding a server with more requests than it can handle. Hence, a server that can dictate how much traffic it should receive will not be susceptible to DoS. Even though technologies like "pushback" [17] and Internet Indirection Infrastructure (i3 [33]) can be used for such a purpose, they require changes to the existing Internet infrastructure, which raises deployment concerns. Proposals requiring radical architectural changes indeed experience an increasingly difficult path to adoption, due to the ossification of the Internet [20].

Uplink Protection   It may be possible to protect a server from DoS attacks by constraining direct connectivity to the server, using, for instance, a Virtual Private Network (VPN). However, even if the server is not directly vulnerable, DoS can still occur if the Internet uplink of the server is flooded. This situation can notably occur if an attack is targeted at the router that connects the server to its Internet uplink. Thus, it is crucial that the uplink router itself be shielded from attacks, an issue generally ignored by related proposals. Also note that the uplink protection must be implemented in routers, and its wide deployment is likewise hindered.

Traffic Protection   The third key component to protect against DoS attacks is the traffic itself. DoS traffic which masquerades as legitimate traffic and as such cannot be easily filtered, e.g., a few million valid HTTP requests targeted at a web server, may prevent valid traffic from reaching its destination, and must be filtered. CAPTCHAS [36] and "fight fire with fire" [37] are two existing technologies that address this issue. However, CAPTCHAS requires modifications to the existing server while Walfish *et al.* acknowledge that for "fight fire with fire" to be deployed, a front-end server with enormous capacity is needed to accept all requests, i.e., both DoS and legitimate. Whether or not deploying such a powerful front-end is feasible remains an open problem.

3.2.2 Protection Type

Protection Independent of Offered Service   Some DoS protection architectures are based on content replication as in Akamai. Replicating content is (thus far) mostly limited to HTTP and streaming traffic. Conversely, we strive for an architecture that can be resilient to DoS attacks regardless of the contents being served.

Ingress Filtering   An attack should be filtered as near its source as possible, to minimize the use of Internet resources by unwanted traffic. A simple way to do this is for all the networks connected to the Internet to cooperate by performing stateful filtering to detect and drop DoS packets at Internet ingress points. However, according to the MIT Spoofer Project [19], using cooperative ingress filtering mechanism (here, to prevent IP source spoofing) is extremely hard to deploy as it requires total collaboration to be effective. To be worse, due to the lack of efforts from parties indifferent to security, Internet ingress DoS filtering becomes ineffective even to those who invest resources to implement them. Reducing such negative externality to implement ingress filtering mechanism, to the best of our knowledge, has not been addressed thus far.

Traceback Mechanism   Due to IP's inability to authenticate packet origins, perpetrators of attacks cannot be reliably identified from the source IP addresses of malicious packets. Even though traceback mechanisms to identify attackers are well-researched, e.g., edge sampling [29], Single Packet IP Traceback [31], and Fast Internet Traceback [41], they again require modification to existing Internet infrastructure components such as routers, which is an obstacle to their deployment.

3.3 Economic Properties

In addition to the security properties outlined above, a DoS-resilient architecture must adhere to certain economic properties to be deployable.

Minimizing Negative Externality   In designing an architecture for adoption on the Internet, we have to ensure that the effectiveness of the architecture is minimally affected by people who choose not to adopt the architecture. Failing to do so will result in the architecture being overlooked by even its most ardent advocates.

Realigning Economic Incentives   Infrastructure providers may not experience direct economic losses due to DoS, while end-users with web presence do. Since end-users are more likely to have incentives to deploy DoS mitigation mechanisms, we need an architecture which empowers end-users with the ability to secure their servers without requiring aid from infrastructure providers.

Small "Critical Mass" Effectiveness   Most DoS mitigation mechanisms require substantial deployment size, or "critical mass," to be effective. Building critical mass requires time and effort. Instead, we strive to design for incremental deployment that ensures that even with as few as two participants, both participants extract tangible benefits from their involvement.

Backwards Compatibility   We need existing servers to be able to utilize our DoS mitigation mechanism with little or no modification. We also need the mechanism to be completely transparent to existing clients. Indeed, unless the architecture has an extremely compelling property, e.g., total DoS elimination, an *evolutionary* (backwards compatible) solution is more likely to gain acceptance than a *revolutionary* one [30].

