STRIDE: Sanctuary Trail – Refuge from Internet DDoS Entrapment

Hsu-Chun Hsiao* Tiffany Hyun-Jin Kim* Sangjae Yoo* Xin Zhang†
Soo Bum Lee* Virgil Gligor* Adrian Perrig*

*CyLab / Carnegie Mellon University †Google

{hchsiao, hyunjin, yoo, soobum, gligor, perrig}@cmu.edu xinzh@google.com

ABSTRACT

We propose STRIDE, a new DDoS-resilient Internet architecture that isolates attack traffic through viable bandwidth allocation, preventing a botnet from crowding out legitimate flows. This new architecture presents several novel concepts including tree-based bandwidth allocation and long-term static paths with guaranteed bandwidth. In concert, these mechanisms provide domain-based bandwidth guarantees within a trust domain – administrative domains grouped within a legal jurisdiction with enforceable accountability; each administrative domain in the trust domain can then internally split such guarantees among its endhosts to provide (1) connection establishment with high probability, and (2) precise bandwidth guarantees for established flows, regardless of the size or distribution of the botnet outside the source and the destination domains. Moreover, STRIDE maintains no per-flow state on backbone routers and requires no key establishment across administrative domains. We demonstrate that STRIDE achieves these DDoS defense properties through formal analysis and simulation. We also show that STRIDE mitigates emerging DDoS threats such as Denial-of-Capability (DoC) [6] and N^2 attacks [22] based on these properties that none of the existing DDoS defense mechanisms can achieve.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

Keywords

DDoS defense, DDoS-resilient Internet architecture, bandwidth allocation, bandwidth guarantees.

1. INTRODUCTION

DDoS attacks are still prevalent in the Internet today. In fact, a recent world-wide security survey [1] suggests that Botnet-driven DDoS attacks have become common as a low cost, high-profile form of cyber-protest. Both attack intensity and frequency have drastically accelerated: the largest reported attack size doubled every year, to more than 100 Gbps seen in 2010. The majority of network operators in the survey also ranked DDoS attacks as the biggest threat.

The recently proposed N^2 attack [22], also referred to as a Coremelt attack, poses a new threat and has not been effectively addressed by any system to date. In an N^2 attack, an adversary uses a large-scale botnet whose bots communicate only with each other to overload network links. Current DDoS defense mechanisms that attempt to eliminate undesired traffic are rendered ineffective, because all inter-bot traffic is desired by the bot endhosts. Other DDoS defense mechanisms that perform per-source or per-computation fair sharing at congested links may in fact give disproportionate advantage to sources with small uplink bandwidth or with high computational resources, respectively. Moreover, malicious domains can misuse per-source fair sharing by creating multiple bogus senders, and per-computation fair sharing may be too expensive to protect every data packet. Furthermore, global fair sharing implies global fate sharing – a source's share is affected by bots in distant domains over which the source has no influence.

Current DDoS countermeasures have encountered fundamental limitations to address the challenges we describe above to be compatible with the current Internet. Thus, an exciting research challenge is to study if a next-generation network infrastructure could be more effective against DDoS attacks – what architectural primitives can effectively defend against DDoS attacks?

In this paper, we formulate a new network architecture called STRIDE that provides *domain-based guarantees* for intrinsic DDoS protection within a Trust Domain (TD), which contains a set of contiguous Autonomous Domains (AD) with a common root of trust. Specifically, STRIDE provides precise bandwidth guarantees to AD-level paths, or the "sanctuary trails" that isolate attack traffic from legitimate communication. Each endpoint AD can then internally split the guarantee among its endhosts.

Our architecture is based on the following insights: (1) Bandwidth allocation is simple in a tree-based topology, as the available bandwidth can be split from the root down to each leaf; (2) with network capabilities encoded in packet-carried state and fixed bandwidth classes, routers can perform enforcement in a per-flow stateless fashion using probabilistic detection of the largest flows; (3) by combining a static long-term traffic class guaranteeing low bandwidth with a dynamically-allocated short-term traffic class guaranteeing high bandwidth, we can provide a variety of

ASIACCS'13, May 8–10, 2013, Hangzhou, China.

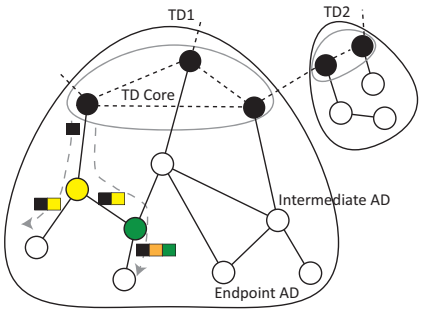

Figure 1: Trust domain (TD) example. Each node represents an AD, and the five black nodes represent the tier-1 ADs that constitute the TDCs. Each square corresponds to the node's path information.

guarantees to both privately communicating endhosts and public servers.

We leverage the SCION next-generation Internet architecture [31] to perform the tree-based bandwidth allocation: paths are created and available bandwidth is allocated as paths branch out like trees from the network core. The packet-carried forwarding state of SCION also provides us with a natural way to encode network capabilities [4, 26].

Note that existing schemes for bandwidth reservation (e.g., RSVP [8]) or Quality of Service are insufficient to guarantee timely end-to-end data delivery in the presence of DDoS adversaries for several reasons: (1) their reservation requests are unprotected against DDoS attacks, and (2) they lack lower-bound guarantees for reservable bandwidth. In contrast, STRIDE provides domain-based guarantees, achieving previously unachievable DDoS defense properties for communication within a TD; many of the properties also translate for communication between TDs.

STRIDE provably guarantees connection setup for private communication, bounds the waiting time for accessing public services, and provides precise bandwidth guarantees for established flows, all achieved regardless of the size or distribution of the botnet outside the source and destination ADs. These guarantees enable STRIDE to mitigate emerging threats such as the Denial-of-Capability (DoC) and N^2 attacks. Furthermore, STRIDE does not require backbone routers to keep per-flow state.

2. BACKGROUND: SCION

In this section, we review the SCION network architecture [31] that STRIDE leverages for DDoS resilience. Among new Internet architecture proposals [27, 31], we base our design on SCION because its notion of Trust Domain (TD) and secure top-down topology discovery enable tree-based bandwidth allocation within a uniform legal environment, whereby bandwidth guarantees and attack isolation can be enforced. Note that although SCION enables natural isolation of attack traffic from untrusted entities, it is still vulnerable to DDoS threats within a TD; in contrast, STRIDE seeks to provide domain-based bandwidth guarantees for intra-TD communication.

We consider an Internet topology at the Autonomous Domain (AD) level. In this topology, nodes represent ADs, each of which has several gateway routers (or interfaces) connecting it to neighboring ADs. Endpoint ADs provide Internet access to endhosts.

2.1 Trust Domains

SCION divides ADs on the Internet into several trust domains (TDs), where a TD is defined as "a set of ADs that agree on a coherent root of trust and have mutual accountability and enforceability for route computation under a common regulatory framework" [31]. Each TD contains a *TD Core* (TDC) consisting of the tier-1 ISPs that manage the TD. The primary advantage of such a trust domain division is that it avoids having a single root of trust for the entire Internet, which is difficult to unanimously agree on in practice. For ease of presentation, we will focus on operations within one TD unless we mention otherwise explicitly. Figure 1 illustrates these concepts.

2.2 Secure Top-Down Topology Discovery

For finding routing paths in SCION, the TDC periodically broadcasts *Path Construction Beacons* (PCB), which establish half-paths back to the TDC as they are disseminated throughout the network in a top-down manner (i.e., from the TDC to endpoint ADs). End-to-end communication paths are established by combining the source's and destination's half-paths.

PCBs are constructed as follows. The TDC initiates PCBs which contain *one-hop* paths starting from the core to its adjacent customer ADs with their expiration times. Upon receiving a PCB, an intermediate AD updates the PCB for each of its downstream ADs (e.g., customers and peers) with the authenticated local topology: the i^{th} intermediate AD (AD_i) appends to the PCB the local path information, ingress and egress interfaces I_i, for a particular downstream AD (AD_{i+1}) followed by an opaque field O_i, which encodes the forwarding decision as ingress/egress points at AD_i.

$$\begin{aligned} I_i &= ingress_i \| egress_i \| AD_{i+1}, \\ O_i &= \mathrm{MAC}_{K_i}(I_i \| O_{i-1}), \end{aligned} \tag{1}$$

where $ingress_i$ and $egress_i$ stand for the ingress and egress interfaces of AD_i. O_i is computed using a secret key K_i known only to AD_i to protect the integrity of the routing information. Also, AD_i digitally signs PCBs to prevent fake route injection. Note that PCBs in SCION do not announce bandwidth availability.

