# CoDef: Collaborative Defense Against Large-Scale Link-Flooding Attacks

Soo Bum Lee*
Qualcomm
San Diego, CA USA
soobuml@qti.qualcomm.com

Min Suk Kang
ECE and CyLab
Carnegie Mellon University
Pittsburgh, PA USA
minsukkang@cmu.edu

Virgil D. Gligor
ECE and CyLab
Carnegie Mellon University
Pittsburgh, PA USA
gligor@cmu.edu

## ABSTRACT

Large-scale botnet attacks against Internet links using low-rate flows cannot be effectively countered by any of the traditional rate-limiting and flow-filtering mechanisms deployed in individual routers. In this paper, we present a collaborative defense mechanism, called *CoDef*, which enables routers to distinguish low-rate attack flows from legitimate flows, and protect legitimate traffic during botnet attacks. CoDef enables autonomous domains that are uncontaminated by bots to collaborate during link flooding attacks and reroute their customers' legitimate traffic in response to requests from congested routers. Collaborative defense using multi-path routing favors legitimate traffic while limiting the bandwidth available to attack traffic at a congested link. We present CoDef's design and evaluate its effectiveness by exploring the domain-level path-diversity of the Internet and performing simulations under various traffic conditions.

## Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General—*security and protection*; C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*collaborative defense*

## Keywords

DDoS defense; collaborative defense; link-flooding attack; rerouting; bandwidth guarantees

## 1. INTRODUCTION

Current botnets can flood most Internet links in coordinated attacks. Unlike attacks that target an individual service, link-flooding attacks are particularly dangerous since they can disrupt *all* services that happen to use targeted links, including some that need stable connectivity; e.g., electric grid networks, financial services, or government services.

Several link-flooding attacks have been described in literature and observed in the Internet recently. For example, the Coremelt attack [30] can flood several core network links using only bot-to-bot flows that cross those links. The Crossfire attack [18] can flood a small set of selected network links using low-rate flows from bots to publicly accessible servers and degrade connectivity of, and even disconnect, chosen end-point servers. Recently, we witnessed the first large-scale Internet attack where an adversary flooded a few selected links of four major Internet exchange points (IXPs) in Europe and Asia [1] to degrade the connectivity of a targeted cloud service [3]. This attack, known as the Spamhaus attack, used high-rate flows, which were easily distinguished from the legitimate flows by the targeted IXPs and quickly pushed back by filters installed at IXPs' upstream providers. However, one could anticipate large-scale attacks, such as Coremelt and Crossfire[1], where service providers could not easily distinguish low-rate attack flows from legitimate flows. Thus pushback and other traffic filters could not be selectively applied only to attack flows and consequently would cause massive collateral damage to legitimate flows. The unpleasant alternative to collateral damage would be *persistent* flooding. That is, the adversary would be able to flood a network link, virtually indefinitely, by employing adaptive attacks that circumvent weaker defenses designed to avoid collateral damage; e.g., simple flow rerouting to disperse excess traffic. We note that limited adaptation has already been employed to increase persistency of flooding attacks in practice; e.g., the Spamhaus attack switched to flooding network links once the targeted server was massively replicated in a cloud to disperse the initial attack flows [3] .

To be persistent, a link-flooding attack has to meet two necessary conditions: first, it should use large numbers of flows that are *indistinguishable* from legitimate ones, and second, it should *adapt* to and circumvent deployed countermeasures. Attack flows must be indistinguishable from legitimate flows because otherwise they would be easily detected and filtered out in routers. Similarly, attack adaptation to circumvent deployed countermeasures is required because otherwise the attack would be easily handled by ordinary measures such as simple rerouting.

*Problem.* The problem we address in this paper is that current link-flooding countermeasures are unable to mitigate persistent attacks. They cannot distinguish low-rate attack flows from legitimate flows, and thus typical rate-limiting and flow filtering mechanisms in *individual* routers [13,19,28] cannot selectively block only

*This work was done while the author was at the University of Maryland and CyLab, Carnegie Mellon University.

[1]Legitimate flows are those not originated by an adversary. The Crossfire and Coremelt attacks use flows that are indistinguishable from legitimate ones. Crossfire uses *legitimate-looking* low-rate flows to publicly accessible servers, which would not trigger any alarms [18]. Coremelt relies on bot-to-bot communication and uses only *wanted* flows by destinations [30], which could also be low-rate.


*CoNEXT'13,* December 9-12, 2013, Santa Barbara, California, USA.

http://dx.doi.org/10.1145/2535372.2535398 .

attack flows and avoid collateral damage to legitimate flows. Furthermore, even if low-rate attack flows could be distinguished from legitimate flows at a targeted link, current countermeasures would be inadequate to provide bandwidth guarantees to legitimate flows. For example, the flow rerouting necessary for mitigating such attacks could not be employed since the inter-domain routing (i.e., BGP4) and all proposed multi-path routing decisions [25, 33, 36] are made at the source ASes, where precise attack information against a remote target link is unavailable. Hence, an adversary could achieve attack persistency with little or no attack adaptation.

*Solution.* In this paper, we present *CoDef*, a collaborative defense against persistent link-flooding attacks. CoDef comprises two complementary mechanisms, namely collaborative rerouting and collaborative rate control. In *collaborative rerouting*, an AS whose links are flooded sends reroute requests (along with the preferred paths) to all flow-source ASes. It does this to provide detours around a flooded router or link to the legitimate flows and to relieve congestion. Furthermore, the target AS performs a *rerouting compliance test* based on the flow-source ASes' reaction to the request. The test helps deny the first necessary condition of attack persistence since low-rate attack flows would have to defy rerouting requests and, as a result, could be distinguished from legitimate ones which would reroute. That is, the test forces the adversary to make an untenable choice: either conform to rerouting and hence give up attack persistence at the targeted link, or not conform and have its flows discovered and bandwidth-limited.

In *collaborative rate control*, CoDef implements an additional compliance test that helps distinguish between bot-contaminated and uncontaminated ASes. That is, a router that is subject to a flooding attack sends *rate-control requests* to all ASes. Upon receiving the rate-control requests, bot-uncontaminated ASes prioritize all their outgoing packets that reach the targeted AS so that they would be rate-control compliant. Again, this would force the adversary to make another untenable choice: either its bot-contaminated ASes would conform to rate-control requests and hence diminish the attack's persistence, or not conform and have their flows bandwidth-penalized. In contrast, compliant ASes would be rewarded with higher bandwidth allocation.

Once the potential attack flows and ASes are identified, the targeted router denies the second necessary condition for attack persistence. That is, the router prevents the adversary's ability to adapt and keep affecting rerouted legitimate flows; e.g., decrease their available bandwidth. It accomplishes this by forcing attack flows to maintain their original path and limiting their bandwidth.[2]

*Deployment.* CoDef can be easily deployed in the current Internet, since it does not require any changes to the existing routing systems. In fact, CoDef is a *complementary routing system* that operates on top of the existing inter-domain and intra-domain routing systems (e.g., BGP and OSPF), and honors the routing policies of individual ASes. Furthermore, CoDef requires neither disclosure of intra-domain topology nor route import/export policies of an AS. Finally, CoDef offers both economic and technical incentives to ASes that implement it. For example, it reroutes legitimate flows so that they experience better quality of service even during a link-flooding attack. Thus, CoDef can create new business opportunities for ISPs by supporting premium customer services. The added cost of rerouting would be more than offset by the added revenue generated by premium-service charges.

