# Tracking Down Sources of Spoofed IP Packets

Osvaldo Fonseca[†] Ítalo Cunha[†] Elverton Fazzion[†‡] Wagner Meira Jr.[†]
Brivaldo Junior[⋆] Ronaldo A. Ferreira[⋆] Ethan Katz-Bassett[♯]

[†]Universidade Federal de Minas Gerais, Brazil [‡]Universidade Federal de São João del-Rei, Brazil
[⋆]Universidade Federal de Mato Grosso do Sul, Brazil [♯]Columbia University

***Abstract*—The lack of authentication in the Internet's data plane allows hosts to falsify (*spoof*) the source IP address in packet headers, which forms the basis for amplification denial-of-service (DoS) attacks. Current approaches to locate sources of spoofed traffic lack coverage or are not deployable today. We propose a mechanism that a network with multiple peering links can use to coarsely locate the sources of spoofed traffic in the Internet. More precisely, the network can monitor and map spoofed traffic arriving on a peering link to the set of sources routed toward that link. We propose mechanisms the network can use to systematically vary BGP announcement configurations to induce changes to Internet routes and to the set of sources routed to each peering link. A network using our technique can correlate observations over multiple configurations to more precisely delineate regions sending spoofed traffic. Evaluation of our techniques on the Internet shows that they can partition the Internet into small regions, allowing targeted intervention.**

***Index Terms*—IP spoofing, security, amplification, denial-of-service, routing policies, topology discovery**

## I. Introduction

The lack of authentication in the Internet's data plane allows hosts to falsify (*spoof*) the source IP addresses of their traffic and send unsolicited traffic to arbitrary destinations. These vulnerabilities form the basis for amplification denial-of-service attacks [1], which have been effectively employed against large-scale distributed service providers (e.g., [2]). The spoofed source addresses make the origins of such attacks seemingly untraceable, complicating attribution, mitigation efforts to squelch the attack, or targeted efforts to convince networks to disallow spoofed traffic.

Over the last two decades, researchers have proposed dozens of *IP traceback* techniques for identifying the routes taken by spoofed packets [3]–[9]. Approaches include temporarily congesting links to perturb (attack) traffic, modifying routers to encode information (usually in the IP ID field) about routers traversed by a small fraction of packets, modifying routers to send information about a fraction of forwarded packets towards destinations, or modifying routers to store packet digests and provide an interface for querying for a packet's signature. Despite all the research, none of these approaches has been deployed and increased our ability to locate the origins of spoofed traffic, because they require changes to routers, cooperation from other networks, and wide deployment to provide accurate identification. Since these techniques face nearly insurmountable barriers to adoption, today's networks get a single data point on the spoofed traffic's route: which peering link receives the traffic.

In this paper, we explore how a network can manipulate this information source—the peering link where traffic ingresses a network—to more precisely locate sources of spoofed traffic. Our key observation is that the routes are partially under an origin network's control, and so the network receiving the spoofed traffic has some ability to impact on which link it receives traffic, instead of relying on routers that are not under its control. We propose techniques that are fundamentally different from existing traceback approaches and can be used today, requiring no changes to deployed equipment nor cooperation from other networks. Our techniques work best when the spoofed traffic originates from few sources, as is common in amplification DoS attacks [10].

With our approach, a network announces an IP prefix through multiple peering links, a practice known as *anycast*. Each link attracts traffic from non-overlapping regions of the Internet called the link's *catchment*. The network can infer the sources in each catchment by inspecting non-attack traffic at each ingress link and mapping the source IP addresses to their respective prefixes and controlling autonomous systems (ASes), or by sending out pings and measuring which link replies arrive at [11]. To measure the amount of spoofed traffic on each link, the network can run an amplification honeypot that does not receive legitimate traffic (e.g., AmpPot [10]) or infer the set of valid source addresses from each peering link and label the traffic from other addresses as spoofed [12], [13]. The amount of spoofed traffic arriving at each peering link can then be attributed to the sources routed toward that link. Many sources are routed toward the same peering link, however, so simply attributing an attack traffic volume to a peering link is not precise enough to isolate attack sources.

To track down sources of spoofed traffic, we present systematic approaches to vary IP prefix announcement configurations that allow networks to induce changes to routes toward their prefixes and, more importantly, in the set of ASes routed toward each peering link (the catchment). Networks using our techniques can correlate spoofed traffic observed from sets of sources across multiple announcement configurations to infer regions of the Internet sourcing spoofed traffic. Figure 1 provides intuition for how such measurements can be combined to identify networks that allow spoofed packets:

- In configuration 1, the operator announces a prefix through three peering links with networks $m$, $n$, and $p$;

ISBN 978-3-903176-28-7 © 2020 IFIP

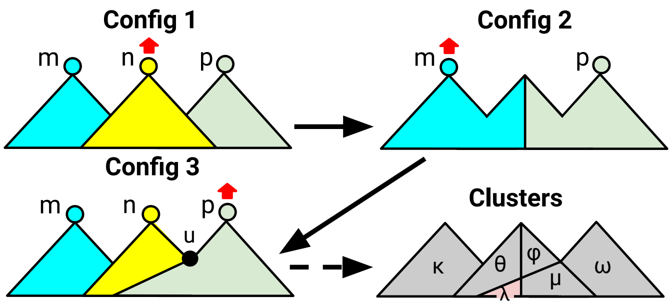

Figure 1: Example with catchments and resulting clusters for three announcement configurations performed by an origin network peering with ASes $m$, $n$, and $p$.

measures the catchment (colored polygons) and traffic arriving on each peering link; and identifies that the spoofed traffic is concentrated on the link with $n$, i.e., sent by networks in $n$'s catchment (red arrow).

- The operator later withdraws the announcement to $n$ (configuration 2), measures catchments and traffic volumes again, and identifies that the spoofed traffic is now concentrated on the peering link with $m$.
- Configuration 3 announces the prefix from $n$ again, but poisoning AS $u$ (which causes AS $u$ to ignore the route from $n$ and choose the route from $p$ instead). The operator can measure catchments and traffic to identify that the spoofed traffic is concentrated on the peering link with $p$.
- Finally, the operator can intersect the measured catchments to partition networks into *clusters* (bottom right), and correlate clusters with observed spoofed traffic (red arrows) to identify that the spoofed traffic is concentrated on networks comprising $\lambda$.