AD_i propagates the updated PCB to the designated downstream AD (AD_{i+1}). AD_i repeats this process for other downstream ADs on different paths, and upon receiving PCBs, the downstream ADs learn the path to reach the TDC. PCBs travel along a special control channel which has isolated bandwidth from all data packets and hence is protected from data-plane DDoS attacks.

For each received PCB, an endpoint AD learns a series of interfaces and opaque fields as an unforgeable *path token* that represents the forwarding decisions of the corresponding path. To send packets on the path, the sender embeds in the packet header the path token, which reminds every intermediate AD of its own routing decision for carrying the packet based on its policy. Hence, no forwarding state at routers is needed.

Among all the half-paths that an endpoint AD learns from PCBs, the endpoint AD selects some as up-paths for reaching the TDC and some as down-paths for receiving packets from the TDC. To form end-to-end paths, the destination AD publishes its down-paths (i.e., the path tokens of these

down-paths) to the Path Server, which is a DNS-like system, in the TDC. A source AD wishing to communicate with the destination can query the Path Server for the destination's down-paths. An end-to-end path is then constructed by splicing the source's and destination's half-paths.

3. THREAT MODEL

We consider massive Distributed Denial of Service (DDoS) attacks launched by a botnet, which consists of a large number of malware-infected bot endhost machines. In particular, we address two types of DDoS attacks: (1) disabling connection setup in capability-based protocols (DoC attack [6]), and (2) exhausting link bandwidth to crowd out established legitimate connections. In bandwidth exhaustion attacks, we especially focus on the N^2 attack [22], which aims to overload a target ISP's backbone network using a large number of legitimate-looking flows established among colluding bots (hence, any attempt to identify the attack based on the flow's bandwidth would fail). This attack is called an N^2 attack, because the N bots can open $O(N^2)$ connections among each other.

3.1 Desired Properties

We aim to achieve the following properties for a DDoS-resilient network architecture.

Domain-based guarantees within a TD. Precise bandwidth guarantees should be provided to communication between endpoint domains residing in the same TD, and each endpoint domain can internally split the guarantee among its endhosts based on its local policy. A domain that intends to achieve highly available communication could identify malicious bots and remove them to provide better guarantees to legitimate endhosts. Domain-based guarantees ensure that the effect of attacks is confined to infested domains, such that the endhosts can establish a bandwidth-guaranteed flow with high probability.

Robustness and efficiency. To be resilient to DDoS attacks, network elements require efficient protocols and network devices. indicates that the architecture should avoid per-flow or per-host state at backbone routers and should avoid expensive operations such as digital signature generation or verification in the fastpath.[1]

Flexible route control. Endpoint ADs should be able to control paths to avoid congestion: a source AD needs to have multiple paths to reach a destination; a destination AD should be able to hide/disclose paths for private/public communication, and change inbound paths to shift traffic.

3.2 Assumptions

In designing a new DDoS resilient Internet architecture, we only make two fundamental assumptions that can be justified using existing security mechanisms.

TDCs support congestion-free communication. Since the TDC topology is small and relatively fixed, ADs in the TDC can accurately assess and provision the capacity requirement of each link to ensure no packet loss for traffic below a certain rate. For example, given a topology and its link capacity, the TDC can adopt congestion-free routing [30] to determine how much congestion-free traffic it can support on each incoming link.

TDC detects and revokes malicious AD members. Since all ADs of a TD are within a uniform legal environment (as described in Section 2), the TDC can revoke the membership of misbehaving ADs. In Section 8, we discuss technical approaches to detect compromised or poorly-administered ADs that fail to correctly monitor traffic.

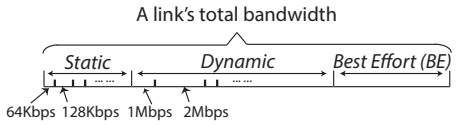

Figure 2: Three bandwidth classes of STRIDE.

4. STRIDE: DESIGN OVERVIEW

We first sketch how our new architecture, STRIDE, provides guaranteed end-to-end data delivery to legitimate flows even in the presence of DDoS attacks. In a nutshell, STRIDE protects end-to-end data delivery by establishing communication *channels*[2] that confine the effect of attacks to their originating domains, leaving other domains unaffected. Such communication channels also form the "sanctuary trails" that isolate attack traffic from legitimate communication. Moreover, since bots within a domain will compete among each other for a fixed amount of bandwidth, bandwidth guarantees can be provided to channels regardless of the size or distribution of the botnet outside the source and destination domains.

Hence, the primary challenge STRIDE faces are how to secure such channel establishment and adjust allocation in response to dynamic traffic patterns. STRIDE addresses these challenges by combining (1) a low bandwidth but long-lived and guaranteed traffic class for channel establishment with (2) a high bandwidth but short-term dynamically-allocated traffic class. Specifically, a channel is constructed using any of the 3 types of *bandwidth classes*: static, dynamic, and best effort (BE). Hybrid channels are also possible, as we explain later. For simplicity, we focus on presenting STRIDE using bidirectional links and elaborate how STRIDE can support directional links in Section 8.

Bandwidth classes. As shown in Figure 2, link bandwidth is split up into three bandwidth classes:

- *Static class* is for guaranteed, persistent long-term bandwidth that ADs allocate, for example to protect initial connection setup request packets between a source and a destination. Each AD allocates a small portion of its total bandwidth (e.g., 5–15%) to this class.
- *Dynamic class* is for guaranteed, short-term end-to-end bandwidth allocations, and supports high-capacity channels. The dynamic class may account for the majority of the link capacity (e.g., 60–65%).
- *Best-effort (BE) class* is allocated with the remainder of the bandwidth (e.g., 30%).

In case of congestion within a bandwidth class, it can take over the unused bandwidth from the other (uncongested) classes using statistical multiplexing.

[1] Fastpath and slowpath refer to different processing elements that handle packet forwarding in a router. The fastpath refers to packet processing by dedicated hardware, for example, by the linecard in a router. The slowpath refers to packet processing by a general CPU, which often results in a packet processing latency that is usually about two orders of magnitude slower than the fastpath.

[2] The bandwidth of a *path* is divided into separate *channels*, and multiple channels dynamically share the bandwidth within a path.

Within the static or dynamic bandwidth class, an AD can assign different bandwidth sub-classes (e.g., 500 Kbps, 1 Mbps, etc.) to individual paths/flows based on empirically measured flow size distribution. For example, the fraction of dynamic bandwidth allocated to the 1 Mbps sub-class can be derived based on the fraction of flows with 1 Mbps rate in the current Internet. We provide guidelines on how an AD can divide its total link capacity to the above three classes in Appendix A.

A half-path announced by a PCB is a path on the BE class, offering no guarantees. However, for each upstream AD, an endpoint AD can *activate* up to k BE half-paths to convert each into a static half-path, offering a guaranteed amount of bandwidth from that AD to the TDC. The parameter k is determined by contract and is enforced by provider ADs.

Static and BE channels. A communication channel is a conduit to carry traffic from a source to a destination. A single half-path can be used as a channel to reach the TDC, and two half-paths can be combined to form an end-to-end channel between a source and a destination. In this paper, we call the half-path of the source the up-path (as traffic on that channel traverses ADs upwards towards the TDC) and the half-path of the destination the down-path. If two BE half-paths (i.e., half-paths using the BE bandwidth class) are combined, the resulting channel is a BE channel, and similarly, two static half-paths create a static channel. Hybrid channels are also possible, and we will make use of a static up-path that is combined with a BE down-path. Packets flowing through different types of channels have different properties.

Dynamic channel. An endhost can send a request on static, BE, or hybrid channels to reserve a dynamic channel, which provides high bandwidth for a short amount of time (on the order of seconds to enable fast revocation). All ADs, including the destination, need to agree on the amount of bandwidth offered for the dynamic channel. This channel is similar to network capabilities [4, 26] with bandwidth guarantees [29].