*Contributions.* In summary, CoDef offers:

- a practical mechanism for defending against persistent large-scale link-flooding attacks.
- two effective tests to identify the attack flows to ASes. Since these tests exploit an adversary's untenable choices (i.e., either loss of attack persistency or attack discovery) rather than rely exclusively on anomalous traffic detection, they would work against any variant of persistent link-flooding attacks.
- significant deployment incentives; i.e., it can be used without changing the operation of existing routing systems and provides both technical and economic advantages to its adopters.

# 2. CODEF OVERVIEW

In this section, we present an overview of CoDef consisting of collaborative rerouting and rate-control mechanisms.

## 2.1 Collaborative Rerouting

Collaborative rerouting is an extension to existing Internet routing system (i.e., BGP) that allows congested routers to send reroute requests to the source ASes for the purposes of relieving congestion, distinguishing attack flows from legitimate flows accurately, and initiating timely defense against persistent link-flooding attacks.

An essential ingredient of collaborative rerouting is a path identification mechanism [21], where every packet carries an *identifier* that captures all the ASes traversed from a packet's origin to its destination[3]. Path identifiers are required by CoDef because attack targets must discover the flows' source ASes and forwarding paths both to identify attack ASes and locate ASes that are best suited to perform rerouting. (We note, in passing, that path identifiers are not a unique requirement of CoDef; e.g., recent proposals for secure Internet protocols (e.g., AIP [7], Passport [23], SCION [37] ) also need such identifiers.)

Collaborative rerouting starts a congested router's requests to source ASes to reroute their traffic and relieve congestion at that router. A reroute request includes all the ASes that need to be avoided on the forwarding path, the list of preferred ASes (ordered by their priority) through which forwarded traffic would experience less, or no, congestion,[4] and the destination prefix(es) to which rerouting is applied. A reroute control message is exchanged between *route controllers* located in individual ASes, where a controller communicates with all routers in its own AS to handle requests from congested routers and to apply route selection policies. For single-homed source ASes (i.e., when no alternate path is available at a source AS), a rerouting request is made to its provider AS, which usually has multiple connections to tier-1 or tier-2 ASes, or is a tier-2 AS in itself. The architecture and the function of route controllers will be discussed in detail in Section 3.

In response to a reroute request, source ASes refer to their routing (BGP) table to find alternate paths to the destination via the preferred ASes. If such a path does not exist, a source AS selects another path which excludes the ASes that need to be avoided on the forwarding path. The alternate path found in the routing table is set to the default path to the destination prefix; i.e., it is set to the forwarding table (FIB).

[2]Notice that CoDef does *not* block potential attack flows but instead pins them down on their initial path and controls their bandwidth (viz., Section 2.3 for details). This avoids any possibility of collateral damage for misidentified flows and ASes.

[3]A path identifier is placed on each packet leaving an AS by the AS' border router and is interpreted only by upgraded routers, Hence the path identification mechanism can be independently and incrementally deployed. Protection of path identifiers from potential attacks (i.e., forgery and replay attacks) is described elsewhere [21].

[4]Preferred ASes are used when the attack target cannot be excluded in the path.

Collaborative rerouting is effective whenever source ASes can easily find alternate routes in forwarding packets. This is the case for the vast majority of ASes. For example, AS-level path diversity, which was comprehensively explored in MIRO [33], shows that most of ASes (at least 95% of 300 million AS pairs tested) have alternate AS paths to reach a specific destination when 1-hop immediate neighbors' paths are counted. Unlike MIRO, where the alternate routes are found through negotiation among the neighboring ASes, we use the AS path diversity in an end-to-end manner. End-to-end path negotiation is more appropriate in link-flooding defense because a source AS receives an accurate and timely alert of persistent router congestion enabling it to find alternate paths without delay[5].

**Rerouting Compliance Test.** Using the collaborative re-routing mechanism, a target router can perform a compliance test to identify ASes that originate (low-intensity) attack flows. Specifically, the router can determine whether a source AS is contaminated by bots; i.e., whether an AS is attack or a legitimate AS.

To perform a rerouting compliance test, a congested router sends a reroute request to a source AS specifying a flow aggregate with specific destination prefix(es). A flow *aggregate* is a subset of the flows carrying the same path identifier; that is, following the same path from a source AS to the target AS. Then, the router monitors the flow aggregate from the source AS and identifies whether the source AS is an attack AS. That is, the router either observes a *persistent* influx of the same flow aggregate (which would mean that the source AS ignored the reroute request) or discovers the influx of *new* flows from the source AS instead of the initial flow aggregate (which would mean that the source AS *pretends* to be legitimate and yet creates new to attack the targeted link). Thus, the only way for an attack AS to be identified as legitimate is to behave as a legitimate AS; i.e., to conform to the reroute request and not create new attack flows. In other words, the attack has to fail (i.e., lose persistency) in order to pass the compliance test[6]. It is worth noting that this compliance test is effective against any variant of link-flooding attacks because it denies the *goal* of the adversary and does not exclusively rely on detection his/her anomalous traffic behavior. Giving up the attack is the only way for an attack (i.e., bot-infested) AS to pass the compliance test.

## 2.2 Collaborative Rate Control

CoDef's collaborative rate-control mechanism requires source ASes to establish the service priorities of their out-going flows to comply with a rate-control request received from a congested router. This mechanism enables a congested router to control the bandwidth of legitimate flows originating from compliant ASes. Collaborative rate control is used independently of collaborative rerouting and is necessary in the rather rare cases when collaborative rerouting cannot separate the legitimate flows from the attack flows due to the lack of multi-path diversity.

The basic mechanism works as follows. The route controller of an AS with a congested router requests a source AS to classify its flows into three categories (i.e., high priority, low priority, and to-be-filtered) by providing two threshold values: the guaranteed bandwidth and the (maximally) allocated bandwidth. Then, the egress router(s) of the source AS adds corresponding priority markings to its out-going packets based on its identification of attack flows or its business relationship (e.g., a premium service by contract) with customers. The congested router guarantees service to the high priority packets, provides a best effort service to low priority ones, and drops all other packets until its link becomes idle.

Our rate-control mechanism provides sufficient incentive to compliant source ASes for several reasons. Compliant source ASes can provide better service to premium customers since they need not severely restrict the bandwidth of their outgoing traffic, and can be allocated higher bandwidth at the congested link by conforming to rate-control requests. This is explained in the next subsection.

**Rate-Control Compliance Test.** Using the collaborative rate-control mechanism, a target router can test whether source ASes comply with its rate-control requests and adjusts the bandwidth allocation in order to reward rate-control compliant ASes. Although each AS is allocated the same guaranteed bandwidth, whenever these allocations are not fully subscribed, the target router would reallocate residual bandwidth to rate-control compliant ASes. Thus, a compliant source AS could be allocated additional bandwidth that would be unavailable otherwise. In effect, the target router penalizes non-compliant ASes while rewarding compliant ones.

## 2.3 Path Pinning

Once the potential attack flows and ASes are identified by either of the two compliance tests described above, the target router pins them down to the initial path while rerouting the other legitimate flows[7]. The path pinning mechanism denies an adversary the ability to adapt and thus guarantees better quality of service to the legitimate flows by minimizing the effect of the attack.

A congested router sends path-pinning requests to source/provider ASes that originate/forward attack flows. Whenever a route controller of a source AS receives a path-pinning request, it configures its BGP routers to suppress any route-update message containing the requested destination prefixes. In effect, this leaves the initial route to the destination prefixes unchanged. If a path-pinning request is made to a provider, the provider sets up a tunnel for the flows destined to the target link. Various tunneling mechanisms are discussed in Section 3.2.