We evaluate our techniques running experiments on the PEERING platform [14]. We deploy 705 different announcement configurations from seven peering links, identifying multiple, different routes from each source covered in our measurements. We show that correlating information across multiple announcement configurations on PEERING allows us to partition the Internet into small regions with as few as one AS and with 1.40 ASes on average. The small size make these regions candidate targets for countermeasures or notifications. Our results indicate that networks with peering footprints larger than PEERING's, as will be the case for most regional transit networks, can more effectively manipulate routes to achieve even higher accuracy and quicker localization.

Our techniques allow identification of networks that do not employ BCP38 (ingress filtering) [15] and allow spoofed traffic, helping Internet bodies focus efforts and drive adoption of best practices. They can also be used to drive automatic DoS mitigation systems that use, e.g., BGP communities to trigger remote traffic blackholing [16] or BGP flowspec to configure traffic filters [17].

## II. Background on BGP

The BGP best-path selection algorithm defines the preferred route to an IP prefix as the route with the highest *local preference* (LocalPref), a value set by the AS according to private routing policies. If multiple routes have the same LocalPref, BGP chooses the route with the shortest AS-path length. If multiple routes remain tied for best, BGP applies other tiebreakers that include intra-domain (IGP) routing costs, hints received from neighboring ASes (MED), and route age (to reduce oscillations) [18].

An AS that controls an IP prefix can configure its BGP announcements to influence routes, e.g., to achieve traffic engineering goals [19], [20]. First, an AS can announce (anycast) an IP prefix from all or a subset of its peering links. This strategy is used by content distribution networks so remote ASes, and users therein, route to a topologically close location, improving performance and increasing reliability [19], [21]. Second, an AS can influence BGP's tie breaking at remote ASes by *prepending* its AS number to the announcement's AS-path, making the AS-path artificially longer. This strategy is used by multihomed ASes to signal on which link it prefers to receive traffic [20], [22]. Third, an AS can influence the use and propagation of its announcements through a remote AS using BGP *poisoning* [18], [23]–[25]. A poisoned announcement targets one or more ASes, and includes the target ASes' numbers in the AS-path; this triggers loop prevention and causes poisoned ASes to ignore the announcement.

## III. Locating Sources of Spoofed Traffic

We define an *announcement configuration* for an IP prefix as a triple $c = \langle \mathcal{A}_c; \mathcal{P}_c; \mathcal{Q}_c \rangle$. We denote the set of peering links of an origin AS by $\mathcal{L}$. $\mathcal{A}_c \subseteq \mathcal{L}$ is the set of locations from which the prefix is announced. Each location in $\mathcal{A}_c$ announcing the prefix will attract traffic from non-overlapping regions of the Internet that we call a *catchment*. $\mathcal{P}_c \subseteq \mathcal{A}_c$ is the set of locations where the prefix is announced with prepending, and $\mathcal{Q}_c$ is a mapping from announcement locations in $\mathcal{A}_c$ to sets of poisoned ASes. We drop the subscripts when the configuration is clear from context. For example, consider an AS with four peering links labelled from $l_1$ to $l_4$. A configuration $c = \langle \{l_1, l_2\}; \{l_1\}; \{l_1 : \emptyset,\, l_2 : \{a, b\}\} \rangle$ means the prefix is announced through peering link $l_1$ with AS-path prepending, announced through peering link $l_2$ poisoning ASes $a$ and $b$, and not announced through links $l_3$ and $l_4$.

### A. Systematic Route Changes

We propose a method that an *origin AS* with multiple peering links can use to generate announcement configurations that systematically induce route and catchment changes.

*a) Varying announcement locations:* Announcing a prefix from more peering links increases route diversity and leads to smaller catchments, on average. Smaller catchments provide better localization of spoofed traffic.

We propose that the origin AS deploy a sequence of configurations starting by announcing from all available peering links, i.e., $\mathcal{A} = \mathcal{L}$; then make announcements from all proper subsets of available locations $\mathcal{L}$ in decreasing size order. Deploying all configurations removing up to $r$ links from $\mathcal{L}$ is guaranteed to discover *at least* $r + 1$ routes for *all* sources in the Internet.

Whenever we withdraw the prefix from the peering link a source is routed to, that source will need to be routed to an alternate link. This is a deterministic way to uncover route diversity that scales with a network's peering footprint (i.e., the size of $\mathcal{L}$).

In Figure 1, Configuration 1 shows catchments when the origin AS announces a prefix through three peers: $m$, $n$, and $p$; Configuration 2 shows catchments when the origin AS announces through $m$ and $p$ only.

*b) Varying the AS-path length with BGP prepending:* For any given announcement configuration, a router may have multiple routes with the same LocalPref to choose from. In these cases, the router chooses the preferred route based on the AS-path length or subsequent BGP tiebreakers.

Given a configuration with a set of announcement locations $\mathcal{A} \subseteq \mathcal{L}$, we propose that the origin AS generate and deploy additional configurations prepending announcements from subsets of locations $\mathcal{P} \subseteq \mathcal{A}$, in increasing size order. To make prepended routes longer than most other routes, the origin can prepend its AS number four times, which is longer than most AS-paths in the Internet [26]. Deploying configurations that prepend announcements from all combinations of up to $s$ locations induces BGP's tie-breaking mechanism to choose up to $s$ alternate routes. More precisely, prepending will cause a router to change away from its (previously shorter and preferred) route whenever an alternate route with the same LocalPref and no prepending is available.

Manipulating BGP tiebreakers like the AS-path length is a general idea. Unfortunately, BGP tiebreakers after the AS-path length cannot be controlled (e.g., IGP costs) or do not propagate to distant ASes (e.g., MED), and thus cannot be employed by the origin for route manipulation.

*c) Controlling route propagation with BGP poisoning:* BGP AS-path prepending is ineffective when routers choose routes based on LocalPref, i.e., before applying BGP tiebreakers. In these cases, the origin AS can still try to induce route changes by making a remote router's preferred route (with highest LocalPref) unavailable using BGP poisoning. The origin AS can try to induce routers in a remote AS $r$ to change routes by poisoning $r$ (or other intermediate ASes between itself and $r$) in some announcements. Target ASes for BGP poisoning can be chosen using different strategies depending on the goal [18], [23]–[25].