One surprising aspect of STRIDE is that only the access routers in the endpoint ADs have to keep per-flow state for flow admission and policing. STRIDE does not require intermediate routers to maintain any per-flow or per-path state in the fastpath for forwarded traffic; only the initial channel establishment requests require slowpath operations for admission control. Performing per-flow management at edge routers is shown to be practical [21], and we can further relax this state requirement using probabilistic detection of the largest flows. With these ingredients, we construct a series of mechanisms that achieve the highly available communication we strive for.

4.1 Static Half-Path Setup

We now describe how an endpoint AD (AD_S) establishes a guaranteed static path (i.e., a path using static bandwidth class) to its TDC.

① **Bandwidth announcement:** The TDC assesses its link capacity and adds information regarding current reservable static bandwidth to periodic PCBs (TDC and PCBs are explained in Section 2). The TDC ensures no congestion on its internal links even if the announced static bandwidth becomes fully reserved. As a PCB travels from the TDC to endpoint ADs, each AD adds information regarding its own bandwidth availability. In particular, an intermediate AD splits the reservable bandwidth and announces the split amount for each of its children for the static path. Figure 3 depicts the bandwidth availability announcement (denoted by the numbers) of PCBs for the static path. This particular diagram shows that AD_S has three possible static paths:

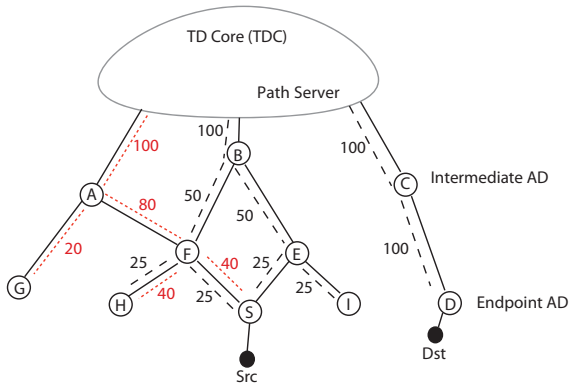

Figure 3: Illustration of STRIDE and how bandwidth is split and announced through PCBs.

❶ $TDC \overset{100}{\rightarrow} AD_A \overset{80}{\rightarrow} AD_F \overset{40}{\rightarrow} AD_S$
❷ $TDC \overset{100}{\rightarrow} AD_B \overset{50}{\rightarrow} AD_F \overset{25}{\rightarrow} AD_S$
❸ $TDC \overset{100}{\rightarrow} AD_B \overset{50}{\rightarrow} AD_E \overset{25}{\rightarrow} AD_S$

② **Activation:** Recall that each endpoint AD learns a set of BE half-paths to the TDC through PCB propagation. The AD can then activate k BE half-paths per provider AD to static class as follows. Upon receiving a PCB with the bandwidth availability information, AD_S sends an activation request to the TDC to reserve the static bandwidth as specified in the PCBs. While forwarding this request, all the intermediate ADs temporarily reserve the requested static bandwidth for AD_S if they can support the request. For example, AD_S may request 40 units of the static-class bandwidth through path ❶, 25 units through path ❷, etc.

③ **Confirmation:** The TDC sends a confirmation to AD_S for the guaranteed static path that AD_S (and all of its hosts) can use to reach the TDC. The new opaque fields constructed along the confirmation acts as a *static path token* that enables AD_S to communicate with the TDC on the static path. Note that STRIDE allows each endpoint AD to create up to k static half-paths per provider AD, and each endpoint AD has the freedom to keep a subset of static half-paths in private for privileged access, as we will explain in step ④ below, while registering others at the Path Server for public usage.

4.2 Static and BE Channel Setup

After the half-path setup, each endpoint AD obtains a set of static bandwidth-guaranteed half-paths in addition to a set of BE half-paths provided by the original PCBs. We now describe how two half-paths (i.e., an up-path and a down-path) can be combined to setup an end-to-end channel. Combining static and BE half-paths results in four types of channels (i.e., static, BE, static+BE, or BE+static) with different guarantees, as summarized in Table 1.

④ **End-to-end path selection:** When source *src* of AD_S wants to communicate with destination *dst* of AD_D, *src* queries the Path Server for the down-paths to reach *dst*. *Src* reaches the Path Server in the TDC using a BE or static half-

path as a communication channel. The Path Server then returns unconcealed static down-paths and/or BE down-paths to *src*. Alternatively, *dst* can inform *src* of a private static path over an Out-Of-Band (OOB) channel. By combining one of its up-paths and one of the down-paths provided by the Path Server, *src* now establishes an end-to-end channel for sending a dynamic channel setup request.

4.3 Dynamic Channel Setup

⑤ **Dynamic channel setup request:** To acquire guaranteed bandwidth along a dynamic channel, *src* sends a dynamic channel setup request on this newly-established end-to-end (static, BE, or hybrid) channel. While this request travels toward *dst*, all ADs on the path specify the bandwidth that they can provide for the dynamic channel and forward it toward *dst*.

In case *src* sends packets beyond the allocated static bandwidth of the static up-path, AD_S sets an overuse bit on each extra packet to utilize unused static or BE bandwidth, thereby indicating that the extra packets are beyond permitted allocation for efficient traffic policing.

In case *src* cannot send the request on the static channel, possibly due to congestion on any of the announced down-paths, STRIDE flexibly allows the endpoint AD to send requests on the BE channel. We discuss several alternatives for channel composition and their priorities in Section 5.3.

⑥ **Dynamic-class bandwidth allocation:** When *dst* receives *src*'s dynamic channel setup request, *dst* can deduce sustainable dynamic-class bandwidth for *src* based on the reported dynamic-class bandwidth availability of all the intermediate ADs (e.g., minimum of the dynamic-class bandwidth allocations of all the intermediate ADs). Then *dst* sends this information to *src*, during which the *dynamic capability* is constructed on the return path. STRIDE provides flexible options for sending the reply. For example, *dst* may send the reply through the allocated dynamic channel or through the reverse channel that sent the request.

⑦ **Guaranteed data transmission:** When *src* receives the reply, it can enjoy sending data traffic using the dedicated dynamic-class bandwidth by embedding the dynamic capability in the data packets. Since this bandwidth is short-lived, *src* may renew this dynamic-class bandwidth using actual data packets. Similar to step ⑤, *src* can also send more than permitted dynamic bandwidth allocation, in which case AD_S sets the overuse bit for the extra data traffic.

5. STRIDE PROTOCOL DESCRIPTION

In this section, we elaborate on STRIDE's mechanisms.

5.1 Static Half-Path Setup

① **Bandwidth announcement:** Upon receiving a PCB, an intermediate AD (AD_i) forwards it to the downstream AD (AD_{i+1}) after appending the bandwidth information, which includes: (1) currently reservable static-class bandwidth for AD_{i+1}, (2) currently underutilized static-class bandwidth that AD_{i+1} can use, (3) currently available dynamic-class bandwidth, and (4) currently available BE-class bandwidth. This bandwidth information enables downstream ADs and endhosts to deduce congestion status and make informed decisions in selecting paths. When a PCB reaches an endpoint AD, it contains a path from the TDC to the endpoint AD with reservable bandwidth that can be provided once this path is activated (details in step ②).

On each PCB, AD_i adds an opaque field O_i, as described in Eq. (1). The resulting collection of opaque fields in the PCB access the BE channel on the route that the PCB traversed. Note that AD_i should use different MAC keys to construct opaque fields for different bandwidth classes and expiration time, so that an attacker cannot forge an opaque field for another traffic class or extend the expiration time. For example, the MAC key K_i can be derived from the master secret key $\hat{K}_i$: $K_i = F_{\hat{K}_i}(BE, timestamp)$, where $F(\cdot)$ is a pseudo-random function.

Path diversity vs. quality. The bandwidth announcement mechanism in STRIDE enables an AD to divide bandwidth among its customers. However, bandwidth allocation is still challenging because of the tradeoff between path diversity (i.e., the number of different paths) and path quality (i.e., the allocated bandwidth to each path). An intermediate AD can offer higher path diversity by propagating a PCB to more children (i.e., egress routers). Yet, increasing path diversity reduces the bandwidth that can be allocated to each child AD since the bandwidth contained in a PCB must be split for all children *recursively* as the PCB propagates to downstream ADs.