4. PROPOSED DESIGN

In this section, we describe the architecture of Burrows and discuss how we fulfill the objectives described in Section 3. Then, we walk through an example of how Burrows works and describe additional architectural components, which may not be directly used to achieve our design objectives, but remain important.

4.1 Burrows Architecture

Throughout the design of Burrows, we ensure that our architecture requires minimal changes to the existing servers and clients. Designing Burrows as an overlay network makes it possible to embed various DoS mitigation mechanisms (see following subsections) to achieve the properties identified in Section 3 without modification to the current Internet.

Server and Uplink Protection   As depicted in Figure 1, at the core of Burrows is a "private" overlay network. Systems within the private overlay network are interconnected via Border Gateway Protocol Multi-Protocol Label Switching Virtual Private Network (BGP-MPLS-VPN) [28], which is usually utilized to interconnect multiple remote locations to form a single private network. In Burrows, BGP-MPLS-VPN is used to provide Burrows servers and their uplinks protection from direct DoS attacks. This is possible since the systems in BGP-MPLS-VPN only have private IP addresses.

Since VPN protects Burrows servers from direct reachability, connectivity from an Internet host (i.e., a system outside Burrows) to Burrows servers is achieved through Burrows gateways and via uplinks between Burrows gateways and Burrows servers. Burrows gateways not only protect a server against direct attacks, but also make its uplink DoS-resilient. Indeed, each Burrows gateway added replicates an uplink to the server, so that DoS on a server may only happen if all of the server's uplinks are simultaneously flooded.

In addition, the gateways can be equipped with pushback mechanism to enable the Burrows server to control how much traffic each gateway is allowed to route to the server.

Ingress Filtering and Traceback Mechanism   With a private VPN, we can filter a packet so that it may only be routed into Burrows (1) if its source IP address corresponds to that of a Burrows gateway or a Burrows server *and* (2) if it enters Burrows from the physical network interface of the router to which the Burrows gateway (or server) is known to connect. This prevents source IP spoofing for traffic coming to/from Burrows and therefore obviates the need for complex traceback mechanisms within Burrows. Indeed, misbehaving Burrows servers and Burrows gateways can be identified from the malicious packets' source IP addresses. Furthermore, by only deploying ingress filtering at Burrows gateways, which do have economic incentives (server protection) to implement such filtering, we avoid economic inefficiencies. Lastly, having multiple gateways not only makes filtering more effective, but also helps to
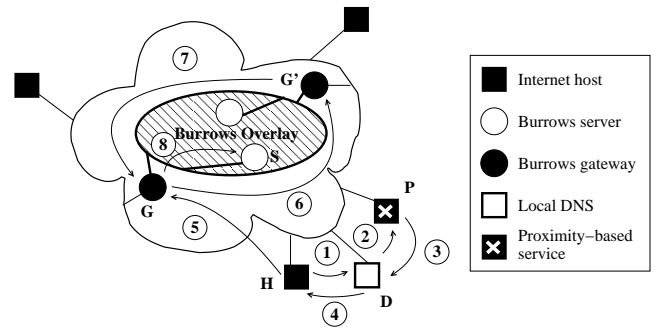
Figure 2: High-level walkthrough of Burrows

filter DoS attacks as close to the sources as possible.

Traffic Protection   Using multiple gateways also provides a possible way to construct a front-end server powerful enough to implement a "fight fire with fire" mechanism [37] and filter out automated DoS traffic. In the "fight fire with fire" mechanism, a front-end system receives all (legitimate or not) requests on behalf of a server and only forwards the number of requests that the Burrows server can support while dropping the rest. During a DoS attack, the front-end will demand retransmission for all requests. Since DoS bots are already utilizing their bandwidth to the maximum, the volume of DoS traffic stays at the same level while the retransmission rate of legitimate requests increases, thereby enhancing the probability that legitimate clients can obtain services even under DoS.

Small Critical Mass Effectiveness   Even with a minimum of two gateways, a Burrows server acquires the benefit of having twice the capacity (two Burrows uplinks) to resist DoS than it had before (its sole Internet uplink). We use VPN to force connectivity to Burrows servers through gateways and then use multiple gateways to achieve the effect of uplink replication and small critical mass effectiveness.