We propose a specific targeting strategy that attempts to induce large sets of ASes to change routes. Figure 2 illustrates how the strategy works. Suppose the origin AS $o$ is directly connected to neighboring ASes $x$, $n$, and $y$. Figure 2a shows the routes used by each AS to reach AS $o$ when the origin $o$ anycasts the prefix to all neighbors without poisoning. Figure 2b shows the routes when the origin $o$ poisons AS $u$ on announcements through link $o$–$n$. Poisoning an upstream AS $u$ that is a neighbor of AS $n$ will prevent routes (and traffic) from traversing the link $n$–$u$ (red *X* in Figure 2b), causing routing changes at all sources previously routed through link $n$–$u$. Therefore, ASes $a$, $b$, $c$, and $u$ need to find an alternate path to reach AS $o$ (dashed lines). Configuration 3 in Figure 1 illustrates the change in catchments when we poison AS $u$ on the announcement through $n$. We use BGP poisoning to move traffic away from links that are close to (1 AS-hop away from) the origin AS, as those links are used to route more sources toward the origin AS than links that are farther away, thus inducing a higher number of path changes.

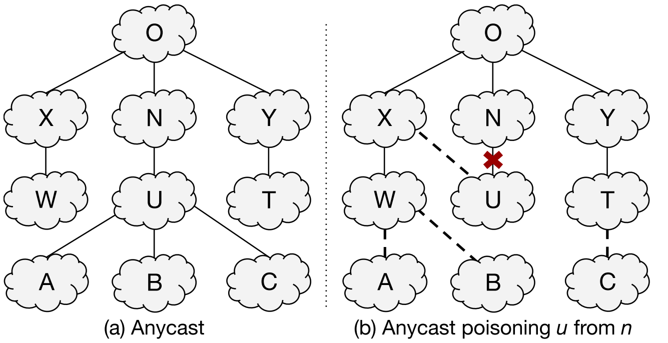

Figure 2: Example of AS $o$ poisoning AS $u$ through link $o$–$n$ to force all ASes previously routing through link $n$–$u$ to choose a different route.

This approach is similar and complementary to our proposal to control announcement locations (choosing $\mathcal{A}$), as it attempts to control route propagation through a directly-connected network's links. Unfortunately, BGP poisoning may be ineffective: an AS may disable BGP loop prevention for traffic engineering, e.g., when interconnecting multiple sites over the Internet by announcing different prefixes from each site; and ASes may filter poisoned announcements, e.g., tier-1 ASes often filter announcements from clients whose AS-path contains other tier-1 ASes, as such announcements normally indicate a route leak [27]. As a result, we use BGP poisoning as a best-effort approach to complement the previous two techniques, which are more reliable.

## B. Correlating Observations

We define a *cluster* as a set of sources that are in the same catchment across all announcement configurations. We start by placing all sources into a single cluster. We iterate over all catchments in all configurations; for each catchment $\alpha$ we iterate over all clusters $\kappa$ identified so far and split any cluster $\kappa$ that overlaps $\alpha$ into up to two clusters: $\kappa \cap \alpha$ and $\kappa \setminus \alpha$ (we do not split $\kappa$ if $\kappa \cap \alpha = \kappa$). The bottom right corner of Figure 1 shows the clusters obtained after performing the three announcement configurations.

Our techniques generate different announcements to induce route changes with the goal of reducing the size of clusters. Small clusters allow the identification of networks responsible for sending spoofed packets and enable targeted intervention.

## C. Estimating Volume of Spoofed Traffic

An origin AS can estimate the presence or volume of spoofed traffic received on each catchment by hosting a honeypot that emulates a service vulnerable to (but that does not contribute to) amplification attacks to attract spoofed traffic [10]. Another approach is to infer legitimate sources

Table I: PoPs and providers of the PEERING platform used in the experiments.

| Mux | Transit Provider |
|---|---|
| AMS-IX | Bit BV (AS12859) |
| GRNet | GRNet (AS5408) |
| USC/ISI | Los Nettos (AS226) |
| NEU | Northeastern University (AS156) |
| Seattle-IX | RGnet (AS3130) |
| UFMG | RNP (AS1916) |
| UW | Pacific Northwest GigaPoP (AS101) |

for each peering link and label all traffic received from other sources as spoofed [12], [13].

## IV. Experimental Setup

We evaluate our techniques in the Internet by making announcements from the PEERING platform [14]. PEERING is a research platform that operates an AS with multiple points-of-presence (PoPs) in various locations spread across three continents. We make announcements from seven PEERING PoPs, using one provider at each PoP. At PEERING PoPs at IXPs, which have multiple providers and peers, we choose one provider and use it throughout the experiment. Table I summarizes information about the PoPs and providers we used. We next describe how we generate configurations using our techniques and how we measure catchments.

*a) Announcement configurations:* BGP convergence delay plus the time required to measure catchments (§III-B) implies an origin AS cannot change configurations frequently and limits the rate at which catchments can be discovered. Here we discuss how we use our techniques to maximize information from a limited number of configurations.

We start with a configuration that announces (anycasts) a prefix to one transit provider in each of 7 active PEERING PoPs. Given that the number of announcements required by our first technique to discover $r$ routes from each AS on the Internet grows exponentially, we limit $r$ to 4, which requires $\sum_{x=0}^{3}\binom{7}{7-x} = 64$ configurations. For each such configuration $c$, we generate an additional $|\mathcal{A}_c|$ configurations, prepending from each active location in turn. This requires an additional $\sum_{x=0}^{3}[7-x]\binom{7}{7-x} = 294$ configurations. Finally, we identify 347 neighbors of PEERING's directly-connected transit providers using a combination of CAIDA's AS-relationship database [28], our own traceroute measurements, and public BGP feeds from RouteViews [29] and RIPE RIS [30]. We then generate 347 additional configurations, each announcing from all locations but poisoning one neighbor of a transit provider (on the announcement through that transit provider). In total, we generate 705 configurations.

*b) Measuring catchments:* PEERING prefixes carry no production traffic, so we cannot passively observe traffic to infer catchments. Also, concerns about executing Internet-wide scans from the PEERING platform limits our ability to issue measurements from the platform to the wide-area Internet. Instead, we measure catchments using a combination of AS-paths observed on BGP update messages towards PEERING prefixes collected from public feeds and traceroutes issued from RIPE Atlas toward PEERING prefixes [31]. We use all public BGP feeds from RouteViews [29] and RIPE RIS [30]. We partnered with RIPE and received permission to issue traceroute measurements every 20 minutes from 1600 RIPE Atlas probes, 7x more measurements than normally supported (our probing rate is still low, only targets PEERING prefixes, and has not raised complaints). Our dataset covers 1885 ASes, including all Tier-1 ASes and 73% of ASes with customer cone larger than 300 ASes [28].