To address this issue, STRIDE uses a *bandwidth overbooking* technique to enhance path diversity and quality simultaneously as follows. An endpoint AD can choose up to k paths out of all announced paths. This indicates that not all announced paths will be activated. Hence, intermediate ADs can announce greater reservable bandwidth (i.e., overbook) to their downstream ADs than the actual link capacity, and defer the actual bandwidth reservation later during the activation step (②). However, if intermediate ADs overbook their bandwidth aggressively, path activation could be frequently denied. To address this issue, STRIDE allows each intermediate AD to overbook its bandwidth such that the probability of path-activation failure along its link is below a certain threshold, which we analyze in Appendix B.

From the bandwidth announcement in the latest PCB, an endpoint AD learns the amount of static bandwidth it may reserve on the corresponding route. Note that the actual allocation is not performed until the activation step (②), as the reservable bandwidth may be greater than what an intermediate AD can support because of overbooking or stale bandwidth information.

② **Activation:** An endpoint AD requests for a path activation along the reverse path to the TDC. Each request consists of desired (1) expiration time of the path, and (2) amount of static-class bandwidth, which does not exceed the announced reservable bandwidth in the latest PCB. STRIDE considers an activation request as a control message, like PCBs, that is protected from data-plane DDoS attacks. To avoid congestion on the control plane, each AD can rate-limit the activation requests on a per customer basis and advertise the limit with the reservable bandwidth during the announcement.

Upon receiving an activation request, an intermediate AD, which has sufficient unallocated bandwidth (i.e., spare capacity $\geq$ desired bandwidth), temporarily allocates the requested bandwidth for this path. Otherwise, the AD sends back an error message. Also, to minimize bandwidth waste, the AD recycles temporarily allocated bandwidth when the activation fails, which is indicated by the lack of a confirmation (step ③) or an error message, before the arrival of the next PCB.

For efficient bandwidth management (e.g., allocation and recycling), the expiration time and bandwidth is chosen from a pre-defined finite set of values. For example, the expiration time can be 6, 12, 18, or 24 hours; the sub-class bandwidth assigned to each activation request can be 64 Kbps, 128 Kbps, etc.

Each endpoint AD is allowed to activate up to k distinct paths per upstream AD (or provider). This policy is made in accordance with the observation that in the current Internet, large endpoint ADs often subscribe to multiple providers for increased capacity and path diversity. This k-path policy can be enforced either by the providers or the Path Server in the TDC.

③ **Confirmation:** The TDC informs the endpoint AD of a successful static path activation by sending a confirmation message along the activated path. The confirmation message contains the expiration time and the allocated bandwidth. The TDC also updates the Path Server to include this activated path. Upon receiving the valid confirmation from the TDC, each intermediate AD on the path converts the temporarily-allocated bandwidth to be long term (until the expiration time).

Before forwarding the confirmation to the next hop, the AD adds a new opaque field using a different MAC key K_i, derived from the master secret key $\hat{K}_i$ similarly as before: $K_i = F_{\hat{K}_i}(static, timestamp, BW)$, where BW is the amount of bandwidth allocated to the path. After receiving the confirmation, the endpoint AD can forward packets on the static channel of the path by including a *static path token*, which consists of the new opaque fields, in the header of packets.

5.2 Static and BE Channel Setup

After the half-path setup, endpoint ADs learn multiple long-term, bandwidth-guaranteed static half-paths (in addition to multiple BE half-paths) to communicate with the TDC.

④ **End-to-end path selection:** When an endhost *src* in AD_S attempts to make a connection to another endhost *dst* in AD_D, *src* contacts AD_S for path resolution. In turn, AD_S requests the Path Server in the TDC for a list of static paths to *dst*, and the server returns down-paths. As a result, the AD_S can select an up-path (from itself to its TDC) and a down-path (from the TDC to AD_D), and splice them to form an end-to-end path.

Path Server availability. A successful DDoS attack against Path Servers would disable end-to-end path establishment in STRIDE. Such an attack can be prevented in two ways: (1) the TDC can detect the origin of attack traffic against Path Servers and throttle their traffic, and (2) the bandwidth-guaranteed static paths can be used to contact a Path Server.

Path selection policy. If AD_S keeps on selecting paths based on the highest available bandwidth, it may end up selecting a single best path for all the source endhosts (besides *src*), eventually congesting this path. To resolve this issue, endpoint ADs in STRIDE perform probabilistic path selection as follows: the endpoint AD selects a path with probability proportional to the path bandwidth guarantees. With this policy, the endpoint ADs are more likely to select uncongested paths and reduce the average number of trials. We evaluate a specific instance of this policy in Section 7.

Private paths. We introduce the notion of private paths, which endpoint ADs can use to provide guaranteed down-paths to preferred endhosts. In a nutshell, an endpoint AD keeps a subset of half-paths as private and provides them to its destination endhosts such that they can selectively provide them to preferred sources. We define private services to be provided by those servers that can predict future customers (e.g., premium customers on Amazon). A private server providing access to a closed community can provide guaranteed connection setup to community members with private down-paths as follows: a destination can selectively disclose its private down-paths to preferred sources. The private paths can be distributed via OOB channels or by uploading encrypted private paths to a Path Server. As a result, a valued customer of Amazon, for example, can obtain a bandwidth-guaranteed static down-paths for sending dynamic channel setup requests to Amazon.

Table 1: Guarantees of dynamic channel setup delay and bandwidth for different types of end-to-end channels.

Up-path	Down-path	Delay	Bandwidth guaranteed?
Static	Static (private)	Constant	✓
Static	Static (public)	Linear	✓
Static	Best-effort	Linear	✗

5.3 Dynamic Channel Setup

Using a (BE, static, or hybrid) channel, *src* sends dynamic channel setup requests to establish an end-to-end dynamic channel. With such an end-to-end dynamic channel, STRIDE can provide bandwidth guarantees to short-term, high-bandwidth dynamic flows. Note that *src* can send any types of packets on the end-to-end channel, but we focus on the discussion of sending dynamic channel setup requests, as it is a part of our DDoS defense mechanism.

⑤ **Dynamic-channel setup request:** After selecting an end-to-end channel in step ④, a source endhost can send a dynamic channel setup request for guaranteed dynamic-bandwidth allocation. Table 1 describes guarantees of dynamic channel setup delay and bandwidth for different types of end-to-end channels.

A request header carries two additional indicators that enable congested intermediate ADs to efficiently control link bandwidth:

- *Overuse bit:* A source AD sets an overuse bit of a packet on a static up-path in case its endhost is sending packets more than the reserved static-class bandwidth of the up-path.
- *Congestion bit:* Any AD that experiences link congestion sets a congestion bit in BE packets.

Traffic priority of requests. When an AD receives more packets than what its outgoing links can afford, the AD has to discard some of them while maintaining the static-class bandwidth guarantee. Based on where the congestion occurs, we discuss different techniques to prioritize packets.

- *Host contention at source ADs:* Static bandwidth contention may occur on the source AD's outgoing links. Each AD can have a different way to resolve contention. For example, it can adopt a payment-based scheme: each client informs its host AD how much it is willing to pay for using a static up-path, and the endpoint AD can arrange based on some objectives (e.g., maximize the AD's revenue) [23]. For ease of analysis, we consider per-host fair share within an endpoint AD.

Table 2: Traffic priority of dynamic channel setup requests on the down-paths that experience link congestion.

Priority	Up-path	Bits set	How requests arrived
1	Static	-	Within allocated static BW
2	BE	-	On uncongested BE link
3	Static	Overuse	Beyond allocated static BW
4	BE	Congest	On congested BE link
5	Outside TD	-	From outside TD

- *Link congestion on up-paths:* During link congestion, each source domain obtains *a weighted fair share* of the available (unallocated, or allocated but unused) static bandwidth. A weighted fair share is proportional to the source domain's static allocation on the congested link. Hence, static packets with the overuse bit would be transmitted on the static channel up to the weighted fair share, and packets beyond the share are converted into BE packets to compete with the standard BE traffic.
- *Link congestion on down-paths:* STRIDE assigns priority levels to the packets such that low priority packets are dropped first in the case of congestion. Table 2 summarizes the priority levels ordered from the highest to the lowest.

This priority applies to both static and BE classes. If the congestion persists within the first-priority traffic, the congested link assigns to it a weighted fair share of the available static-class bandwidth proportional to each *destination domain*'s static allocation.