Note that CoDef does *not* block the potential attack flows but instead pins them down on the initial path and controls their rates at the *source* ASes in order to avoid any denial of service to legitimate flows. There are two cases where a potential attack AS generates legitimate flows: (1) the AS is actually legitimate but could not find alternative paths to honor a rerouting request; (2) the AS is malicious but contains some legitimate users who generate legitimate traffic. In both cases, CoDef's path pinning can protect the legitimate flows from being unintended victims of collateral damage.

# 3. CODEF ARCHITECTURE

In this section, we describe the system architecture and implementation details of the proposed mechanisms.

## 3.1 Route Controller

CoDef introduces a specialized server, named the *route controller*, in each participating AS. Route controllers send/receive route-control messages to/from other route controllers or routers in their AS, as shown in Fig. 1. A route controller processes congestion-notification

[5]In MIRO, a source AS would need to spend significantly more time to determine accurately which AS is being targeted by a flooding attack and then find the best alternate path.

[6]The attack AS might try to hibernate for a long enough time to have the target router falsely conclude that the attack AS is legitimate, and then resume the flooding. However, the attack would still fail because the target router will request rerouting again and thus the flooding could not be persistent after all.

[7]SNAPP [26] introduced the concept of path-pinning as a way to authenticate legitimate flows along their path. In contrast, we use the same concept to trap (potential) attack flows. This would deny an adversary's ability to adapt and keep affecting rerouted legitimate flows.

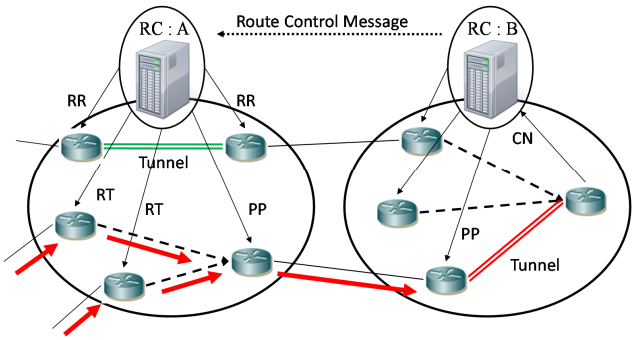



Figure 1: A congested router sends congestion notification (CN) message to its route controller, RC:B, which sends route-control messages to route controller RC:A. Route-control messages comprise: rerouting (RR) messages, which establish a tunnel between ingress and egress routers; path-pinning (PP) messages, which suppress route updates from the downstream ASes; and the rate-control (RT) message, which make their routers mark/drop packets.

(CN) messages received from routers in its AS and sends route-control messages to source/provider ASes. The route controller also sends rerouting (RR) messages to legitimate ASes, rate-control (RT) messages to both legitimate and attack ASes, and path-pinning (PP) messages to attack ASes. The detailed packet formats for these messages are described in Section 3.4. The recipient of a route-control message identifies the BGP routers that can handle the control request and configures them to direct flows to the congested link as requested.

A route controller has complete knowledge of the network topology of its AS and can establish secure sessions with BGP routers. Route controllers discover network topologies by overhearing or participating in intra-domain routing protocols (i.e., IGP). The route controllers and BGP routers can establish secure connections (e.g., by sharing a secret key) as they operate within the same administrative control. Route controllers can be gracefully integrated with the current Internet architecture since they do not require any modification to the existing control-plane protocols. Furthermore, they can be easily implemented in a software-defined networking (SDN) architectures [11, 24, 35], where a central controller defines policies (e.g., forwarding), makes control decisions, and configures the switches of its domain to monitor flows.

*Security of Route-control Messages:* For message authentication, a route controller shares secret keys with each router of its AS and has a private/public key pair certified by a trusted third party (e.g., ICANN). Intra-domain messages are protected by message authentication codes (MAC) generated with the shared secret keys; e.g., $MAC_{K_{ASR_i}}$ is generated with key $K_{ASR_i}$ shared by controller of AS and router $R_i$. When a route controller receives a route-control message from a congested router, the route controller verifies the MAC of the message and, if the MAC is correct, forwards the message to the route controller of the intended destination AS, after replacing the MAC with its signature. This message is authenticated by the destination route controller using the sender's public key. We assume that ASes' certificates are available from a globally trusted repository such as RPKI [22].

## 3.2 Route Management

During flooding attacks, a congested router constructs a traffic tree using the path identifiers it receives. (Recall that a path identifier is the ordered list of ASes along which a packet traveled.) It also estimates the proportion of attack traffic (i.e., the aggregate flows originating from the same AS) that each path identifier delivers. This estimate is done by using the rerouting and rate-control compliance tests. Then, the congested router categorizes ASes into two classes: (1) attack ASes, which originate attack traffic, and (2) legitimate ASes, can still be affected by attack traffic due to their locations; i.e., many of their paths to destinations may be shared with attack traffic. The congested router controls the route of each AS based on its class.

### 3.2.1 Rerouting

A route controller of a source or provider AS performs rerouting as follows.

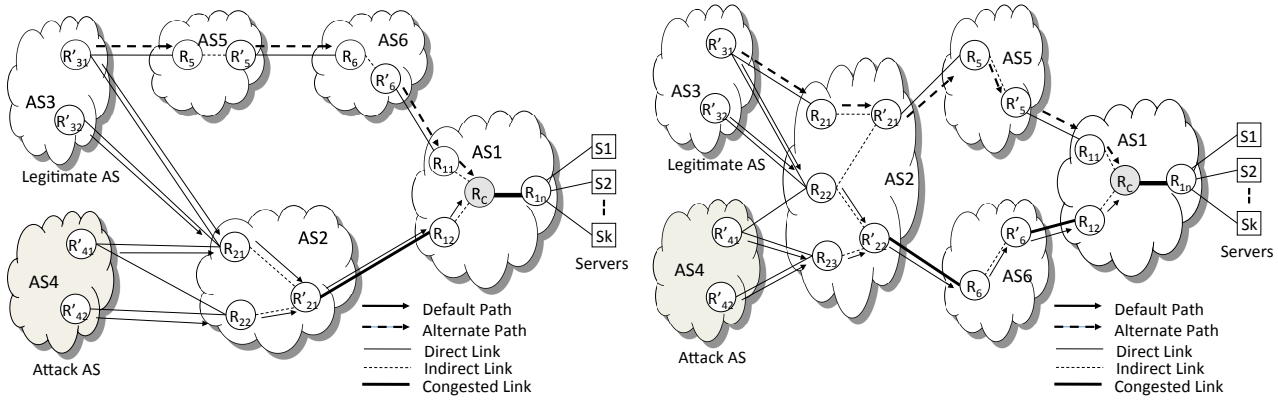
*Source AS.* When the route controller of a multi-homed source AS receives a reroute request from a congested router, it finds a new next-hop AS among its multiple provider ASes for the corresponding flows. Referring to its BGP table, the route controller selects the one provider AS that would deliver the corresponding flows through the particular ASes suggested by the target router . If multiple paths satisfy the reroute request, the route controller selects a path based on the priority defined in the route selection process [12]. The route controller sets the selected path as the default path in forwarding traffic to the router, by assigning the highest local preference value to the path[8]. The local preference value is set by the BGP router before propagating a route update internally; e.g., configure $R'_{31}$ to advertise higher Local Preference value than $R'_{32}$ in Fig. 2(a) to change the default path.