We keep each announcement configuration active for 70 minutes to wait for route convergence and ensure, with high probability, that we collect at least three rounds of traceroutes *after* routes to our prefixes have converged, as convergence takes less than 2.5 minutes 99% of the time [25].

We map traceroute hops into ASes using IP-to-AS data from Team Cymru [32] and using IXP-specific data from PeeringDB [33]. In a traceroute measurement, if consecutive unresponsive hops are surrounded by responsive ones, we check whether the surrounding hops have a single sequence of responsive hops between them in other traceroutes; if that is the case, we substitute the unresponsive hops with the responsive ones. After this step, we map unresponsive hops whose surrounding responsive hops map to a single AS $a$ to the same AS $a$. If surrounding hops map to different ASes, we check whether public BGP feeds have a single sequence of ASes between them in AS-paths; if that is the case, we substitute the unresponsive hops to match the public AS-paths. If we still have unmapped or unresponsive hops, we ignore those hops on the AS-level path.

*c) Source granularity:* Our techniques are orthogonal to the granularity at which sources are defined. The only requirement is that each source appears in at most one catchment for each announcement configuration. For the evaluation, we define sources at the AS granularity. Different routers within an AS may choose different routes to a destination [34], e.g., routers in the US and Europe may choose different routes towards the announced prefix. In our dataset, this may also happen due to incorrect IP-to-AS mapping. Whenever we observe multiple routing decisions by an AS from multiple vantage points, we give higher priority to BGP measurements (over traceroute) to minimize errors due to IP-to-AS mapping. If multiple measurements of the same type remain, we assign the AS to the catchment most common across the available measurements. On average, we observe 2.28% of ASes in multiple catchments in an announcement configuration.

*d) Source visibility:* A source observed in some configurations may not respond to measurements in other configurations (e.g., due to route changes, BGP poisoning, or measurement errors). In these configurations, it is impossible to identify the catchment where the missing sources belong. We approach this problem in two steps. First, we limit our analysis to the set of sources that are observed in the first announcement anycasting the PEERING prefix from all 7 locations, without prepending or poisoning (i.e., the ASes observed on the default routes to the prefix). This avoids considering ASes observed only in a few, specific configurations. Second, we compute the

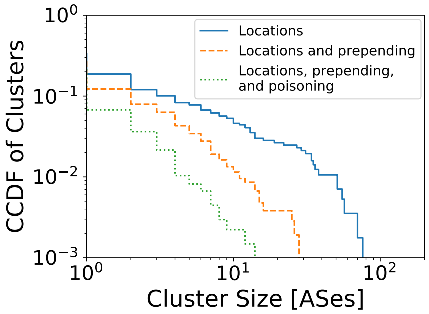

Figure 3: Distribution of cluster sizes after each phase.

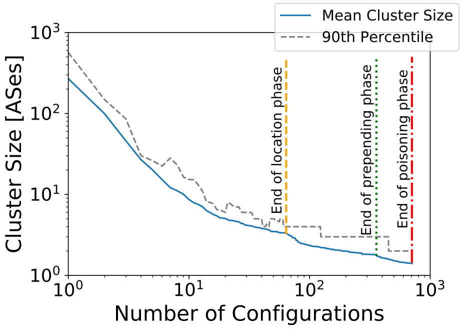

Figure 4: Cluster sizes as function of number of configurations.

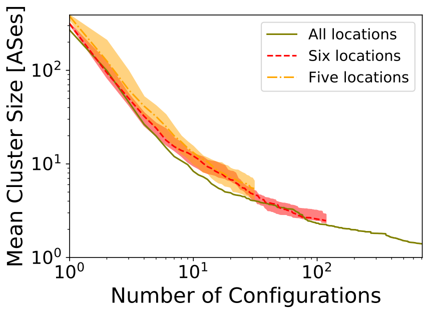

Figure 5: Mean cluster size when removing peering locations.

frequency that a source $s$ and each other source are in the same catchment across all configurations where $s$ was observed. We define $s_{\max}$ as the other source whose catchment $s$ appears most frequently in (i.e., $s$ and $s_{\max}$ route similarly). For each deployed configuration where source $s$ was not observed, we assign $s$ to the same catchment as $s_{\max}$.

*e) Ethical concerns:* PEERING prefixes do not carry production or user traffic, so no user traffic was impacted during our experiments. In practice, we expect networks to use prefixes dedicated to the location of sources of spoofed traffic to avoid impacting real users or applications.

Our route manipulation does not affect other prefixes or networks in the Internet. We note that anycast and AS-path prepending are common traffic engineering practices and can be observed on thousands of prefixes in routing tables today.

BGP poisoning has been used for more than a decade in research as a mechanism to route around failures and traffic blackholes [25], identify static default routes [23], discover network links [24], and characterize interdomain routing policies [18]. BGP poisoning does *not* impact the poisoned AS, routes to its prefixes, or its traffic. The PEERING platform conservatively limits each announcement to two poisoned ASes. To clearly signal BGP poisoning, PEERING requires experiments to surround each poisoned AS with PEERING's own AS47065. This avoids incorrect inference of peering links from BGP AS-paths (any false links inferred from poisoning would be with AS47065, which is easy to filter), and makes attribution to the PEERING platform trivial. PEERING maintains a blacklist of ASes that opt-out of BGP poisoning from the platform, but this list is currently empty as no ASes have complained about poisoning.

## V. Evaluation

In this section, we provide results on cluster sizes and show that we can manipulate routes to locate sources of spoofed traffic with good precision. Our results also indicate that large networks can apply the technique proposed to even greater effect, and that our techniques may potentially be used to locate sources of spoofed traffic during DDoS attacks.