Determine reservable dynamic bandwidth. Recall that the dynamic bandwidth class has defined bandwidth sub-classes, such as 512 Kbps, 1 Mbps, 2 Mbps, etc., and each dynamic channel is associated with a given sub-class. Intuitively, to provide precise bandwidth guarantees for established flows, we have to ensure that every STRIDE-protected request packet can be offered sufficient dynamic bandwidth (e.g., at least 512 Kbps). Ideally, we would like to provide guaranteed flow bandwidth to every request packet traversing static channels, as Table 1 shows.

One key challenge here is how to flexibly determine the amount of reservable dynamic bandwidth for such requests. To address this challenge, STRIDE limits the rate of the dynamic channel setup requests (thus the dynamic allocation) within the static class to be *proportional to the static allocation*. For example, if each dynamic channel is guaranteed 10 units per second and expires in 2 seconds, and the rate limit is 3 requests per second, then the dynamic-class link bandwidth should be greater than $10 \cdot 2 \cdot 3$ units per second to accommodate the worst case where all requests arrive using the static class.

The AD assigns the smallest sub-class to the initial request and flexibly upgrades to a higher sub-class for the subsequent requests for the allocation renewal if the link is not congested. In the case of link congestion on the up-path (down-path), each source (destination) domain obtains a weighted fair share of the available dynamic-class bandwidth that is proportional to the source's (destination's) static allocation on the congested link.

Each AD on the path (including the destination AD) either approves the requested dynamic bandwidth, or indicates the maximum available bandwidth (which is at most the available bandwidth indicated by the previous AD).

⑥ **Dynamic-class bandwidth allocation:** Through a dynamic-channel setup request, a destination endhost can discover the bottleneck link(s) and the available dynamic-class bandwidth along the path. The destination constructs a reply packet, which carries (1) reserved dynamic-class bandwidth of this flow, (2) opaque fields (which include a *dynamic flow capability*), and (3) expiration time which indicates the lifetime of the guaranteed dynamic bandwidth. The destination also indicates which AD-to-AD link(s) is the bottleneck for determining the bandwidth reservation.

As the packet travels back to the source, ADs update their dynamic bandwidth allocation and opaque fields to accurately reflect the available bandwidth and reduce the potential waste of bandwidth. If the allocated end-to-end dynamic bandwidth does not meet the source's need (e.g., determined by an application service), the source may select an alternative path. Furthermore, the source can make an informed decision to avoid the bottleneck link when selecting an alternative path.

Capability update. A source can renew the short-term dynamic capability while communicating with the destination as follows: the sender sets a renewal bit in the header of the capability-protected dynamic-class packets. If the destination renews, the source AD invalidates the old dynamic capability (e.g., by keeping track of the latest capability for each flow and rejecting packets carrying old capabilities) to prevent misuse.

⑦ **Guaranteed data transmission:** Upon receiving a dynamic capability (step ⑥), *src* can use the end-to-end dynamic channel for guaranteed data transmission. *Src* can also flexibly choose other types of end-to-end channels for different guarantees.

Regulation. For per-flow bandwidth guarantees, endpoint ADs monitor per-flow data usage and regulate potential violation. For example, every endpoint AD ensures that the overuse bit is set in data packets whose flow rate exceeds the allocated value. ADs are responsible to drop some of the data packets with the overuse bit to resolve link congestion. For example, similar to the static channel regulation, the AD can drop packets that are beyond the weighted fair share of the source or the destination. In addition, intermediate ADs and the TDC can perform both real-time probabilistic monitoring and offline traffic analysis to identify misbehaving endpoint ADs that fail to regulate their clients. TDCs and ADs also monitor per-TD bandwidth usage of dynamic-class traffic at each interface at the TD boundary to isolate attack traffic from other TDs.

6. BANDWIDTH GUARANTEE ANALYSIS

We first show that STRIDE achieves domain-based guarantees for communication between the source (AD_{src}) and the destination (AD_{dst}) domains within a TD; specifically, we analyze what domain-based guarantees AD_{src} can obtain using different types of channels. We then discuss how an endpoint AD can divide such domain-based guarantees among its endhosts.

In Theorem 1, we show that by leveraging private down-paths, AD_{src} and AD_{dst} can establish bandwidth-guaranteed static channels for congestion-free communication. Let u_i and d_i be AD_i's total static up-path and down-path bandwidth allocations, respectively, where $1 \leq i \leq m$ and m is the number of ADs. Since each AD is expected to assess its bandwidth requirement of static half-paths based on its contractual agreements with human subscribers, u_i and d_i are constant irrespective of the number of ADs or the power of the botnet (which consists of compromised endhost

machines). We denote $d^p(i,j)$ to be the total bandwidth of AD_j's private down-paths known only to AD_i.

THEOREM 1. *For private communication (using private static down-paths), AD_{src} can successfully send packets to AD_{dst} at rate $r^p = \min\{u_{src}, d^p(src,dst)\}$ without experiencing congestion on any intermediate links.*

Proof sketch: The first domain-based guarantee is straightforward. Since TDC is congestion-free (as described in Section 3.2), AD_{src} can establish end-to-end congestion-free channels by splicing its static up-paths and AD_{dst}'s private static down-paths. The sending rate of the resulting channels is dominated by the bottleneck bandwidth, which is the minimum of u_{src} and $d^p(src,dst)$. Both u_{src} and $d^p(src,dst)$ are independent of the botnet and other ADs' allocations. □

Note that r^p is a lower-bound guarantee of the sending rate, and the congestion-free property ensures that packets, such as connection setup requests, can be delivered at the first trial.

In Theorem 2, we show that AD_{src} can obtain a weaker guarantee (which depends on the static allocations of other ADs) when using public static down-paths. Let $U = \sum_{i=1}^{m} u_i$, and $U(i)$ be the total static up-path bandwidth activated by ADs that desire to communicate with AD_i. Let b_{dst} be the minimum cut of AD_{dst}'s BE-class bandwidth on the down-paths.

THEOREM 2. *For public communication (using unconcealed static down-paths), AD_{src} can successfully send packets to AD_{dst} at an average rate $r = u_{src}(\frac{d_{dst}}{U(dst)} + \frac{b_{dst}}{U})$.*

Proof sketch: Sources that desire to communicate with AD_{dst} compete for the limited bandwidth of static down-paths, d_{dst}. Sources do not need to compete with packets to other destinations on congested links because STRIDE performs weighted fair sharing on the static down-paths. To obtain the highest traffic priority and increase the chance of successful delivery, each source sends packets with no overuse bit on its static up-paths at full speed, resulting in $U(dst)$-amount of high-priority traffic to AD_{dst}. Hence, AD_{src} with u_{src} static allocation can send packets through static down-paths at rate $\frac{u_{src} \cdot d_{dst}}{U(dst)}$ (e.g., bit/s) on average. A similar result can be shown for the case where sources compete for the BE bandwidth on the congested links (b_{dst}). Since the BE class is shared, in the worst case AD_{src} has to compete with traffic between all ADs in the TD, resulting in a sending rate $\frac{u_{src} \cdot b_{dst}}{U}$. Hence, the average waiting time before successfully delivering a packet of size w is $\frac{w}{r}$. The waiting time for using the static channel and the static+BE channel is linear to $U(dst)$ and U, respectively. □

Theorem 3 shows that AD_{src} can obtain a lower-bound guarantee on the dynamic allocations to AD_{dst}. Let γ be the ratio of the dynamic-class bandwidth to the static-class bandwidth, and we assume γ is the same for every link for ease of description. $\Delta(i,j)$ is the guaranteed dynamic-class bandwidth that AD_i would like to allocate for communication with AD_j.

THEOREM 3. *For dynamic channels, STRIDE guarantees $\min\{\Delta(src,dst), \Delta(dst,src)\}$ amount of dynamic-class bandwidth for the flow aggregate between AD_{src} and AD_{dst}, where $\sum_{i=1}^{m} \Delta(src,i) \le \gamma \cdot u_{src}$ and $\sum_{i=1}^{m} \Delta(dst,i) \le \gamma \cdot d_{dst}$.*

Proof sketch: Because STRIDE performs weighted fair sharing within the dynamic class based on the static allocation, the flow aggregates from AD_{src} and to AD_{dst} are guaranteed to have $\gamma \cdot u_{src}$ and $\gamma \cdot d_{dst}$ bandwidth, respectively. Endpoint ADs can then freely divide the guaranteed bandwidth to flow aggregates going to/from different ADs. □

Splitting guaranteed bandwidth. Each endpoint AD decides how to divide its bandwidth guarantees among its endhosts based on its local policy. For example, a simple policy would be to split the static allocation based on per-host fair share. Suppose the sender and the receiver obtain a fair share u'_{src} and d'_{dst}, respectively, from their local domains. Similar to Theorems 1–3, we can show guarantees for the sender and receiver by replacing u_{src} with u'_{src} and d_{dst} with d'_{dst} in the proofs above. Moreover, because the number of dynamic channel setup requests within the static class is small and bounded, STRIDE can provide guaranteed high bandwidth to flows established through static channels. An interesting observation is that since compromised endhosts send traffic all the time whereas uncompromised endhosts do not, the available bandwidth is much higher than the fair share if the domain contains fewer bots.

