2013-10

Speedtrap: Internet-Scale IPv6 Alias Resolution

Luckie, Matthew; Beverly, Robert; Brinkmeyer, William; claffy, kc

Speedtrap: Internet-Scale IPv6 Alias Resolution

Matthew Luckie
CAIDA / UC San Diego
mjl@caida.org

Robert Beverly
Naval Postgraduate School
rbeverly@nps.edu

William Brinkmeyer
Naval Postgraduate School
wdbrinkm@nps.edu

kc claffy
CAIDA / UC San Diego
kc@caida.org

ABSTRACT

Impediments to resolving IPv6 router aliases have precluded understanding the emerging router-level IPv6 Internet topology. In this work, we design, implement, and validate the first *Internet-scale alias resolution technique* for IPv6. Our technique, *speedtrap*, leverages the ability to induce fragmented IPv6 responses from router interfaces in a particular temporal pattern that produces distinguishing per-router fingerprints. Our algorithm surmounts three fundamental challenges to Internet-scale IPv6 alias resolution using fragment identifier values: (1) unlike for IPv4, the identifier counters on IPv6 routers have no natural velocity, (2) the values of these counters are similar across routers, and (3) the packet size required to collect inferences is 46 times larger than required in IPv4. We demonstrate the efficacy of the technique by producing router-level Internet IPv6 topologies using measurements from CAIDA's distributed infrastructure. Our preliminary work represents a step toward understanding the Internet's IPv6 router-level topology, an important objective with respect to IPv6 network resilience, security, policy, and longitudinal evolution.

Categories and Subject Descriptors

C.2.5 [**Local and Wide-Area Networks**]: Internet; C.2.1 [**Network Architecture and Design**]: Network topology; C.2.3 [**Computer Communication Networks**]: Network Operations—*network monitoring*

Keywords

Internet topology; alias resolution; IPv6

1. INTRODUCTION

The Internet operations and engineering community is putting significant effort into deploying IPv6 [10, 8, 6]. As IPv6 gains importance, the research community is in a position to study the deployment of this new Internet protocol using lessons learned studying the IPv4 Internet [5]. In this paper, we introduce *speedtrap*, our Internet-scale IPv6 alias resolution technique. IPv6 alias resolution is the process of determining if two IP addresses are assigned to different interfaces of the same physical router [15]. *speedtrap* is an active measurement technique that reduces an interface-level graph inferred from traceroute measurements to a router-level graph, facilitating better understanding of the resilience and robustness properties of the network [30].

Several IPv4 router alias inference techniques exist, each empirically providing varying degrees of success such that a combination of methods yields the best results: Ally [27], RadarGun [2], and MIDAR [16] use the Identifier (ID) field built into the IPv4 header; DisCarte [25] uses the record route IP option and graph analysis; Mercator [11] uses common source addresses in reply packets; Sherry et al. [24] use the pre-specified timestamp IP option. However, these methods rely on characteristics of IPv4 that are not present in IPv6. The IPv6 header does not include an ID field, there is no record route or pre-specified timestamp options, and the source address of ICMP6 responses must match the destination probed if it exists on the host [7].

Because of the protocol-level differences between IPv4 and IPv6, prior work on IPv6 alias resolution has sought protocol features in IPv6 that could be exploited to resolve aliases, in particular the IPv6 source routing feature (e.g. [29, 23, 22]). However, the IPv6 source routing has been deprecated [1] and the number of probes required scale $O(N^2)$ with the number of interfaces to compare. More recently, we showed that it is possible to obtain ID values in IPv6 useful for alias resolution by inducing routers to send fragments [4]. However, that method scales $O(N^2)$, limiting its application.

This paper describes an Internet-scale application of the technique, dubbed *speedtrap* as it induces velocity in normally stationary ID counters; other than responses to our probes, routers do not send fragments. We show its efficacy by running it on the IPv6 Internet where we obtain $\approx$ 11k routers from $\approx$ 53k interfaces. Validation against ground truth from network operators yields 451 of 453 correct inferences, comprising 2% of the 11,181 routers we infer. While not exhaustive, our technique represents a step toward understanding the IPv6 topology. *speedtrap* is implemented in the freely available scamper [19] tool, and the IPv6 router-level graphs are available from CAIDA. The remainder of the paper reviews related work (§2), details the *speedtrap* algorithm (§3), and presents results (§4). We then provide preliminary properties of the IPv6 topology (§5) and suggest avenues for further exploration.