### A. Cluster Sizes

For our PEERING announcements, Figure 3 shows the complementary cumulative distribution of cluster sizes, with logarithmic scales on both axes. We show one distribution at the end of each phase (i.e., after 64 configurations varying locations, 358 varying locations and prepending, and 705 including all techniques). We find all our techniques are effective in reducing cluster sizes. After deploying all 705 configurations built by the three techniques, 92% of clusters have a single AS. This indicates that, depending on the number and locations of the sources of spoofed traffic, our techniques may precisely locate them. Although most clusters are small and most ASes are in small clusters, large clusters account for a significant number of ASes. After deploying all 705 configurations, 14 clusters are larger than 5 ASes and contain 7.9% of the ASes in our dataset. Reducing cluster sizes at the tail is important to identify sources of spoofed traffic into small clusters and allow targeted intervention.

The lines in Figure 4 show the mean and 90th percentile of cluster sizes as a function of the number of configurations. Axes use a logarithmic scale and we sort configurations by the order in which they were deployed. We indicate when each phase finishes with vertical lines. The 90th percentile may increase whenever announcements partition large clusters. We observe diminishing returns in our ability to reduce cluster sizes by deploying additional announcement configurations. However, an origin AS *can* effectively manipulate routes towards each prefix, systematically causing catchment changes even after hundreds of configurations. In particular, the results indicate we could have obtained even smaller clusters by performing more announcements. The small steps following the vertical bars indicate that changing techniques used to generate configurations induces different route changes (new routes) and reduces cluster sizes.

### B. Impact of Peering Footprint

Figure 5 is similar to Figure 4, but it shows different lines for cases where we consider only a subset of our configurations, emulating networks with fewer PoPs by discarding one or two of the seven PoPs we used. The "all locations" line includes all $64+294 = 358$ configurations using all 7 locations

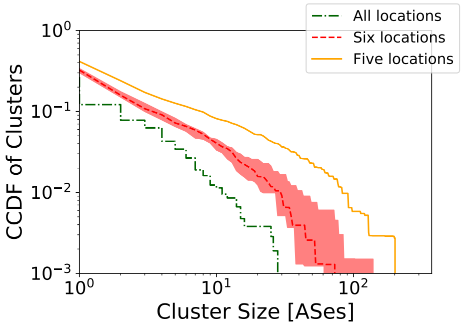

Figure 6: Distribution of cluster size after removing locations.

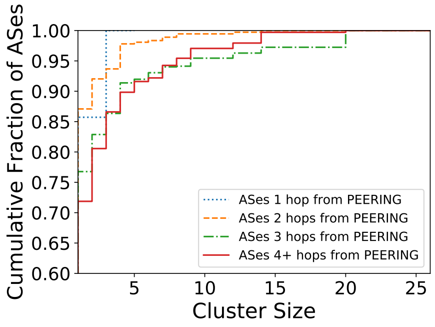

Figure 7: Cluster size as function of AS-hop distance from origin AS.

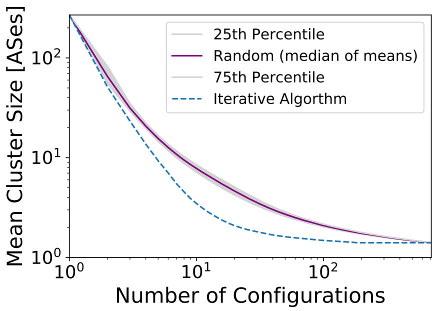

Figure 8: Mean cluster size as function of announcement schedule.

and prepending. The "six locations" line includes a subset of $\sum_{x=0}^{2}\left\{\binom{6}{6-x}+[6-x]\binom{6}{6-x}\right\} = 118$ configurations using up to 6 locations. The shaded area shows the minimum and maximum mean cluster sizes across all $\binom{7}{1} = 7$ possible subsets (each subset discarding one of the 7 PEERING PoPs we used). Similarly, the "five locations" line includes a subset of $\sum_{x=0}^{1}\left\{\binom{5}{5-x}+[5-x]\binom{5}{5-x}\right\} = 31$ configurations using up to 5 locations. The graph shows that having more locations allows the generation of more configurations, leading ultimately to smaller cluster sizes, and also yields smaller cluster sizes for the same number of announcements. This result indicates that a network with a footprint larger than PEERING's could achieve even higher localization precision. Moreover, although small networks with few peering locations may not be able to apply our techniques effectively, any network with a large peering footprint can deploy our techniques to identify the sources of spoofed traffic and help other networks mitigate amplification DDoS attacks.

Figure 6 is similar to Figure 3 and shows the complementary distribution of cluster sizes but considering fewer announcement locations. The three lines and shaded areas correspond to the same three scenarios described in Figure 5. The "all locations", "six locations", and "five locations" lines show the distributions of cluster sizes after 358, 118, and 31 announcements, respectively; in other words, we show the distribution of cluster sizes at the end of the curves in Figure 5. We observe that discarding some announcement locations leads to larger cluster sizes at the tail. While the "all locations" line shows 0.1% of clusters with more than 25 ASes, the "six locations" and "five locations" lines show 1.27% and 4.29% of clusters with more than 25 ASes, respectively.

We have also evaluated the distribution of cluster sizes as a function of the distance, in number of AS-hops, between PEERING PoPs and ASes. Figure 7 shows the distribution of cluster sizes across all ASes in our dataset. We break ASes into groups based on their AS-hop distance to the closest PEERING location observed across all configurations. We find that ASes that are 1 or 2 AS-hops away from PEERING PoPs are in clusters with 1.85 ASes on average, while ASes 3 or more AS-hops away are in clusters with 2.64 ASes on average. As we expected, ASes closer to announcement locations are easier to isolate (in smaller clusters), but most ASes farther away are also in small clusters, indicating that we may still be able to identify sources of attacks that are farther away with actionable precision. Figure 7 shows that large clusters (e.g., with 10 ASes or more) are usually further away from announcement locations. As future work, we plan to investigate targeted poisoning of distant ASes to induce route changes specific to split these large distant clusters.

### C. Localization Speed

The number of possible announcement configurations grows exponentially with the number of peering links $|\mathcal{L}|$. A straightforward approach to speed up localization is to use multiple prefixes and deploy multiple configurations concurrently. This approach, however, requires spare IP space, which may be limited in IPv4. In the following we discuss heuristics that do not depend on additional resources.