Resilience against DoC and N^2 Attacks. Theorems 1 and 2 imply that a DoC attacker (1) cannot crowd out a capability request if the request is placed along a static channel with a private path, and (2) can delay a capability request at most by time linear to the static allocation of other domains if the request uses a public path. These bounds are independent of the size or distribution of the botnet. Theorem 3 implies that STRIDE can defend against the N^2 attack as the guaranteed flow bandwidth between the source and destination ADs is unaffected by bots outside those ADs.

7. EVALUATION

In this section, we evaluate STRIDE with respect to its effectiveness against DDoS attacks. We show the effectiveness of end-to-end bandwidth guarantees under large-scale attack scenarios. We also test the packet forwarding performance of STRIDE via real-field implementation.

Simulation setup. For realistic simulation, we use a CAIDA AS-relationship dataset to construct a TD; a tier-1 AD connecting to 2164 endpoint ADs is chosen as the TDC. Although the AS-relationship dataset does not include all interface-level paths, our analysis of the dataset reveals that AD-level path diversity is high enough to support STRIDE's path control and hence to evaluate STRIDE's path construction. Specifically, the endpoint ADs in the dataset have more than 40 different paths to the TDC on average. If interface-level paths are constructed, path diversity at endpoint ADs would become much higher since the number of paths grows exponentially as PCBs propagate downstream.

Bandwidth allocation. During PCB propagation, each AD allocates bandwidth to each child AD proportional to the child size. We assume that the size of an AD is proportional to its degree.

7.1 Resilience against DoC Attacks

We evaluate the resilience of STRIDE against DoC attacks, under the following simulation scenario. We randomly label one hundred ADs as *clean* (i.e., ADs containing no bots) and configure them to send traffic to a destination AD using 10 different down-paths (i.e., k=10), with a send rate equal to one tenth of the down-path capacity. Hence, in the absence of attacks, all down-paths are fully (but not overly) utilized. Then, we randomly label ADs as *contam-*

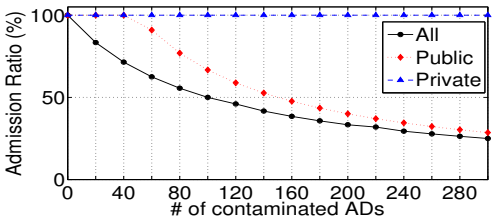

Figure 4: The admission ratio of legitimate packets for the number of contaminated ADs.

inated (ADs containing bots) and set their send rate equal to that of a clean AD so as to make individual contaminated ADs indistinguishable from clean ones. The number of contaminated ADs is increased from 0 to 300.

We evaluate the effectiveness of STRIDE against the above attacks in the following two scenarios.

1. *Public Paths:* All clean and contaminated ADs use the k activated down-paths to setup capabilities with the destination AD.
2. *Public and Private Paths:* Half of the clean ADs use a private down-path that was provided to the source ADs via a secret out-of-band channel. Meanwhile, the remaining half of the clean ADs and contaminated ADs use the public paths as before.

As an evaluation metric, we use the *admission ratio*, which is defined as the percentage of the legitimate packets (i.e., packets from the clean domains) that successfully traverse the bottleneck link/path.

Figure 4 shows that when all source ADs use the public paths ("All"), the admission ratio of the legitimate packets decreases as more contaminated ADs are added since the per-AD bandwidth decreases. When half of the clean ADs acquire a private path from the destination ("Private"), their packets are unaffected by the attack as the 100% admission ratio shows; and the packets of the remaining half of the clean ADs ("Public") obtained higher admission ratio along the public paths because the use of the private path reduced bandwidth contention along the public paths. This result illustrates how destination ADs can protect their valued customers' traffic from DDoS attacks in STRIDE.

While STRIDE enables private parties to use private paths to avoid congested static paths, it also protects clean ADs' traffic from large-scale DDoS attacks via packet prioritization: i.e., capability requests made through the static up-paths would have a higher priority than others through the best-effort up-paths. To examine this, we use the following simulation scenario: the attack strength is increased (by adding more attack sources within contaminated ADs) up to 10 times the bandwidth of the static up-paths; source ADs put the high priority marking on their outbound packets such that the bandwidth of high priority packets would not exceed that of the static up-paths (e.g., if the attack strength grows 10 times, 90% of attack packets would have a low priority marking). Legitimate source endhosts, on identifying congestion on static down-paths, use the best-effort down-paths; and attack source endhosts use the same path selection strategy as that of the legitimate sources to maximize their effects. The above attack scenario is the strongest attack scenario we consider for the given number of legitimate and attack sources since all packets from the 100 clean and 300 contaminated ADs compete for the bandwidth of public paths.

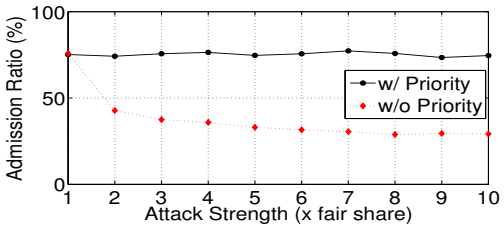

Figure 5: Effects of attack strength.

Figure 5 shows that even if the attack strength grows, the effects on legitimate traffic are marginal: attack sources, regardless of their strength, can only consume bandwidth proportional to their fair share both on the static and the best-effort channels. Meanwhile, 25% of legitimate packets sent through the static down-paths reach their destination without loss, and the other legitimate packets (i.e., 75% of them) sent through the best-effort channel reach the destination with a ratio close to 66.7%. Overall, 75% of the legitimate requests overcome the massive DDoS attack whose total send rate is 30 times higher than that of the legitimate sources, even if routers cannot distinguish between legitimate and attack packets. The figure also shows that without packet prioritization, the admission ratio of legitimate packets decreases as attack strength grows. This result shows the effectiveness of using static up-path and packet prioritization in STRIDE.

7.2 Flow Bandwidth Guarantees

STRIDE's bandwidth guarantees effectively isolate the bandwidth of attack traffic from that of legitimate traffic. As a consequence, in STRIDE, the effects of attacks are confined within the paths they follow regardless of whether attack sources flood a single path (or a link) or multiple paths simultaneously. We show this bandwidth isolation via large-scale simulations. For realistic simulations, we construct simulation topologies using a CAIDA SkitterMap [2], attach 10,000 legitimate sources to 200 ADs proportional to the AD size, and attach attack sources (hosts) to 100 ADs. Paths are probabilistically sampled from the SkitterMap to satisfy both the number of sources and the number of ADs. Legitimate sources control their packet sending rate based on the TCP congestion control mechanism, while attack sources send constant, high-rate traffic to flood a target link. We increase the attack size from 10K to 100K to compare STRIDE's bandwidth guarantees with those of a per-flow fair-sharing based mechanism. We consider a baseline case, labeled as "No Defense", where packets are randomly dropped during congestion.

Figure 6 shows the bandwidth used by the legitimate flows that originate from clean ADs. Under "No Defense", the legitimate flows obtain almost no bandwidth. DDoS attacks. When per-flow fair-sharing bandwidth control is em-

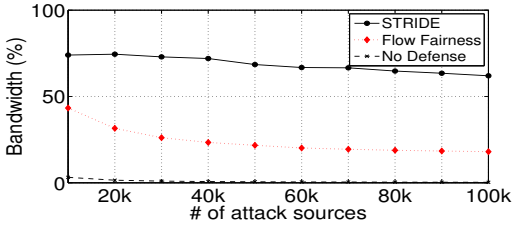

Figure 6: The effects of attack size.