Incentive Realignment and Minimal Negative Externality   We adopt a *self-scaling* model (as seen in peer-to-peer applications) to empower end-users to build Burrows thereby realigning economic incentives, that is, each participant is entitled to harbor her server in Burrows only if she contributes at least one Burrows gateway. Through this model, it is likely that Burrows comprises only the security-conscious participants. We accordingly decrease the probability that insecure systems exist within Burrows, further reducing negative externality. PlanetLab [24] is an extremely successful example of an infrastructure built incrementally using a self-scaling model with a similar incentive alignment mechanism.

4.2 High-Level Walkthrough

When traffic flows in and out of Burrows, there are three possible connectivity scenarios; (1) an Internet host initiates a connection to a Burrows server, (2) a Burrows server initiates a connection to another Burrows server, and (3) a Burrows server initiates a connection to a Internet host. In Figure 2, we walk through the first scenario, while forgoing the latter two since they do not provide additional insights to mitigating DoS.

An Internet host H that needs to communicate with a server protected by Burrows is "unaware" of the existence of Burrows. It resolves the Burrows server hostname using its local DNS server D (Step 1 in Figure 2). The local DNS must be linked to a proximity-based service P (Step 2) such as OASIS [11] and Meridian [40, 39] in order to find the nearest Burrows gateway which can serve the

Internet host using that local DNS. The IP address of the nearest Burrows gateway is returned to the querying Internet host (Steps 3 & 4).

The Internet host now communicates with the Burrows gateway G (Step 5) as if the Burrows gateway were the server it wants to interact with. The gateway will receive packets addressed to it but whose final destinations are not itself but Burrows servers. Therefore, the gateway has to look at the application layer to find the hostname of final destination.

Once the hostname of the destination server of the packet is found, the Burrows gateway needs to resolve the hostname again to get the actual IP address of the Burrows server. The Burrows gateway queries a distributed domain name service (DDNS) such as CoDoNS [26] to obtain the answer.[2] In the figure, DDNS resolution is represented by steps 6 and 7. Note that the DDNS service itself is distributed over the set of Burrows gateways; in the example of Figure 2, a gateway G' is able to return the answer to G. Finally, the Burrows gateway G forwards the packet to the actual Burrows server S to complete the connection (Step 8). The connection state is kept at the gateway so that a reply packet from the Burrows server can find its way back to the corresponding Internet host.

4.3 Other Burrows Components

In the walkthrough, we briefly touched on the components necessary for Burrows, namely, a proximity-based service and DDNS service. In addition, we also need a component to alleviate "free-riding" and another to detect "hidden actions." While these components do not directly contribute to DoS mitigation capabilities, they are necessary for Burrows to function and they prevent Burrows from introducing new security vulnerabilities into the Internet.

4.3.1 Free-Riding

Any peer-to-peer overlay network is always faced with the problem of free-riding participants, i.e., participants who benefit from the overlay network without contributing to it. In the case of Burrows, a given end-user's server can consume more traffic than what her Burrows gateway routes. To alleviate free-riding, a traffic accounting mechanism is required. The traffic accounting mechanism is nothing more than a database that keeps track of how much traffic each Burrows server has consumed and each its corresponding gateway has routed. We can adopt existing wide-area resilient database such as a public Distributed Hash Table [27], to store traffic accounting information. In order to prevent the traffic accounting system from being tainted with bogus information, we assign public/private key pairs to all the Burrows servers for data signing and verification.

Whenever a gateway routes traffic for a server, it keeps track of the traffic. At the preset interval, the gateway will update the traffic accounting system with how much traffic it has routed and for which server it has routed the traffic.

Every time, before a gateway routes packets for a server, it checks the traffic accounting system to see if that server's corresponding gateway has contributed enough to the Burrows overlay. If not, the gateway will notify the owner of that server about the free-riding violation and discard the packets. The owner can react to the violation notice by increasing the uplink and capacity of her Burrows gateway. If there is no free-riding violation, the routing gateway routes for the destination server.

4.3.2 Hidden Actions

By routing traffic between Internet host and Burrows servers, malicious Burrows gateways can perform subtle attacks known as hidden actions, i.e., they can eavesdrop, modify and discard packets which transit through them without the communicating parties being aware of such mishandling.