When locating the sources of spoofed traffic at runtime, e.g., during an attack, a network can reuse previous catchment measurements or remeasure catchments during identification. For example, an origin AS employing our techniques deploys time-consuming announcement configurations and measures catchments prior to the occurrence of an amplification DDoS attack. This involves a trade-off between identification accuracy (reusing previous catchment measurements may incur errors due to route changes) and identification delay (measuring catchments during identification takes time), which depends on route stability and could be improved by resource-efficient solutions for inferring path changes. While an attack is ongoing, the origin AS can then assume that catchments remain unchanged since their last measurement and deploy configurations in optimal order to quickly reduce cluster sizes.

The solid line in Figure 8 shows the mean cluster size as function of the number of announcement configurations when the origin AS chooses the sequence of configurations at random, without repetition. The shaded area shows the variance across 30,000 random sequences. The dashed line shows the mean cluster size as a function of the number of announcement configurations deployed when the origin AS chooses the configuration that results in the smallest mean

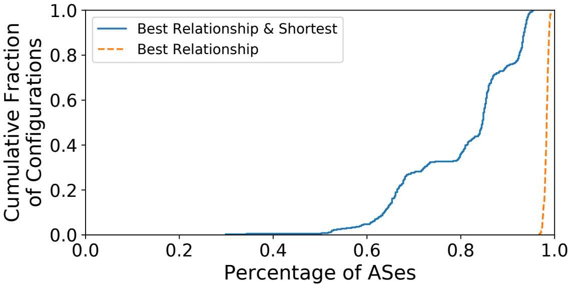

Figure 9: Percentage of ASes following well-known routing policies across configurations.

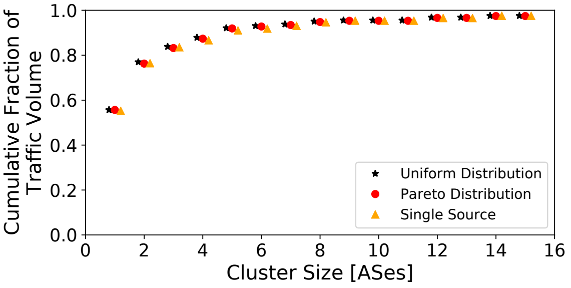

Figure 10: Distribution of cluster size as function of traffic volume for different distributions of spoofed sources.

cluster size before deploying each configuration. Compared to the solid line, we observe that localization can be made significantly faster if catchments are measured prior to an attack, and configurations deployed in optimal order. For example, after running ten configurations, while the random sequence yields a mean cluster size of 7.8 ASes, the optimal sequence yields a cluster size of 3.5 ASes. While our techniques can be used offline to identify networks that allow spoofed packets and drive adoption of filtering, this result indicates the techniques might be useful at run time as a source of information for active attack mitigation mechanisms.

Another approach to increase localization speed is to predict the catchments of announcement configurations and only deploy the most promising configurations, discarding or postponing configurations predicted to provide little additional information (i.e., not induce new, different route changes).

An AS in the Internet can receive multiple routes from its neighbors via BGP and choose the best according to its policy. We evaluate routing choices of ASes in our dataset according to BGP's first two decision criteria: (i) *best relationship*, which states an AS prefers routes through a client network first, through a peer second, and through a provider last; (ii) *shortest path*, which states that, when multiple equally-preferred routes are available (tied according to the relationship criterion), the AS chooses the shortest one. We do not consider additional decision criteria as we cannot observe them from AS-paths we collect from BGP updates or traceroute measurements.

The dashed line in Figure 9 shows the distribution of the fraction of ASes observed to follow the best relationship criterion across all configurations. We observe that most ASes in the Internet follow the best relationship criterion. The solid line shows the distribution of ASes observed to follow *both* best relationship and shortest path criterion (also referred to as the Gao-Rexford model [35]). This result indicates that most ASes in the internet follow a well-defined, known behavior. Although predicting routes in the Internet is challenging [36], new catchment prediction strategies may be able to use our techniques to boost attack localization speed.

### D. Study of Spoofed Traffic

Given the impossibility of attracting real spoofed traffic using PEERING testbed resources,[1] we study identification accuracy using simulation. We perform simulations where we choose the number of sources of spoofed traffic across ASes according to the uniform and Pareto distributions. For the Pareto distribution, we set the shape parameter such that 80% of sources of spoofed traffic are concentrated in 20% of ASes. We also run simulations with a single source of spoofed traffic placed in an AS chosen at random. We assume the volume of spoofed traffic originated in an AS is proportional to the number of sources in it. The first two scenarios are challenging scenarios, although previous work indicate that amplification attacks usually originate from a single source [10]. For each distribution, we generate and run simulations for 1000 placements.

For each distribution, Figure 10 shows the cumulative fraction of spoofed traffic ($y$-axis), averaged over the 1000 placements, in clusters up to a given size ($x$-axis). We observe that for all three distributions, most spoofed traffic originates from ASes in small clusters, which follows from Figure 3, where we showed that most clusters are small.

## VI. Other Applications

Although we designed our techniques to generate configurations with the goal of tracking down the sources of spoofed traffic, route manipulation has several other applications. In this section we discuss how our techniques can be adapted or extended for other work. We also discuss how our dataset,[2] which we make publicly available, can be used as a starting point of analysis. Although PEERING and RIPE Atlas are publicly accessible, deploying hundreds of announcement configurations takes weeks, and our measurements have higher coverage than usually possible as we partnered with RIPE to collect a larger body of traceroute measurements.

Our techniques generate configurations that systematically explore routes and are applicable to previous work that manipulate BGP announcements to identify alternate paths [25],

[1] PEERING operators expressed concerns about hosting a honeypot on PEERING, subsequent blacklisting of PEERING resources, and deterioration of the platform's future usability by the community. Some PEERING locations have bandwidth limitations which complicates hosting honeypots: although AmpPot [10] can enforce a limit on the sending rate, one cannot control the rate at which malicious packets are received from attackers.