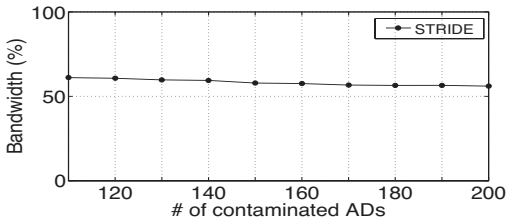

Figure 7: The effects of attack dispersion.

ployed, attack flows cannot completely exhaust the target's link bandwidth, yet the attack effects grow linearly with the attack size.

STRIDE provides consistent bandwidth guarantees to legitimate traffic under different attack sizes, which proves the effectiveness of path bandwidth isolation. The bandwidth of legitimate flows decreases slightly as the attack size grows, because (1) the extra bandwidth that is not fully used by some paths (due to the TCP congestion control) is shared by other flows, and (2) as the number of contaminated ADs increases, the number of clean ADs decreases (as the total number of ADs is fixed).

Next, we increase the number of contaminated ADs by 10 up to 200 ADs. As one can imagine, the bandwidth of legitimate flows decreases as Figure 7 shows. However, the effects of attack dispersion are marginal (i.e., proportional to the number of attack ADs) because the dynamic channel bandwidth is proportional to the static channel bandwidth and the static channel bandwidth that can be used by attack traffic is limited by the number of attack ADs in STRIDE.

7.3 Throughput

STRIDE introduces additional computational work for capability (or opaque field) verification. To gauge the computational overhead, we measure the throughput of a STRIDE router for various packet sizes and compare the result with that of the default IPv4 forwarding. We implement a STRIDE router as a user-space process using the Click Modular Router [10]. The capability generation and verification are implemented as CBC-MAC with AES-ni. We perform the measurement with a simple topology where a source and a destination are directly attached to a STRIDE router. NetPerf [3] is used for throughput measurement.

As described earlier, STRIDE forwards packets based on the interface identifier in the packet header; hence, unlike in today's routers, no overhead will be incurred for FIB (forwarding table) lookup. Meanwhile, the IPv4 forwarding in our experiments would produce the highest throughput that it can achieve since the FIB has only one entry in our network configuration.

Figure 8 shows that for small packets, both IPv4 and STRIDE routers under-utilize the link bandwidth while the IPv4 packet forwarding outperforms that of STRIDE; for

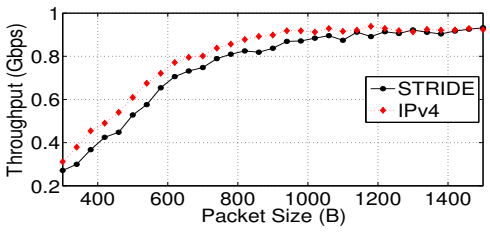

Figure 8: Throughput vs. packet size.

large packets, they both utilize more than 90% of the link bandwidth. In practice, the packet overhead becomes negligible since as small packets account for less than 10% of bandwidth and most of remaining packets are full sized [2].

8. DISCUSSION

Inter-TD traffic guarantees. While STRIDE provides domain-based guarantees for communication within a TD, many of the properties also translate for communication between TDs. For example, static channels on the up-paths still guarantee low-capacity throughput. Only BE channels are getting lower guarantees, as there is no explicit indication that the receiver desires the communication. Thus, establishing a connection to a public service that is under attack will be challenging for an external host. However, as soon as the service receives one initial packet and desires to serve that client, it can set up a dynamic channel with the same protected bandwidth guarantees within each TD assuming no congestion on high-capacity links between TDs.

Malicious ADs inside TD. ADs within a TD may get compromised, failing to regulate traffic. Although this is unlikely in well-administered ADs, attackers can nevertheless exploit software vulnerabilities in routers or administrative workstations. STRIDE can identify malicious ADs using neighborhood monitoring and existing fault detection protocols [32]. For example, if any AD sends more traffic than their allocated share, the AD must be malicious or misconfigured and the neighboring AD can block the offending traffic. As a technical defense, once the malicious AD is detected, hosts can avoid the malicious AD by selecting paths that avoid traversing that AD. Most importantly, since all ADs of a TD are within a uniform legal environment, the TDC can revoke the membership of misbehaving ADs.

Directional paths and asymmetric bandwidth requirements. In practice, network links may be directional or asymmetric with different bandwidths in the two directions. STRIDE can flexibly accommodate asymmetric paths with minimal modifications as follows. To request a packet on a directional path, the source puts in the header both the forward and backward paths. For example, dynamic flow capabilities can be requested on both the forward and return paths, and sent back to the other party through sufficient space allocated in the packet header. Bandwidth requests may be asymmetric, as in downloads the client-to-server bandwidth is one to two orders of magnitude smaller (acknowledgment packets are smaller than data packets). In this case, STRIDE supports asymmetric bandwidth allocations, where a uni-directional path token or flow capability is requested with different amounts of bandwidth depending on the direction.

9. RELATED WORK

Network-layer DDoS defense mechanisms can be largely classified into two categories: router-level bandwidth control and architectural extensions. One can generally combine approaches in these two categories for stronger properties.

Router-level bandwidth control. Typical router-level approaches to DDoS defense aim to filter or limit identified attack flows.

Filtering: Filtering approaches [5, 13] install filters against attack sources near their origins (i.e., source ADs) to prevent collateral damage of attack traffic. This would es-

sentially require trust establishment between ADs and rely on source ADs' cooperation that would incur substantial overhead for managing flow state and packet inspection. In contrast, STRIDE facilitates natural trust relationships between ADs within the same trust domain. Network-layer capability schemes [26, 29] enable routers to perform stateless filtering without needing any trust on other routers, but are vulnerable to the DoC attack [6]. Though Portcullis [19] addresses the DoC attack, it requires high computational overhead even on benign source hosts.

Bandwidth throttling: Many bandwidth control mechanisms (especially the fair queueing mechanisms) proposed to date can be used to prevent some (malicious) flows from exhausting the network bandwidth [14, 15, 18, 21, 25]. However, per-flow or per-sender fair sharing does not provide any guarantees by design as the fair bandwidth becomes too low as more entities (e.g., flows) compete for a limited resource. Moreover, existing mechanisms require source authentication, which is difficult to provide efficiently.

Bandwidth guarantees: Existing approaches [7, 8] aiming to provide bandwidth guarantees to flows fail in cases where all available bandwidth is exhausted. FLoc [12] differentiates legitimate flows from attack flows to provide differential bandwidth guarantees. Low-rate attack flows, however, can often not be precisely distinguished from legitimate flows, thereby the lower bound of bandwidth may not be observed.

Architectural support. SCION inherently provides a default level of protection against DDoS attacks. For example, SCION's periodic topology discovery and resulting path-diversity by default enable agile path adjustment to avoid attacked areas. However, SCION itself does not provide any DDoS defense guarantee. Several other next-generation Internet architectures [17, 20, 27], instead of providing intrinsic DDoS resilience, aim to provide routing flexibility, path diversity [27], expressive routing policies [17, 20], etc. A line of multi-path routing protocols have also been proposed [9, 11, 16, 24, 28] to provide path diversity to the *source* nodes. Although the source nodes can utilize the path diversity to circumvent victim links/routers under a DDoS attack, the destinations are still left with little inbound traffic control. Furthermore, these protocols are built on top of the current Internet, thus still suffering from the underlying weaknesses of today's Internet. For example, local identification of attack sources can be imprecise or impossible to counter large-scale botnet attacks that do not directly target a specific service or endpoint.

10. CONCLUSION

A core goal of the STRIDE architecture is to achieve intrinsic DDoS defense with relatively simple routers. In particular, we avoid per-flow state in the fastpath, asymmetric cryptographic operations, reliance on untrustworthy domains, and key establishment across ADs. Even with our relatively simple operations, we can achieve protection against DDoS from large botnets. Reflecting on the STRIDE architecture, we observe that measured trust in ADs located within the same legal environment providing viable prosecution helps to simplify the architecture and results in higher efficiency, meanwhile the untrustworthy ADs outside the trust domain cannot inflict damage against local within-trust-domain communication. We anticipate that STRIDE provides a useful point in the design space to study holistic network architectures with strong DDoS defense properties.

11. ACKNOWLEDGMENTS

This research was supported by CyLab at Carnegie Mellon, and by support from NSF under awards CCF-0424422 and CNS-1040801. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of CMU, NSF or the U.S. Government or any of its agencies.