The simplest way to prevent hidden actions is to use end-to-end encryption such as Secure Sockets Layer (SSL) to encrypt traffic between client and server. Without end-to-end encryption, we need to distinguish between the two traffic directions: a malicious Burrows gateway can drop (or tamper with) packets originating from Burrows to the Internet and vice versa.

For traffic exiting Burrows, the Burrows server may occasionally send "predictable reply test packets," i.e., packets that elicit a predictable response. For example, while communicating with an email server at abc.com, the test packet can be crafted to request that the email recipient is kokoro@example.com. If the reply to the test packet differs from the typical "no forwarding permitted"[3] error message, one can conclude that the Burrows gateway has dropped/modified the packet or its corresponding reply.

To examine hidden actions when packets are originating from the Internet and head towards the Burrows servers, we use the same mechanism but a predictable reply test packet has to be sent out by some designated trusted nodes.

4.3.3 Name Resolution from Outside Burrows

An Internet host that wants to initiate a connection to a Burrows server needs to resolve the Burrows server hostname to IP address of the Burrows gateway nearest to the local DNS server the Internet host is using. There are many mechanisms which can perform such proximity-based services [11, 40, 39]. While these proposals offer similar services, subtle differences impact our design choices. For example, to detect hidden actions, it is desirable for the architecture to control which Burrows gateway a so-called "tester" node can connect to send test packets. With Meridian [40, 39], each gateway maintains a distinct IP address so that the tester node may choose which Burrows gateway to test by connecting to that gateway's IP address. However, in OASIS [11], all Burrows gateways share a set of IP address range and which Burrows gateway a tester node ends up connecting to depends on the location of the tester node. For that reason, Burrows currently adopts Meridian.

4.3.4 Name Resolution Within Burrows

Since Burrows gateways proxy all connections from Internet hosts to Burrows servers, once packets from the Internet hosts reaches the Burrows gateways, they need to be forwarded by the Burrows gateways to the Burrows server. The Burrows gateways needs to perform a name resolution from hostname to the actual Burrows server IP address. Any form of name service is sufficient, but since name servers are critical to connectivity, it is best to choose a resilient name service. In our design, we opt for a DDNS [26] formed using Burrows gateways, but point out that a peer-to-peer lookup service [8] could be equally adequate.

5. FEASIBILITY STUDY

This section conducts a feasibility study of the proposed architecture against a set of well-known DoS attacks [16] and presents a back-of-the-envelope calculation on the network traffic performance degradation when the Burrows is in effect.

---

[2] The implementation details of the DDNS is not relevant to this discussion, as long as it provides the same functionality as a DNS system that serves Burrows by translating Burrows server hostname into actual Burrows server private IP address.

[3] This error message is expected since typical email servers will not receive emails for the domain which they are not configured for.

Table 1: Typical Internet Measurements

Label	Measurement	Metrics
M1	Utilizing the Meridian service to find the closest node [40, 39]	400ms
M2	Utilizing a distributed domain name service, e.g., CoDoNS [26]	105ms
M3	Round Trip-Time (RTT) improvements due to overlay routing [1, 25]	20% of RTT without Burrows

5.1 Effectiveness Analysis

We have made a preliminary evaluation on the effectiveness of Burrows against DoS attacks including UDP, ICMP echo, TCP SYN and Smurf attacks. As confirmed by [16], these attacks are the most prevalent DoS attacks seen on the Internet recently. All of these attacks can be detected as network anomalies. Therefore even a single firewall is capable of detecting and defending against such a DoS on a very small scale (500 to 22,000 SYN flood packets/sec) [23]. We can extrapolate this effectiveness to understand how much better a distributed set of Burrows gateways embedded with firewall capabilities can contend with a similar attack but on a much larger scale. The ability of Burrows gateway to drop such anomalous network traffic as near as the DoS origins also increases Burrows effectiveness against DoS attacks.

5.2 Performance Analysis

When a system in the Internet communicates with a system in Burrows, there is an additional store-and-forward point, i.e., the Burrows gateway. The delay caused may rely on the processing capability and the uplink of the Burrows gateway. However, Burrows' free-riding prevention mechanism ensures that each end-user contributes a Burrows gateway with decent performance to ensure that her Burrows server is afforded an equivalent throughput. Thus the additional latency is incurred primarily due to the establishment of an additional connection between the end-points.