*Provider AS.* A provider AS handles a reroute request in the same way as a multi-homed source AS if the request is for all its customer ASes. However, if the request is for a specific subset of customer ASes,[9] the provider sets up tunnels to the next-hop AS to reroute those customer ASes' traffic, while leaving the default path intact; i.e., performs multi-path routing. An AS can choose a tunneling mechanism from several available options, such as IP-in-IP or MPLS tunneling [29], depending on its current intra-domain operations (e.g., routing optimizations and traffic engineering policies). To do this, the route controller sends a tunneling request to the ingress router(s) (e.g., $R_{21}$ in Fig. 2(b)) connected to the corresponding customer AS. The tunneling request contains: (1) the destination prefix that needs to be tunneled and (2) the IP address of the egress router through which the next-hop AS is reached. (Alternatively, the IP address of the next-hop AS' ingress router can be used for tunneling.) Whenever an ingress router (e.g., $R_{21}$ in Fig. 2(b)) receives a packet whose destination address matches the requested prefix for tunneling, it encapsulates the original IP packet in the new IP packet and forwards the packet to the egress router (e.g., $R'_{21}$ in Fig. 2(b)). When the egress router receives the packet, it decapsulates it (i.e., peels off the outer IP header) the packet and forwards it to the next-hop AS. We note that packet processing would not impose much overhead on BGP routers since it only applies to flows carrying destination prefixes. We also note that intra-domain traffic engineering, including potential intra-domain, multi-path routing (e.g., [17, 25]), can be performed independently without having to disclose network topology to outside ASes.

*Target AS.* An AS which contains a link that is subject to an attack (i.e., the target AS) can reroute legitimate traffic through a less congested or non-congested *internal* path (i.e., from ingress

[8] *Local Preference* has the highest priority in the BGP route decision process.

[9] This happens when the provider AS also has *identified* attack ASes as its customers, and the targeted router wants their paths to be pinned.




(a) Multi-homed (b) Single-homed

Figure 2: Collaborative rerouting: AS3 is a legitimate AS whose traffic is rerouted, and AS4 is an attack AS. (a) Rerouting at the multi-homed source AS (i.e., AS 3). (b) Rerouting at the provider AS (i.e., AS 2) for a single-homed source AS (i.e., AS 3).

router to the target link) by adjusting the route-export policies at its border routers. To reroute, the target AS must have *multiple* incoming interfaces to an upstream AS at different locations (i.e., border routers). Whenever an upstream AS receives multiple route announcements for the same address prefix from different routers in an AS, it selects its next-hop router based on the MED (multiple exit discriminator) attribute in the route announcement, where a lower MED value is preferred over a higher value. This enables the target AS to reroute incoming traffic to an alternate router-level path (as opposed to an AS-level path) by changing MED values. Intra-domain rerouting at the target AS is necessary to control the paths of the source ASes that are located close to the target AS, and hence may not find alternate AS paths to the target otherwise.

### 3.2.2 Path Pinning

When the route controller of an AS is requested to pin a path, it configures its BGP routers to suppress any route-update messages containing the requested destination prefix. At the same time, the route controller configures the routers of its AS to disable any intra-domain route optimization (e.g., intra-domain, multi-path routing) on the flows carrying the requested destination prefix. This helps mitigate the side-effects of attacks on other legitimate flows.

Path pinning can be implemented using either standard *multi-topology routing* [2, 27] or *network-layer capability scheme* [32, 34]. In a multi-topology implementation, one of the several topologies (i.e., forwarding tables) stored in a router are assigned to the pinned path. With capabilities, a flow's routing information is embedded in a capability as follows. A router $R_i$ generates capability $C_{R_i}(f)$ for a flow $f$ as $C_{R_i}(f)$ = RID $||MAC_{K_{Ri}}(\text{IP}_S, \text{IP}_D, \text{RID})$, where $K_{Ri}$ is the router's secret key, $\text{IP}_S$ is the source IP, $\text{IP}_D$ is the destination IP, and RID is the id of an *egress router* to which the packet is forwarded. The router issues capability $C_{R_i}(f)$ during the connection setup phase of a flow $f$ and the packet destination provides the capability to the source for further packet transmission[10]. Hence, capability-enabled routers can filter address-spoofed and unwanted packets by their destination, and also tunnel packets to the router identified by an RID. We assume that a unique and private (i.e., meaningful within the AS) RID can be assigned to the BGP routers of an AS, and each RID can be mapped to the IP address of the corresponding router.

[10]The connection setup phase refers to the first phase of creating a network capability [32, 34]. That is, when a client establishes a connection to a destination server, each router en route from the client to the destination issues a capability for the new flow.

## 3.3 Rate Control

In this section, we explain how CoDef allocates bandwidth for each path identifier, how each router at the source AS regulates its outgoing traffic according to the given bandwidth, and how the congested router controls the received packets.

### 3.3.1 Bandwidth Allocation

Let $S_i$ be the path identifier representing source $AS_i$, or more precisely a path from the source $AS_i$ to a congested router, and $\mathcal{S}$ be the set of all active path identifiers seen at that congested router. Let $\lambda_{S_i}$ be the send rate of packets carrying $S_i$ and $C$ be the bandwidth of the congested link. Given $\lambda_{S_i}$ for $S_i \in \mathcal{S}$, bandwidth allocation to $S_i$, denoted by $C_{S_i}$, is made as follows.

$$C_{S_i} = \frac{C}{|\mathcal{S}|} + \frac{C(1 - \frac{1}{|\mathcal{S}|}\sum_{S_i \in \mathcal{S}} \rho_{S_i})}{|\mathcal{S}^H|}\mathcal{P}_{S_i} \qquad (3.1)$$

where $\rho_{S_i} = \min\{\frac{\lambda_{S_i}}{C_{S_i}}, 1\}$, $|\mathcal{S}^H|$ is the number of over-subscribing ASes (i.e., $\mathcal{S}^H = \{S_i | \lambda_{S_i} > \frac{C}{|\mathcal{S}|}, S_i \in \mathcal{S}\}$), and $\mathcal{P}_{S_i} = \min\{\frac{C_{S_i}}{\lambda_{S_i}}, 1\}$.

The term $\frac{C}{|\mathcal{S}|}$ of Eq. (3.1), which represents the bandwidth guaranteed to $S_i$, indicates that the same bandwidth is guaranteed to all ASes. The term $\frac{C(1-\frac{1}{|\mathcal{S}|}\sum_{S_i \in \mathcal{S}} \rho_{S_i})}{|\mathcal{S}^H|}\mathcal{P}_{S_i}$ of Eq. (3.1) represents the differential bandwidth reward given to $AS_i$ whenever it complies with rate-control requests *and* the bandwidths guaranteed to other ASes are not fully subscribed. Note that this reward is proportional to $S_i$'s rate-control compliance $\mathcal{P}_{S_i}$.

### 3.3.2 Source-end Packet Marking/Rate Limiting

A congested router sends a packet-marking request (which implicitly requires rate control) to the source/provider ASes whose transmission rate exceeds the allocated bandwidth (i.e., $\lambda_{S_i} > C_{S_i}$). The packet-marking request includes two threshold values: the guaranteed bandwidth ($\text{B}_{min} = \frac{C}{|\mathcal{S}|}$) and the allocated bandwidth ($\text{B}_{max} = C_{S_i}$). Upon receipt of the request, the egress router of the source AS (or ingress router of the provider AS) writes high priority markings (i.e., 0) on the packets at a rate of $\text{B}_{min}$ and low priority markings (i.e., 1) at a rate of $\text{B}_{max}$ – $\text{B}_{min}$. Also, it either drops the

remaining non-markable packets to comply with the rate-control policy of the destination or write lowest priority markings (i.e., 2) on them depending on the rate-control request parameters.