12. REFERENCES

[1] Arbor networks: Infrastructure security survey. `http://www.arbornetworks.com/sp_security_report.php`.

[2] CAIDA: The cooperative association for internet data analysis. `http://www.caida.org/`.

[3] Netperf benchmark. `http://www.netperf.org/netperf/`.

[4] T. Anderson, T. Roscoe, and D. Wetherall. Preventing Internet denial-of-service with capabilities. In *Proceedings of ACM HotNets*, 2003.

[5] K. Argyraki and D. R. Cheriton. Active internet traffic filtering: real-time response to denial-of-service attacks. In *Proceedings of the ATEC*, 2005.

[6] K. Argyraki and D. R. Cheriton. Network capabilities: The good, the bad and the ugly. In *Proceedings of ACM HotNets*, 2005.

[7] F. Bonomi and K. Fendick. The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service. In *IEEE Network Magazine, vol. 9, no. 2*, 1995.

[8] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (Proposed Standard), Sept. 1997. Updated by RFCs 2750, 3936.

[9] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica. Pathlet routing. In *Proceedings of ACM SIGCOMM*, 2009.

[10] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, August 2000.

[11] N. Kushman, S. Kandula, D. Katabi, and B. M. Maggs. R-BGP: Staying Connected In a Connected World. In *Proceedings of USENIX NSDI*, 2007.

[12] S. B. Lee and V. Gligor. FLoc: Dependable link access for legitimate traffic in flooding attacks. In *Proceedings of IEEE ICDCS*, 2010.

[13] X. Liu, X. Yang, and Y. Lu. To filter or to authorize: network-layer dos defense against multimillion-node botnets. In *Proceedings of ACM SIGCOMM*, 2008.

[14] X. Liu, X. Yang, and Y. Xia. NetFence: Preventing Internet Denial of Service from Inside Out. In *Proceedings of ACM SIGCOMM*, 2010.

[15] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *Comput. Commun. Rev.*, 2002.

[16] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala. Path splicing. In *Proceedings of ACM SIGCOMM*, 2008.

[17] J. Naous, M. Walfish, A. Nicolosi, D. Mazieres, M. Miller, and A. Seehra. Verifying and enforcing

network paths with ICING. In *Proceedings of ACM CoNEXT*, 2011.

[18] R. Pan, B. Prabhakar, and K. Psounis. CHOKE, a stateless active queue management scheme for approximating fair bandwidth allocation. In *Proceedings of IEEE INFOCOM*, 2000.

[19] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu. Portcullis: Protecting connection setup from denial-of-capability attacks. In *Proceedings of ACM SIGCOMM*, 2007.

[20] B. Raghavan and A. C. Snoeren. A system for authenticated policy-compliant routing. In *Proceedings of ACM SIGCOMM*, 2004.

[21] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high-speed networks. *IEEE/ACM Trans. Netw.*, Feb. 2003.

[22] A. Studer and A. Perrig. The coremelt attack. In *Proceedings of ESORICS*, 2009.

[23] A. Vulimiri, G. A. Agha, P. B. Godfrey, and K. Lakshminarayanan. How well can congestion pricing neutralize denial of service attacks? In *Proceedings of ACM SIGMETRICS*, 2012.

[24] W. Xu and J. Rexford. MIRO: Multi-path Interdomain Routing. In *Proceedings of ACM SIGCOMM*, 2006.

[25] Y. Xu and R. Guérin. A double horizon defense design for robust regulation of malicious traffic. *Proceedings of SecureComm*, 2006.

[26] A. Yaar, A. Perrig, and D. Song. SIFF: A stateless internet flow filter to mitigate ddos flooding attacks. In *Proceedings of IEEE Symposium on Security and Privacy*, 2004.

[27] X. Yang, D. Clark, and A. W. Berger. NIRA: a new inter-domain routing architecture. *IEEE/ACM Trans. Netw.*, 2007.

[28] X. Yang and D. Wetherall. Source selectable path diversity via routing deflections. In *Proceedings of ACM SIGCOMM*, 2006.

[29] X. Yang, D. Wetherall, and T. Anderson. TVA: a dos-limiting network architecture. *IEEE/ACM Trans. Netw.*, Dec. 2008.

[30] B. Yener, Y. Ofek, and M. Yung. Combinatorial design of congestion-free networks. *IEEE/ACM Transactions on Networking (TON)*, 1997.

[31] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen. SCION: Scalability, control, and isolation on next-generation networks. In *Proceedings of IEEE Symposium on Security and Privacy*, 2011.

[32] X. Zhang, Z. Zhou, H.-C. Hsiao, T. H.-J. Kim, A. Perrig, and P. Tague. ShortMAC: Efficient data-plane fault localization. In *Proceedings of NDSS*, 2012.

APPENDIX

A. LINK CAPACITY DIVISION

Using estimations, we show how STRIDE ADs can divide their link capacity to the three traffic classes, and how much static bandwidth can be allocated to a path.

Division of link capacity to three traffic classes. We provide guidelines that an AD can follow to divide its total link capacity to three traffic classes: static, dynamic, and BE. First, given that the current real-world link utilization is mostly below 30% based on the CAIDA dataset [2], allocating 30% of the link capacity to the BE class would satisfy legacy Internet traffic in most cases. Subsequently, assuming the static and dynamic classes are allocated s and d fractions of the link capacity, respectively, the following conditions should hold:

$$s + d = 1 - 30\% \tag{2}$$

$$40\text{Gbps} \times s > 500\text{Kbps} \times 10000 \tag{3}$$

The first condition ensures a link will not be overloaded when each bandwidth class is being fully utilized. The second condition assumes an OC-768 link capacity (40 Gbps) to be divided among around 10000 paths, such that each endpoint in a medium-size TD (e.g., a US TD with around 2200 ADs, according to the CAIDA dataset) can choose up to 10 paths in our experiment. The second condition requires the static bandwidth allocated to each path be no less than 500 Kbps.

Based on these guidelines, a reasonable example allocation is to divide 5 – 15%, 60 – 65%, and 30% link capacity to the static, dynamic, and BE traffic classes, respectively. In practice, an AD can adjust the numbers in the conditions based on its own link capacity, number of current paths that the AD supports, etc. Furthermore, when any bandwidth class is not fully utilized, other congested traffic class can take up all the bandwidth that is currently available.

B. BANDWIDTH OVERBOOKING

Section 5.1 introduces bandwidth overbooking for simultaneous enhancement of path quality and diversity, but with possible denial of path activation. To mitigate this issue, we suggest an appropriate overbooking ratio by analyzing the relationship between an overbooking ratio and the corresponding probability of path activation denial as follows.

In the following, we consider an intermediate AD AD_p wanting to determine its overbooking ratio. Let I_i and E_j represent the i^{th} ingress interface from the providers ($0 \leq i \leq l$) and the j^{th} egress interface to the customers ($0 \leq j \leq m$), respectively. Let each ingress interface connect to all m egress interfaces. We assume that m interfaces connect to n customer ADs (i.e., each customer AD has $\frac{m}{n}$ links to AD_p). Then, each customer AD has at least $\frac{l \cdot m}{n}$ distinct paths to the TDC through AD_p. In this setting, suppose each customer AD selects uniformly at random k out of the $\frac{l \cdot m}{n}$ paths to the TDC, then the probability that the customer ADs select I_i more than t times in total would be: $P_{I_i}(t) \approx 1 - \sum_{i=0}^{t} e^{-\lambda} \cdot \frac{\lambda^i}{i!}$, where $\lambda = \frac{n \cdot k}{m}$. This implies that I_i's bandwidth needs to be allocated to t egress interfaces (out of m interfaces), which would increase per-path bandwidth allocation by $\frac{t-\beta}{\beta}$, where $\beta = \frac{n \cdot k}{l}$ is the average number of activated paths through I_i. If $t \gg \frac{n \cdot k}{l}$, sufficient path diversity (as much as $\frac{t \cdot l}{n \cdot k}$) is provided to customer ADs.

As a result, AD_p may determine t such that the probability of the denial of path activation does not exceed some threshold P_{th} (i.e., $P_{I_i}(t) \leq P_{th}$). For example, $P_{th} = 0.2$ means that 80% of path activation requests would be accepted on average; hence, the expected number of trials for successful path activation becomes 1.25. That is, P_{th} determines the number of requests that should be made by an endpoint AD until successful path activation.