[2] https://homepages.dcc.ufmg.br/~osvaldo.morais/dataset_ifip2020/

Table II: Summary of proposals for IP traceback.

| Approach | Manipulates | Cooperation from networks | Router updates | Router overhead | Identification precision | Identification delay |
|---|---|---|---|---|---|---|
| Manual | Logs/monitoring | Required | No | No | Path prefix | Long |
| Flooding [3] | Packet loss | Required | No | High | Path prefix | Moderate |
| Marking [4]–[6] | IP ID field | Deployment | Yes | Low | Closest router | $\approx$ sampling |
| Out-of-band [7] | — | Deployment | Yes | High | Closest router | $\approx$ sampling |
| Digest-Based [8], [9] | Local state at router | Deployment | Yes | High | Closest router | Low |
| Routing (this paper) | Routes | No | No | No | AS | Long |

[37], trigger route changes with specific properties [38], discover network links [24], and characterize interdomain routing policies [18]. In general, our techniques and dataset can be of use to research in these areas: our dataset contains at least four alternate routes towards PEERING for each observed AS, has thousands of route changes (with different properties), and may discover new links (particularly as a result of our poisoning experiments). More specifically, while Anwar et al. generate announcement configurations to infer routing policies of a single target AS [18], our techniques deterministically force routing changes and explore routing decisions across all ASes in the Internet; such an approach could significantly speed up (and scale) inference of routing policies.

Research that involves prefix hijacks and defenses against it frequently deploys BGP announcements in the Internet to perform controlled hijacks and evaluate the effectiveness of their approaches in realistic scenarios (e.g., [39]). A scenario commonly studied in the literature is that of subprefix hijacks, where the hijacker announces a more specific route. This scenario, however, has a predictable outcome: the hijack is guaranteed to attract all traffic as Internet routing follows longest-prefix matching. A partial mitigation to subprefix hijacks is to announce more specific routes. In this context, the impact of a hijack depends on how competing announcements of /24 IPv4 and /48 IPv6 prefixes from a given set of locations propagate, which our announcements can be used to study. Our technique to generate configurations varying announcement locations generates *all* possible scenarios of prefix hijacking from a predefined set of announcement locations. Consider a configuration announcing from $n$ locations: each location can be considered a legitimate announcement or an attempted hijack. Under this view, a configuration announcing from $n$ locations covers $2^n$ possible hijack scenarios.

## VII. Related Work

*a) DDoS attacks:* Recent DDoS attacks have reached peaks of 1 Tbps, significantly disrupting a wide range of Internet services. Amplification reflection DDoS attacks [1], where origins send small queries with the source IP address set to the victim's IP address such that large responses from responders flood the victim, have the potential to be significantly more disruptive and harder to mitigate. Current DDoS protection services (e.g., [40]) only reduce the impact of attacks by absorbing or scrubbing high volume attack traffic, without mitigating the origins of the attack. Locating the origins of reflection attacks, a first step toward mitigation, is challenging as attack origins send spoofed packets.

*b) Locating sources of spoofed traffic:* Previous approaches to locate the sources of spoofed packets either lack coverage or are not deployable. A study that relied on active tests included data from volunteers at 12,500 IP addresses [41], a small fraction of the billion client IP addresses seen by large Internet services that are the victims of attacks.

*c) IP Traceback:* Over the past two decades, several proposals for IP traceback have been put forward in an attempt to track down the sources of unwanted traffic. Table II provides an overview and compares with our proposal.

One way to perform IP traceback is to *iteratively* contact operators of ASes along the path towards the source of spoofed traffic and have them manually identify which links are carrying the traffic (with assistance from monitoring tools). *Controlled flooding* [3] was the first automated approach for IP traceback, and relied on temporarily congesting links to disrupt traffic on a link, allowing the victim to iteratively identify links on the path towards the attacker. Although this approach does not require upgrading routers, it is not viable today as the ability to trigger congestion at will (e.g., using UDP chargen) is considered a serious vulnerability.

Several packet *marking* approaches propose encoding information about routers traversed by a packet on a small fraction of packets (usually in the IP ID field) [4]–[6]. Similarly, routers can inform destinations using *out-of-band* data about a small fraction of packets they have forwarded towards each destination [7]. Under the assumption that attackers generate many packets towards the victim, the victim can correlate information across multiple packets and identify routers on the path to the attacker. Another approach is to compute a *digest* (e.g., a bloom filter) of packets traversing a router, and provide an interface for querying routers for a packet's signature [8], [9]. These techniques allow for fast identification, but require upgrading routers, incur significant overhead, and require widespread deployment across the Internet to provide accurate identification.

## VIII. Conclusion and Future Work

Our control-plane traceback technique can be deployed by any network with rich connectivity today, without changes to routers, and does not require cooperation from other networks. Our results using the PEERING platform indicate that our proposed techniques to generate announcement configurations can effectively manipulate routes and induce catchment changes, allowing tracking down the sources of spoofed traffic. If sources of amplification DDoS attacks are few, as reported by analyzing logs from AmpPot honeypots [10], our techniques can map sources of spoofed traffic into sets that average 1.40 ASes. Our results indicate that precision will be higher if networks with a footprint larger than PEERING's were to deploy our techniques.

We envision two research fronts for future work. One is to expand our techniques to reduce cluster sizes even more, e.g., designing new algorithms for choosing targets for poisoning, and using BGP communities for controlling export policies (and influence routing decisions) on remote networks. Another is to expand the system to allow identification of sources of spoofed traffic during DDoS attacks, e.g., by (i) jointly optimizing for cluster size and traffic volume, giving higher utility to reducing the size of clusters inferred to send more spoofed traffic; and (ii) improving existing catchment prediction techniques [13] to allow generation of announcement configurations without prior knowledge and reducing the need for measuring catchments in advance.

## Acknowledgements

We thank RIPE for the increased measurement budget used to perform traceroutes from RIPE Atlas toward PEERING prefixes. This work was supported by CAPES, CNPq, FAPEMIG, MCTIC/RNP grants 2955 and 2956; projects INCTCyber, Atmosphere, MASWEB, INCT InterSCity; and NSF grants CNS-1740883 and CNS-1835252.

## References

[1] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks," in *Proc. ACM IMC*, 2014.

[2] M. Prince, "Technical Details Behind a 400Gbps NTP Amplification DDoS Attack," Feb 2014. [Online]. Available: https://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack

[3] H. Burch and B. Cheswick, "Tracing Anonymous Packets to Their Approximate Source," in *Proc. USENIX Conf. on System Admin.*, 2000.

[4] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 226–237, 2001.

[5] A. Belenky and N. Ansari, "On Deterministic Packet Marking," *Comput. Netw.*, vol. 51, no. 10, pp. 2677–2700, 2007.

[6] Z. Gao and N. Ansari, "A Practical and Robust Inter-domain Marking Scheme for IP Traceback," *Comput. Netw.*, vol. 51, no. 3, pp. 732–750, 2007.