Table 1 shows a few typical Internet measurements. We explain the significance of these measurements in relation to the expected increase in latency due to the introduction of the Burrows.

We consider the latency for two cases – during connection setup and after connection establishment. In our calculation, we assume the usage of the Meridian proximity-based service. During connection setup, additional latency is incurred during utilization of the Meridian [40, 39] service by the Internet host to find the closest Burrows gateway (similar to Measurement M1 in Table 1), the utilization of the DDNS by the Burrows gateway to resolve the hostname of the Burrows server into its actual private IP address within the Burrows (similar to M2 in table 1) and the round-trip-time (RTT) for packet to traverse between Burrows gateway and Burrows server (similar to M3).

There are two things to note. First, we assume that the RTT from a Burrows gateway to the Burrows server to be similar to RTT from an Internet host connecting directly to the same server when the server is not in Burrows. Note that our assumption is an over-estimation since packets from Burrows gateway are switched at layer 2 by BGP MPLS (rather than routed at layer 3) to the Burrows server. Moreover, the number of hops from Burrows gateway to the Burrows server is also probably smaller since the Burrows gateway is likely to be an intermediary hop within the direct connection between the Internet host and the destination server. Thus, summation of M1, M2 and M3 provides a very conservative estimate of the increase in latency introduced by Burrows. Secondly, we ignore most packet processing delays at the Burrows gateway including those that require public key cryptography processing such as signing traffic accounting packets and verifying them, since their magnitude is in microseconds while the above-mentioned factors are in the order of milliseconds.

After connection establishment, the increase in latency is only M3, since no name resolution is necessary.

Hence, assuming a round-trip time (RTT) of 200ms, the additional latency introduced by Burrows during connection setup is $400 + 105 + 0.8 \times 200 = 665$ ms. However, the latency incurred during connection establishment due to Meridian [40, 39] lookup happens possibly only once for the entire set of Internet hosts that shares the same local DNS. It is also likely that the DDNS lookup performed by the Burrows gateways occurs infrequently due to caching of DDNS replies. Therefore, it is highly probable that even during connection setup, for most Internet hosts, the additional latency upper-bound is just $0.8 \times 200 = 160$ ms. After connection establishment, the additional latency upper-bound is $0.8 \times 200 = 160$ ms.

Shortly stated, using fairly conservative estimates, Burrows should cause the expected RTT to increase to about 360 ms. Provided a service enjoys the benefit of DoS mitigation, a 360-ms RTT remains within the acceptable response time requirements of a vast majority of applications including interactive video/voice which can arguably accommodate round-trip time of 250 to 500 ms [35].

6. DISCUSSION

6.1 ISP Burrows

The lack of a DoS mitigation architecture with small critical mass effectiveness so far has led to infrastructure providers attempting to defer huge deployment cost to the end-users who refuse to pay such high premiums for DoS mitigation. The small critical mass effectiveness of Burrows, on the other hand, allows to circumvent that problem.

Indeed, the infrastructure provider can build Burrows with the number of Burrows gateways equivalent to the end-users' willingness-to-pay for a DoS mitigation mechanism and sell uplinks into Burrows to end-users. We call this model *ISP Burrows*.

To use an economic term, the "social welfare" of the Internet community increases with the introduction of Burrows. End-users who are willing to pay a premium for DoS mitigation can acquire DoS mitigation capability while the infrastructure providers now can earn this premium which it could not previously without Burrows. With Burrows, both infrastructure providers and end-users are now better off than they were before.

The advantage of ISP Burrows is that it can be comprised of gateways belonging to different organizations with an economic incentive to co-operate and deliver DoS mitigation mechanisms to end-users. Here again, the ability to minimize negative externality presents interesting benefits, since organizations can form federations to mitigate DoS attacks without worrying about the inaction of disinterested organizations. Moreover, if the federation of organizations are "trusted" entities, e.g., (non-competing) ISPs then the need for complex free-riding and hidden action mitigation mechanisms may be obviated thereby simplifying the deployment of Burrows.

6.2 Mitigating Other Forms of DoS Attacks

While the paper mainly focused on mitigating DoS that employs flooding techniques, we found that the introduction of gateways could also deal with other types of DoS such as TCP low-rate attacks [15]. This is made possible by the existence of detection and defense mechanisms against this attack [34]. We generalize that

any DoS attack that can be detected and defended by distributed mechanisms can be effectively deployed at Burrows gateways.