### 3.3.3 Rate Control at a Congested Router

A congested router allocates separate token buckets to path identifiers for per-AS bandwidth allocation. Each token bucket consists of two sub-buckets: a high-priority token bucket (denoted by $\text{HT}_{S_i}$) for bandwidth guarantee and a low-priority token bucket (denoted by $\text{LT}_{S_i}$) for bandwidth reward (viz., Figure 3). The router controls the bandwidth of each path identifier by applying the following packet admission policy. Let $Q(t)$ be the current length of the high-priority queue, and its (desired) normal operating range be $[Q_{min}, Q_{max}]$.[11]

A packet is placed in the high priority queue if its path identifier belongs to

1. *Legitimate Path* and
   - token is available in $\text{HT}_{S_i}$, or
   - token is available in $\text{LT}_{S_i}$ and $Q(t) \leq Q_{max}$, or
   - $Q(t) \leq Q_{min}$.
2. *Priority-Marking Attack Path* and
   - marking is 0 and token is available in $\text{HT}_{S_i}$, or
   - marking is 1, token is available in $\text{LT}_{S_i}$, and $Q(t) \leq Q_{max}$.
3. *Non-Marking Attack Path* and
   - token is available in $\text{HT}_{S_i}$.

Thus, $\text{HT}_{S_i}$ provides bandwidth guarantees to ASes, and $\text{LT}_{S_i}$ reallocates the residual bandwidth to legitimate and priority-marked ASes. However, a router grants the unused tokens of under-subscribing ASes only when $Q(t)$ remains in a normal operating range (i.e., $Q(t) \leq Q_{max}$). This allows the router to handle potential traffic increase from ASes that are allocated the same guaranteed bandwidth. Whenever $Q(t)$ drops below the minimum operating range (i.e., $Q(t) \leq Q_{min}$), the router enqueues the packets of legitimate paths regardless of token availability to avoid link under-utilization. Packets with the lowest priority marking (i.e., 2) are placed in the "legacy" queue (for non-prioritized traffic), which would be serviced only when the high-priority queue is empty. This bandwidth control mechanism is illustrated in Fig. 3.

## 3.4 Control Message Format

In this subsection, we describe the control message in detail. We show the message format in Fig. 4 and explain each field of the message below.

- $\text{AS}_S$: Source AS of the flows that need to be controlled.
- $\text{AS}_D$: Congested AS. The congested router sets this field to its ID (which is uniquely assigned within the domain) when sending a congestion notification message to the route controller. The route controller replaces the router ID with its own AS number before sending the control message to $\text{AS}_S$.
- Addr. Prefix: Destination address prefix(es) of the flows that contribute congestion. This field is set to null if no specific destination prefix is identified.

[11] $Q_{max}$ is chosen to limit the maximum queueing delay and $Q_{min}$ is chosen to allow instantaneous traffic burst.

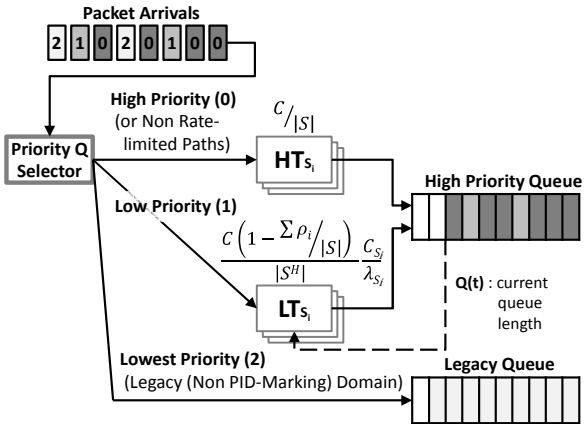



Figure 3: Bandwidth Control in a Congested Router.

| $\text{AS}_S$ | $\text{AS}_D$ | Addr. Prefix | Msg Type | Control Msg 1 | Control Msg 2 | TS | Duration | Sign |
|---|---|---|---|---|---|---|---|---|

Figure 4: Control Message Format.

- Msg Type: Control message type. Multi-path routing (MP), Path-pinning (PP), Rate-Throttling (RT), and Revocation (REV) messages are assigned one bit from the lowest bit.
- Control Msg 1 and 2: For each message type, the meanings of these fields (which are separated by a comma) are defined as follows.
  MP: ASes through which packets are routed ($\text{AS}_I^P$), ASes to be avoided ($\text{AS}_I^C$).
  PP: current AS path, unused
  RT: bandwidth guarantee $\text{B}_{min}^{th}$, bandwidth reward $\text{B}_{max}^{th}$ (viz., Section 3.3.2).
- TS: message creation time.
- Duration: Duration of the control message. Hence, TS + Duration is the expiration time of the control message.
- Sign: Digital signature on the control message.

In the control-message format, the fields $\text{AS}_S$, Addr. Prefix, and Control Msgs can have multiple entries, and hence the first byte of those fields is set to indicate the number of entries. The authenticity and integrity of inter-domain control messages (i.e., those between route controllers) are protected by a sender's digital signature on the control message, and those of intra-domain messages by a message authentication code (MAC) generated using the shared secret key between the route controller and individual routers of the same domain (viz., Section 3.1 for details).

# 4. EVALUATION

In this section, we illustrate the path diversity characteristics of the Internet that relevant to CoDef using real AS topology datasets, and evaluate the effectiveness of CoDef under various traffic scenarios using the ns2 simulator [31].

## 4.1 Path Diversity Analysis

### 4.1.1 Datasets

***Network topology.*** We construct the Internet topology using the CAIDA AS-relationships dataset[12] [5], which describes the AS-level connectivity based on relationships between ASes: provider, customer, peer or sibling. To determine a packet forwarding path, we assume that an AS applies the following rules in order. First, the AS prefers customer links over peer links and peer links over provider links. This preference comes from the fact that ASes are most interested in economic incentives in determining a forwarding path [14, 15]. Second, the AS prefers the shortest AS-path length route. Third, if multiple best routes exist, the AS uses AS number to break the tie.

To evaluate path diversity, we choose a sufficiently large number of attack ASes (see Attack distribution below) and find paths from those ASes to a selected target AS. Then, we test if the other ASes (i.e., non-attack ASes) have an alternate path to the target that does not include any AS found on the attack paths. In discovering alternate paths, we applied three AS exclusion policies explained below (viz., Section 4.1.2).

***Attack distribution.*** To select attack ASes out of more than 30,000 ASes, we use the Composite Blocking List (CBL) [4] that holds the list of IP addresses of spam bots, which could potentially be used for link flooding attacks. We cluster these IP addresses by their AS and select the top 538 ASes (each of which contains more than 1000 bots) as the attack ASes. We note that these top 538 ASes account for over 90% of 9 million bots found in the CBL. We use 6 different ASes, which host root DNS servers, as the attack targets. Since they have different distances from attack sources and widely different connectivity with other ASes (i.e., AS degrees), the alternate paths explored to those ASes would represent the path diversity accurately.

### 4.1.2 AS exclusion policies

We first identify the intermediate ASes located on attack paths toward a target AS, and remove them from the original topology, which we call AS exclusion. After AS exclusion, we find routes from non-attack ASes to the target, which would represent the alternate paths. For this purpose, we apply the following AS exclusion policies:

- ***Strict:*** All intermediate ASes on attack paths are excluded, hence every new path found from a non-attack AS to the destination would be disjoint from all attack paths. However, such disjoint paths would not exist if a source/target AS is single-homed, and cannot be easily found if their AS degree (i.e., the number of providers) is very low.

- ***Viable:*** All ASes on attack paths except the provider AS(es) of the target AS are excluded. With this policy, alternate paths are constructed between source ASes and the provider AS and the provider AS provides differential routing and rate-control services for its customer. This could be *viable* by contract between two ASes.