[7] S. Bellovin, M. Leech, and T. Taylor, "ICMP Traceback Messages," 2003, https://tools.ietf.org/html/draft-ietf-itrace-04.

[8] M. Sung, J. Xu, J. Li, and L. Li, "Large-scale IP Traceback in High-speed Internet: Practical Techniques and Information-theoretic Foundation," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1253–1266, 2008.

[9] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, "Single-packet IP Traceback," *IEEE/ACM Trans. Netw.*, vol. 10, no. 6, pp. 721–734, 2002.

[10] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow, "AmpPot: Monitoring and Defending Against Amplification DDoS Attacks," in *Proc. Intl. Symp. on Research in Attacks, Intrusions and Defenses (RAID)*, 2015.

[11] W. B. de Vries, R. de O. Schmidt, W. Hardaker, J. Heidemann, P.-T. de Boer, and A. Pras, "Verfploeter: Broad and load-aware anycast mapping," in *Proc. ACM IMC*, 2017.

[12] F. Lichtblau, F. Streibelt, T. Krüger, P. Richter, and A. Feldmann, "Detection, Classification, and Analysis of Inter-domain Traffic with Spoofed Source IP Addresses," in *Proc. ACM IMC*, 2017.

[13] P. Sermpezis and V. Kotronis, "Inferring Catchment in Internet Routing," in *SIGMETRICS Poster Session*, 2019.

[14] B. Schlinker, T. Arnold, I. Cunha, and E. Katz-Bassett, "PEERING: Virtualizing BGP at the Edge for Research," in *Proc. ACM CoNEXT*, 2019.

[15] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," 2000, http://www.ietf.org/rfc/rfc2827.txt.

[16] V. Giotsas, G. Smaragdakis, C. Dietzel, P. Richter, A. Feldmann, and A. Berger, "Inferring BGP Blackholing Activity in the Internet," in *Proc. ACM IMC*, 2017.

[17] P. Marques, N. Sheth, R. Raszuk, B. Greene, J. Mauch, and D. McPherson, "RFC 5575: Dissemination of Flow Specification Rules," Aug. 2009, http://www.ietf.org/rfc/rfc5575.txt.

[18] R. Anwar, H. Niaz, D. R. Choffnes, I. Cunha, P. Gill, and E. Katz-Bassett, "Investigating Interdomain Routing Policies in the Wild," in *Proc. ACM IMC*, 2015.

[19] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye, "Analyzing the Performance of an Anycast CDN," in *Proc. ACM IMC*, 2015.

[20] P. Sun, L. Vanbever, and J. Rexford, "Scalable Programmable Inbound Traffic Engineering," in *Proc. ACM SOSR*, 2015.

[21] D. Cicalese and D. Rossi, "A Longitudinal Study of IP Anycast," *SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 1, pp. 10–18, 2018.

[22] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng, "Engineering Egress with Edge Fabric: Steering Oceans of Content to the World," in *Proc. ACM SIGCOMM*, 2017.

[23] R. Bush, O. Maennel, M. Roughan, and S. Uhlig, "Internet Optometry: Assessing the Broken Glasses in Internet Reachability," in *Proc. ACM IMC*, 2009.

[24] L. Colitti, *Internet Topology Discovery Using Active Probing*, 2006.

[25] E. Katz-Bassett, C. Scott, D. R. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy, "LIFEGUARD: Practical Repair of Persistent Route Failures," in *Proc. ACM SIGCOMM*, 2012.

[26] Y.-C. Chiu, B. Schlinker, A. B. Radhakrishnan, E. Katz-Bassett, and R. Govindan, "Are We One Hop Away from a Better Internet?" in *Proc. ACM IMC*, 2015.

[27] J. M. Smith, K. Birkeland, and M. Schuchard, "An Internet-scale feasibility study of BGP poisoning as a security primitive," *CoRR*, vol. abs/1811.03716, 2018.

[28] M. Luckie, B. Huffaker, K. Claffy, A. Dhamdhere, and V. Giotsas, "AS Relationships, Customer Cones, and Validation," in *Proc. ACM IMC*, 2013.

[29] "The University of Oregon Routeviews Project," http://www.routeviews.org.

[30] "RIPE Routing Information Service," http://www.ripe.net/data-tools/stats/ris.

[31] RIPE, "RIPE Atlas," 2013, https://atlas.ripe.net.

[32] Team Cymru, "IP to ASN Mapping," http://www.team-cymru.org/Services/ip-to-asn.html.

[33] G. Nomikos and X. Dimitropoulos, "traIXroute: Detecting IXPs in Traceroute Paths," in *Proc. of PAM*, ser. LNCS, vol. 9631. Springer, 2016, pp. 346–358.

[34] V. Giotsas, M. Luckie, B. Huffaker, and kc claffy, "Inferring Complex AS Relationships," in *Proc. ACM IMC*, 2014.

[35] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, 2001.

[36] I. Cunha, P. Marchetta, M. Calder, Y.-C. Chiu, B. Schlinker, B. V. A. Machado, A. Pescape, V. Giotsas, H. V. Madhyastha, and E. Katz-Bassett, "Sibyl: A Practical Internet Route Oracle," in *Proc. USENIX NSDI*, 2016.

[37] V. Valancius, B. Ravi, N. Feamster, and A. C. Snoeren, "Quantifying the Benefits of Joint Content and Network Routing," in *Proc. ACM SIGMETRICS*, 2013.

[38] U. Javed, I. Cunha, D. R. Choffnes, E. Katz-Bassett, T. Anderson, and A. Krishnamurthy, "PoiRoot: Investigating the Root Cause of Interdomain Path Changes," in *Proc. ACM SIGCOMM*, 2013.

[39] P. Sermpezis, V. Kotronis, P. Gigis, X. Dimitropoulos, D. Cicalese, A. King, and A. Dainotti, "ARTEMIS: Neutralizing BGP Hijacking Within a Minute," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, p. 2471–2486, 2018.

[40] "CloudFlare – Advanced DDoS Protection and Mitigation," https://www.cloudflare.com/ddos/.

[41] R. Beverly, A. Berger, Y. Hyun, and k. claffy, "Understanding the Efficacy of Deployed Internet Source Address Validation Filtering," in *Proc. ACM IMC*, 2009.