7. FUTURE WORK

In the near future, we are planning to implement and deploy Burrows to measure its actual effectiveness and performance. Another possible piece of interesting work will be to evaluate if Burrows can be modified to be "switched on" only during times of attack to address performance concerns.

We would also like to explore the possibility of using indirection and overlay routing among the Burrows servers and Burrows gateways. The use of indirection will enhance the Burrows servers with IP mobility, i.e., the ability to change IP addresses if it comes under attack directly by malicious gateways, while overlay routing will provide some anonymity, i.e., the Burrows server will be reachable only through its overlay node ID rather than IP address.

8. CONCLUSION

We postulate that the main impediment to large-scale deployment of existing DoS mitigation infrastructure lies in the misalignment of economic incentives among network participants. Our main contribution in this paper is to re-factor key DoS mitigation mechanisms to incorporate economic incentive realignment mechanisms. We identify minimizing negative externality and empowerment of end-users with the ability to protect themselves as the two key incentive realignments necessary.

We realize these realignments with a secure overlay, Burrows, which employs gateways controlling the flow of negative externality between the Internet and the Burrows servers. We also adopt a self-scaling (peer-to-peer) model to empower end-users to build Burrows without requiring aid from infrastructure providers. We show that with Burrows, we generate positive externality which drives adoption and Burrows also increases the social welfare of the Internet community.

By using an overlay network, we can deploy key DoS mitigation technologies on the gateways without requiring modifications to the existing Internet infrastructure. Instead of replicating servers, like most DoS mitigation proposals, we propose to replicate the gateways and uplinks, which has the effect of increasing reachability to servers, while making DoS mitigation transparent to existing servers.

A preliminary analysis of the soundness of the Burrows architecture indicates Burrows can hold out against a typical set of DoS attacks. Likewise, the expected degradation in performance that comes as an expense to the increased security Burrows enables appears to remain within acceptable bounds for most applications.

9. ACKNOWLEDGMENTS

10. REFERENCES

[1] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh. A Comparison of Overlay Routing and Multihoming Route Control. In *Proc. of ACM SIGCOMM*, 2002.

[2] D. Anderson. Mayday: Distributed Filtering for Internet Services. In *Proc. USENIX USITS'03*, 2003.

[3] D. Anderson, H. Balakrishnan, M. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. of ACM SOSP'01*, 2001.

[4] R. Anderson. Why Information Security is Hard - An Economic Perspective, 2001. Invited Talk for Symposium on Operating System Principles.

[5] T. Anderson, L. Peterson, S. Shenker, and J. Turner. Overcoming Barriers to Disruptive Innovation in Networking. In *Report of National Science Foundation Workshop*, Jan. 2005.

[6] CERT. Denial of Service Attacks, 1997. http://www.cert.org/tech_tips/denial_of_service.html.

[7] A. Chalitta, M. E. Hassan, S. Maalouf, and A. Zouheiry. A Survey of DDoS Defense Mechanisms. In *Research Paper from American University of Beirut*, 2004.

[8] R. Cox, A. Muthitacharoen, and R. Morris. Serving DNS using a Peer-to-Peer Lookup Service. In *Proc. of International Workshop of Peer-to-Peer Systems*, 2002.

[9] S. Dietrich, N. Long, and D. Dittrich. Analyzing distributed denial of service tools: The shaft case. In *Proc. USENIX LISA '00*, pages 329–340, New Orleans, LA, Dec. 2000.

[10] M. Freedman, E. Freudenthal, and D. Mazieres. Democratizing Content Publication with Coral. In *Proc. USENIX/ACM NSDI'04*, 2004.

[11] M. Freedman, K. Lashminarayanan, and D. Mazieres. OASIS: Anycast for Any Service. In *Proc. of ACM Symposium on NSDI*, 2006.

[12] ISP-Planet.com. Managed Security Service Providers Survey, 2006. http://www.isp-planet.com/technology/mssp/2006/mssp1a.html.

[13] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. In *Proc. of ACM STOC*, 1997.