- ***Flexible:*** All ASes on attack paths except the provider ASes of the source and target ASes are excluded. That is, provider ASes at both end points provide services on behalf of their customers. Since provider ASes usually have a high AS degree, this policy is highly *flexible* in finding alternate paths.

[12]We use the dataset available for June 2012.

### 4.1.3 Alternate path discovery

Table 1 shows the evaluated path diversity in terms of the following metrics:

- ***Rerouting ratio:*** the proportion of rerouted ASes after an AS exclusion policy is applied.

- ***Connection ratio:*** the proportion of connected ASes; this includes the connected ASes through their original paths.

- ***Stretch:*** the average path-length increase of *rerouted paths*.

When the *strict* policy is applied, alternate paths are available for the high-degree target ASes (i.e., top three ASes in the table). The rerouting ratios of those ASes are above 60 %. Though AS 7500 has a lower AS degree than AS 20144 and AS 297, it has a similar rerouting ratio to those high degree ASes because the AS has a slightly longer average path length from other ASes. Obviously, longer paths could be rerouted more easily. The connection ratios of these ASes are slightly higher than their rerouting ratios since some ASes connect to the destination AS via non-attack paths (e.g., 1-hop neighbors). That is, the difference between the connection ratio and rerouting ratio indicates the proportion of ASes that have disjoint paths with the attack paths.

Application of the *viable* policy significantly increases the rerouting and connection ratios since provider ASes have much higher AS degrees than the target AS. However, AS 297 shows a different result: its rerouting ratio is decreased while the connection ratio is increased. This is because AS 297 has more *clean* paths that are not affected by the attack than other ASes. Obviously, these clean paths do not need to be rerouted. In this case, rerouting can be effective if it is performed by the provider of a source AS because the provider has a higher capability of rerouting traffic to a clean path than the source AS. The effectiveness of this rerouting is observed in the result as well: when the flexible policy is used, AS 297 has the highest rerouting ratio and connection ratio gains compared with the result of the viable policy, among top three ASes in the table. Meanwhile, low-degree ASes are still mostly disconnected under the viable policy because their providers (e.g., regional providers) are not connected to many different ASes. In this case, another provider AS' rerouting (i.e., the flexible policy) is necessary and becomes very effective as the result shows.

Finally, when the *flexible* policy is applied, a large fraction of ASes can be connected to the destination AS and have alternate paths. Consequently, with mid-size provider ASes' participation (i.e., the provider of stub-ASes), rerouting becomes highly effective. The stretch does not change much under different policies. Hence, the additional path delay caused by rerouting would not be significant in most cases, or the path delay can be decreased if the original path has a high queueing delay due to the high volume of traffic.

## 4.2 Traffic-Control Simulation

The simulations of this subsection evaluate the effectiveness of CoDef for various types of background traffic (e.g., FTP, Constant Bit Rate (CBR), and Web traffic) for realistic simulation. The simulations are run in a conservative scenario; i.e., a legitimate AS has only a single alternate path to the destination. (Note that the availability of multiple alternate paths is validated by the results of the previous Section 4.1.) Our simulations are extensive in the sense that they utilize all the resources (e.g., number of nodes) allowed by the ns2 simulator [31]. Through these simulations, we illustrate (1) how legitimate flows are protected from an attack by conforming to rerouting and bandwidth-control requests, and (2) how attack

| Target | Path Length | AS Degree | Rerouting Ratio | | | Connection Ratio | | | Stretch | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Strict | Viable | Flexible | Strict | Viable | Flexible | Strict | Viable | Flexible |
| AS 20144 | 3.94 | 48 | 63.31 | 81.90 | 87.21 | 64.42 | 88.59 | 96.32 | 0.53 | 0.40 | 0.41 |
| AS 297 | 3.90 | 34 | 64.23 | 61.70 | 81.06 | 64.25 | 74.17 | 96.56 | 1.44 | 1.43 | 1.34 |
| AS 7500 | 4.15 | 19 | 63.30 | 82.52 | 92.79 | 63.43 | 84.24 | 95.48 | 0.62 | 0.47 | 0.49 |
| AS 27 | 5.08 | 3 | 0.00 | 0.00 | 61.02 | 0.00 | 0.18 | 68.35 | 0.00 | 0.00 | 1.39 |
| AS 2149 | 4.41 | 1 | 0.00 | 0.00 | 44.14 | 0.00 | 8.27 | 86.05 | 0.00 | 0.00 | 0.46 |
| AS 29216 | 4.90 | 1 | 0.00 | 0.00 | 58.12 | 0.00 | 0.08 | 68.66 | 0.00 | 0.00 | 0.56 |

Table 1: Path Diversity in the Internet.

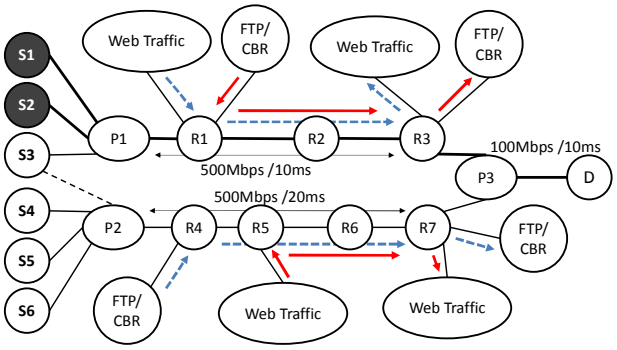



Figure 5: Simulation Topology

flows are penalized in terms of bandwidth allocations. For long TCP flows, we use the flow bandwidth as an evaluation metric to illustrate system resilience under attack. For short flows (i.e., HTTP flows), we use the finish time (or completion time) distribution of flows as an evaluation metric because those flows have different numbers of packets to transmit (i.e., some of flows would disappear very quickly while others last longer, which naturally leads to bandwidth fluctuation), and hence their bandwidth may not capture the attack effects precisely.

**Topology.** The network topology shown in Fig. 5 is used to simulate a conservative scenario of CoDef operation; i.e., the legitimate AS (S3) has *only one* alternate path (S3-P2-R4-R5-R6-R7-P3-D) when a link-flooding attack occurs on its initial path (S3-P1-R1-R2-R3-P3-D) to destination. Source ASes (S1,. . .,S6) and a destination AS (D) are attached to provider ASes (P1, P2, P3). Intermediate ASes (R1,. . .,R7) connect provider ASes via two disjoint paths. In the simulated topology, S3 is connected to multiple providers (i.e., P1, P2). However, it uses P1 as the default next-hop AS to the destination since the upper path (P1-R1-R2-R3-P3) has shorter path length. Each AS is represented by a single router in our simulation. The lower path is 1-hop longer than the upper path, which corresponds to the stretch of alternate paths presented in the previous section. All link delays of the lower path are set to twice the delay of most upper paths to take into account the paths that have a higher stretch.

**Traffic.** To approximate real network-traffic conditions, we use Web packet arrivals with a Pareto distribution, FTP, and CBR traffic as background traffic. We configure 300 Mbps Web traffic, 50 Mbps CBR traffic, and FTP flows to pass through the core network (i.e., $R_i$'s) as shown in Fig. 5 . We attach 30 FTP sources to each of source ASes as legitimate flows which sends 5 MB files to the destination D. Then, we measure the flows' bandwidth at the (attack) target link. As long TCP flows are most vulnerable to link flooding attacks (due to the TCP congestion control mechanism), their bandwidth at the congested link would reflect the worst effect of flooding attacks. We also simulate the effects of attacks on Web traffic using the PackMime package [10] which generates synthetic Web traffic.

### *4.2.1 Persistent TCP Flows*

We place attack sources at two ASes, namely S1 and S2, and configure them to send 300 Mbps Web traffic each. This attack traffic is sufficient to exhaust both the bandwidth of the target link (i.e., 100 Mbps) and the available bandwidth of intermediate links to TCP flows (i.e., 150 Mbps). We configure S5 and S6 to send 10 Mbps traffic and observe how the under-subscribed bandwidth by those ASes is reallocated to other ASes. The congested router (i.e., P3) performs per-path fair bandwidth control using a token bucket mechanism [20], where individual paths are assigned separate token buckets. The remaining routers operate drop-tail queues to model the legacy part of the Internet. To show the advantage of differential bandwidth allocation to rate-controlling source ASes, S2 is configured to control the rate of outgoing traffic as requested by the congested router.

Fig. 6 shows the bandwidth used by individual source ASes for the following scenarios: S3 uses (1) default path, (2) alternate path and (3) alternate path while per-path bandwidth control is performed by all routers on the path. These scenarios are tested with the rate of 200 Mbps and 300 Mbps attack traffic from each attack AS. When S3 uses its default path (viz., SP-200 and SP-300 in Fig. 6) , its TCP flows are significantly affected by the high-rate traffic of S1 before those flows arrive at P3, though P3 limits the bandwidth of S1 via per-path bandwidth control. Meanwhile, the rate-controlling AS S2 uses higher bandwidth than S1 since the under-subscribed bandwidth by S5 and S6 (i.e., 33.4 Mbps - 20 Mbps = 13.4 Mbps) is reallocated to S2, S3 and S4.

When S3 forwards its traffic via P2 (viz., MP-200 and MP-300 in Fig. 6), the bandwidth used by S3 increases as much as that of S4, though the selected path has more hops and link delays to the destination. We also evaluate the effectiveness of multi-path routing by comparing this result with that of global deployment of the per-path bandwidth control mechanism. MPP-200 and MPP-300 in Fig. 6 show that global per-path (fair) bandwidth control allows higher bandwidth to legitimate ASes (S3, S4) since it handles instantaneous bursts of background traffic near their origin. This is better illustrated in Fig. 7, which compares the bandwidth used by S3 over time under different forwarding and bandwidth control scenarios. However, when we consider that global bandwidth control cannot be easily accomplished, especially at the Internet backbone where a large number of flow paths exist, multi-path routing would be an effective way to mitigate the effects of flooding attacks as it produces comparable result with that of global bandwidth control.

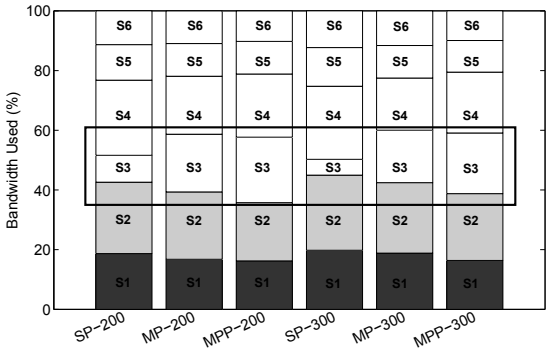



Figure 6: Bandwidth used by source ASes at the congested link. Legend: SP: Single-path Routing, MP: Multi-path Routing, MPP: MP with global per-path bandwidth control. The number that follows a dash represents the send-rate of each attack AS in Mbps.

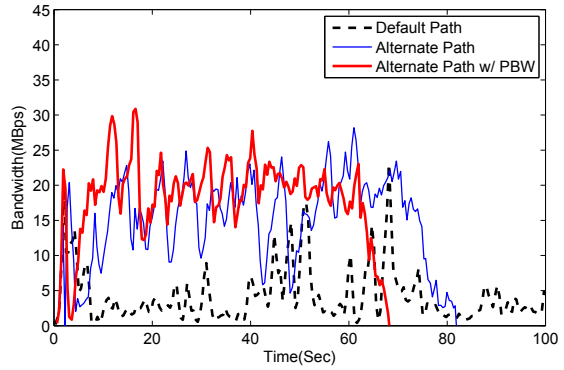



Figure 7: Bandwidth used by S3. PBW stands for the path bandwidth control.

### 4.2.2 *Web Traffic*

Next, we simulate the effects of attacks on Web Traffic. For this simulation, we attach a server cloud (a set of servers implemented in the PackMime-HTTP package [10]) to S3 and a client cloud (a set of clients) to D; and configure them to establish 200 new connections per second so that S3 sends sufficient traffic for the allocated bandwidth to it. The connection-request times and file sizes follow the Weibull distribution [10].

Fig. 8 shows the results of three different simulation scenarios where both the horizontal and vertical axes are plotted in a log scale: *no attack* 8(a), *attack with single-path routing* 8(b), and *attack with multi-path routing* 8(c). When S3 uses its default path in the presence of attacks, the finish times increase significantly over the entire range of file sizes and have a wide variance for a given file size. In contrast, when S3 selects an alternate path, the finish times of flows exhibit a similar distribution as that of non-attack scenario, though they move upward slightly due to the increased path delay. Fig. 8 also shows that, whenever the default path is used, the finish time increases significantly as the file size grows. This is because long TCP flows become more vulnerable if packets drop, as already explained. Multi-path routing is highly effective in protecting those long TCP flows as their bandwidths are primarily determined by link congestion (i.e., packet drops) rather than by path delay.

# 5. RELATED WORK

## 5.1 Anti-DDoS Mechanisms in Practice

Most currently deployed defense mechanisms protect *single* enterprise servers or their network links to the Internet against the server-flooding attacks. However, when attacks targeting an enterprise flood selected upstream network links to degrade a server's connectivity to the Internet server, traditional enterprise defense mechanisms, such as firewalls or intrusion detection/protection systems, can no longer protect the server. These attacks could only be mitigated upstream from the enterprise; i.e., the ISP or content distribution networks (CDNs) [8]. However, neither of these defenses can be used to counter large-scale, link-flooding attacks to date.

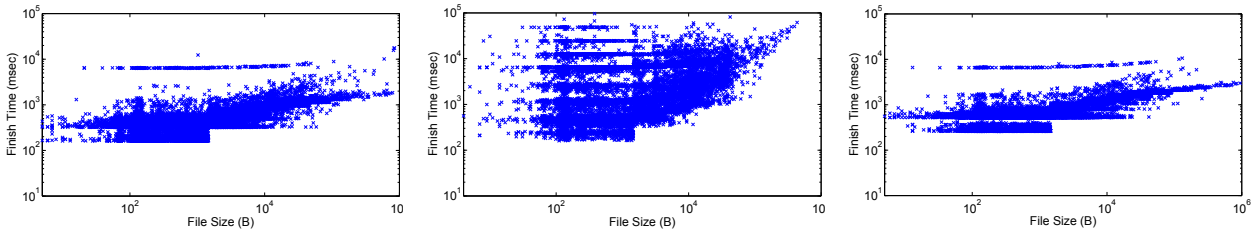
## 5.2 Traffic Filtering Mechanisms

Most of previous countermeasures to link-flooding attacks are filtering-based mechanisms that identify attack flows or flow aggregates and install filters at the target and upstream routers to block excess traffic. With *pushback* [16], a target router identifies flow-aggregates that contribute to its flooding and asks the upstream routers to install filters to block high-rate flows. Furthermore, the upstream routers recursively perform pushback for other upstream routers and eventually the routers directly serving the attack sources install filters. More recent proposals in [13, 19] suggest new methods to directly install filters at the ASes that originate attack traffic.