[14] A. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure overlay services. In *Proceedings of ACM SIGCOMM'02*, pages 61–72, Pittsburgh, PA, Aug. 2002.

[15] A. Kuzmanovic and E. Knightly. Low-rate tcp-targeted denial of service attacks. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003.

[16] D. MacPherson and C. Labovitz. Worldwide Infrastructure Security Report Volume II, September 2006.

[17] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *ACM Computer Communication Review*, 32(3):62–73, 2002.

[18] J. Mirkovic and P. L. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM Computer Communication Review*, 34(2):39–53, 2004.

[19] MIT. Spoofer Project: Current State of IP Spoofing, 2007.

[20] National Research Council. Looking over the fence at networks, 2001.

[21] National Science Foundation. Future Internet design (FIND). http://www.nets-find.net/.

[22] National Science Foundation. The global environment for networking innovations (GENI) initiative. http://www.nsf.gov/cise/cns/geni/.

[23] R. Oliver. Countering SYN Flood Denial-of-Service Attacks. In *Invited Talks of USENIX Security Symposium*, 2001.

[24] Planet Lab. Planet Lab Consortium, 2002. http://www.planet-lab.org/consortium.

[25] H. Rahul, M. Kasbekar, R. Sitaraman, and A. Berger. Towards Realizing the Performance and Availability Benefits

of a Global Overlay Network. In *Proc. of Passive and Active Measurement Conference*, 2006.

[26] V. Ramasubramaniam and E. Sirer. The Design and Implementation of the Next Generation Name Service. In *Proc. of ACM SIGCOMM*, 2004.

[27] S. Rhea, B. Godfrey, B.Karp, J. Kubiatowicz, S. Ratnasamy, and S. Shenker. OpenDHT: A Public DHT Service and its Uses. In *Proc. of ACM SIGCOMM*, 2005.

[28] E. Rosen. RFC2547bis-03: BGP/MPLS IP VPNs, 2004.

[29] S. Savage and D. Wetherall. Network Support IP Traceback. In *Proc. of IEEE/ACM Transactions on Networking*, 2001.

[30] C. Shapiro and H. Varian. Information Rules: Networks and Positive Feedbacks, 1998.

[31] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, and B. Schwartz. Single Packet IP Traceback. *IEEE/ACM Transactions on Networking*, 2002.

[32] A. Stavrou, D. Cook, W. Morein, A. Keromytis, V. Misra, and D. Rubenstein. WebSOS: An Overlay-based system for protecting web servers against DoS. In *Elsevier Journal of Computer Networks*, 2005.

[33] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *Proc. of ACM SIGCOMM*, 2002.

[34] H. Sun, J. C. S. Lui, and D. K. Y. Yau. Distributed mechanism in detecting and defending against the low-rate tcp attack. *Computer Networks*, 50(13):2312–2330, 2006.

[35] Sun Microsystems. Enterprise Quality of Service (QoS) from Prentice Hall Technical Reference - http://www.phptr.com/articles/article.asp?p=26023&rl=1, 2002.

[36] L. von Ahm, M. Blum, N. Hooper, and J. Langford. CAPTCHAS: Using Hard AI Problems for Security. In *EuroCrypt*, 2004.

[37] M. Walfish, H. Balakrishnan, D. Karger, and S. Shenker. DoS: Fighting Fire with Fire. In *Proc. of HotNets*, 2005.

[38] L. Wang, K. Park, R. Pang, V. S. Pai, and L. Peterson. Reliability and Security in the CoDeeN Content Distribution Network. In *Proc. of USENIX 2004 Annual Technical Conference*, 2004.

[39] B. Wong and E. Sirer. ClosestNode.com: An Open-Access, Scalable, Shared Geocast Service. In *Proc. of SIGOPS Operating System Review*, 2006.

[40] B. Wong, A. Slivkins, and E. Sirer. Meridian: A Lightweight Framework for Network Location without Virtual Coordinates. In *Proc. of ACM SIGCOMM*, 2005.

[41] A. Yaar, A. Perrig, and D. Song. FIT: Fast Internet traceback. In *Proceedings of IEEE Infocom*, Mar. 2005.

[42] J. Yan, S. Early, and R. Anderson. The XenoService - A Distributed Defeat for Distributed Denial of Service. In *Proc. of ISW 2000*, Oct. 2000.