Though filtering-based approaches might be effective to defend against traditional attacks, they cannot be used to mitigate the large-scale, link-flooding attacks using low-rate attack flows. In fact, filtering-based approaches always pose a high risk of collateral damage to legitimate flows, when used to mitigate such attacks. Moreover, filtering-based mechanisms have not widely deployed mainly because they fail to provide incentives for upstream ASes to install filters for the benefit of downstream ASes. In contrast, CoDef does not install filters, thereby avoiding that source of collateral damage, even though it is capable of distinguishing attack flows from legitimate flows quite accurately. Furthermore, CoDef provides immediate incentives for source/provider ASes to collaborate with the target ASes in defense againt link flooding; e.g., better quality of service to premium customers even during persistent attacks.

## 5.3 Multi-path Routing Mechanisms

A variety of multi-path routing mechanisms have been proposed to decrease or avoid congestion by forwarding traffic through multiple paths [6,9,25,33,36]. These mechanisms, though they differ in implementation details (e.g., overlay routing, inter-domain routing or intra-domain routing), rely on path diversity to disperse traffic by having individual routers make routing decision based on local congestion information. However, path diversity alone is fundamentally insufficient for handling large-scale flooding attacks for two reasons. The first is the attack-defense scaling asymmetry: an attack launched by distributed bots can scale faster and more inexpensively than multiple path provisioning. Hence, in the absence of additional mechanisms, such as flow distinguishability whereby legitimate and attack flows are separated and then handled (e.g., routed) differently, multi-path routing cannot provide adequate defense. The second reason is a multi-path defense that only diverts traffic on different paths without distinguishing flows would merely disperse attack flows. Hence, attack flows will continue to affect legitimate flows (e.g., decrease available bandwidth) and aggregate downstream thereby making it harder to identify and handle.




(a) No Attack (b) Default Path (c) Alternate Path

Figure 8: File Size vs. Finish Time.

## 6. CONCLUDING REMARKS

In this paper, we presented a complementary routing system that reroutes legitimate traffic via less congested paths using the path diversity of the Internet. Meanwhile, pinning the paths that deliver attack traffic prevents attack dispersion, hence localizes their effects. Such differential routing is implemented by slightly modifying the current routing policies, in order to make them easily adopted. In addition to those routing policies, the bandwidth reward policy provides incentive to source-end defenses which are essential for reducing the collateral damage of flooding attacks to legitimate traffic. By evaluating the path diversity of the current Internet, we show that rerouting would be not only feasible but can be effectively implemented at stub or provider ASes. Simulation results illustrate the effects of flooding attacks on the legitimate flows and how those effects are mitigated by CoDef.

## 7. ACKNOWLEDGMENT

We are grateful to Sharon Goldberg and the referees for their insightful comments and suggestions. The first and third authors' research was supported in part by the US Army Research Office under Contract W911NF-07-1-0287 at the University of Maryland and by the Northrop Grumman Corporation at CyLab, Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Office, the U.S. Government, Northrop Grumman Corporation or Carnegie Mellon University.

## 8. REFERENCES

[1] Internet-Exchange Point, `http://www.bgp4.as/internet-exchanges`.

[2] Multi-Topology Routing, `http://www.cisco.com/en/US/docs/ios/12_2sr/12_2srb/feature/guide/srmtrdoc.html`.

[3] `http://arstechnica.com/security/2013/04/can-a-ddos-break-the-internet-sure-just-not-all-of-it/`.

[4] `http://cbl.abuseat.org/`.

[5] `http://www.caida.org/data/active/as-relationships/`.

[6] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. In *SOSP '01:*, New York, NY, USA, 2001. ACM.

[7] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable Internet Protocol (AIP). In *Proc. ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[8] Arbor Networks. Whitepaper: DDoS Attack Tools and Best-Practices for Defense, 2012.

[9] K. Argyraki and D. R. Cheriton. Loose source routing as a mechanism for traffic policies. In *FDNA '04*.

[10] J. Cao, W. S. Cleveland, Y. Gao, K. Jeffay, F. D. Smith, and M. Weigle. Stochastic models for generating synthetic http source traffic. In *INFOCOMM*, 2004.

[11] M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian. Fabric: a retrospective on evolving sdn. In *Proceedings of HotSDN*. ACM, 2012.

[12] Cisco. BGP best path selection algorithm: How the best path algorithm works. Document ID: 13753, May 2012.

[13] Daniel R. Simon and Sharad Agarwal and David A. Maltz. AS-Based Accountability as a Cost-effective DDoS Defense. *HotBots '07*, 2007.

[14] L. Gao, T. G. Griffin, and J. Rexford. Inherently safe backup routing with bgp. In *IEEE INFOCOM 2001*, volume 1, pages 547–556, 2001.

[15] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford. How secure are secure interdomain routing protocols. In *ACM SIGCOMM Computer Communication Review*, volume 40, pages 87–98. ACM, 2010.

[16] John Ioannidis and Steven M. Bellovin. Implementing Pushback: Router-Based Defense Against DDoS Attacks. In *NDSS*, 2002.

[17] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: responsive yet stable traffic engineering. In *SIGCOMM '05*, 2005.

[18] M. S. Kang, S. B. Lee, and V. D. Gligor. The Crossfire Attack. In *Proceedings of IEEE Symposium on Security and Privacy*, 2013.

[19] Katerina Argyraki and David R. Cheriton. Active internet traffic filtering: real-time response to denial-of-service attacks. In *ATEC '05*.

[20] S. B. Lee and V. Gligor. FLoc : Dependable link access for legitimate traffic in flooding attacks. In *The 30th International Conference on Distributed Computing Systems*, 2010.

[21] S. B. Lee, V. D. Gligor, and A. Perrig. Dependable connection setup for network capabilities. In *IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2010.

[22] M. Lepinski and S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, Feb. 2012.

[23] X. Liu, A. Li, X. Yang, and D. Wetherall. Passport: Secure and adoptable source authentication. In *NSDI*, volume 8, pages 365–378, 2008.

[24] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow:

enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[25] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala. Path splicing. In *SIGCOMM*, pages 27–38, 2008.

[26] B. Parno, A. Perrig, and D. Andersen. SNAPP: Stateless network-authenticated path pinning. In *Proceedings of the ACM ASIACCS*, 2008.

[27] P. Psenak, S. Mirtorabi, A. Roy, L. Nguyen, and P. Pillay-Esnault. RFC-4915: Multi-Topology (MT) Routing in OSPF. 2007.

[28] Ratul Mahajan and Steven M. Bellovin and Sally Floyd and John Ioannidis and Vern Paxson and Scott Shenker. Controlling high bandwidth aggregates in the network. *SIGCOMM Comput. Commun. Rev.*, 32(3):62–73, 2002.

[29] E. Rosen, A. Viswanathan, and R. Callon. Rfc-3031: Multiprotocol label switching architecture. 2001.

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

[31] USC/ISI, Network Simulator 2 (NS2). `http://www.isi.edu/nsnam/ns/`.

[32] Xiaowei Yang and David Wetherall and Thomas Anderson. A DoS-limiting network architecture. In *SIGCOMM '05*, 2005.

[33] W. Xu and J. Rexford. Miro: multi-path interdomain routing. In *SIGCOMM '06*, pages 171–182, 2006.

[34] A. Yaar, A. Perrig, and D. Song. SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. In *Proceedings of the IEEE Security and Privacy Symposium*, 2004.

[35] H. Yan, D. A. Maltz, T. E. Ng, H. Gogineni, H. Zhang, and Z. Cai. Tesseract: A 4D network control plane. In *Proc. NSDI*, 2007.

[36] X. Yang and D. Wetherall. Source selectable path diversity via routing deflections. In *SIGCOMM '06*.

[37] 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 *IEEE Symposium on Security and Privacy*, 2011.