IMC'13, October 23–25, 2013, Barcelona, Spain.

http://dx.doi.org/10.1145/2504730.2504759.

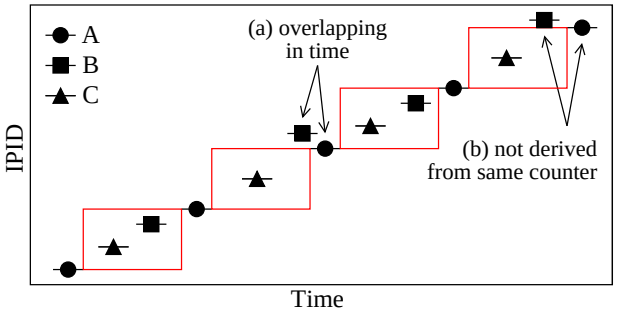

Figure 1: The Monotonic Bounds Test (MBT) used by MIDAR. For two interfaces to share a counter, non-overlapping IPID samples must strictly increase. The horizontal lines with each sample indicate the length of time each probe was outstanding in the network. The boxes between IPID samples from A show the bounds within which samples from aliases must be observed. The MBT suggests that A and C share a counter, but that A and B do not. The samples annotated with (a) are not used to determine shared counter status because it is not possible to determine packet arrival order.

2. RELATED WORK

Three alias resolution techniques for IPv4 use the ID field: Ally [27], RadarGun [2] and MIDAR [16], with RadarGun addressing scaling issues of Ally, and MIDAR addressing accuracy issues of RadarGun. A fundamental property these techniques exploit is that many router implementations use a single, shared ID counter across all of their interfaces. This section summarizes this prior work, with particular focus on the Monotonic Bounds Test (MBT) used by MIDAR, a modified version of which is utilized by *speedtrap*; [16] contains a thorough discussion of MIDAR's MBT and the accuracy of Ally, RadarGun, and MIDAR.

Ally's main limitation is that the number of probe packets required to resolve a graph for aliases scales $O(N^2)$ with the number of interfaces in the graph. RadarGun [2] surmounts Ally's scaling limitation by sending probes to all interfaces in multiple rounds to build a time series: two interfaces are then aliases if they both produce a linear time series and the time series are within a threshold. Unfortunately, the time series of two different routers can be within the threshold, producing a false router. MIDAR [16] provides better accuracy than RadarGun by using a Monotonic Bounds Test (MBT). The MBT is illustrated in figure 1: for two interfaces to be aliases, the ID values returned from a sequence of non-overlapping probes must strictly increase over time. If they do not, as with the points annotated with (b) in figure 1, then the interfaces cannot be using a shared counter.

In this paper, we use the MBT of MIDAR for *speedtrap*. We tailor our probing algorithm to features and challenges unique to IPv6, specifically: (1) no ID field in the IPv6 header, thus requiring us to induce the interface to send an IPv6 fragment header [4], (2) a 32-bit ID counter, as opposed to 16-bit in IPv4, (3) lack of ID velocity (see §3), and (4) the absence of entropy in ID values from unrelated routers. Further, to elicit an IPv6 fragment ID as in [4], we must manage the load induced by the large probe packets we require.

3. TECHNIQUE AND DATA

In this section, we describe the primitive we used to infer aliases, issues in applying existing alias resolution techniques to IPv6, and the Internet-scale alias resolution technique we use to infer aliases. We also describe our application of the technique to the IPv6 interface-level graph captured by CAIDA's Archipelago (Ark) infrastructure [14] for March 2013. The graph consists of *all* the 52,986 IPv6 interfaces numbered within the 2000::/3 unicast prefix captured from all 27 Ark vantage points (VPs) with IPv6 connectivity.

3.1 Obtaining and using the IPv6 ID field

The IPv6 header differs from the IPv4 header in many ways; one important difference is the absence of the ID field used for fragmentation and reassembly. The IPv6 protocol shifts the burden of fragmentation to the sender; no in-network packet fragmentation is done by routers. If a sender must fragment a packet, it includes an IPv6 extension header on the fragments which includes a 32-bit ID field necessary for reassembly.

Building on our technique in [4], *speedtrap* obtains an ID field by sending a router an ICMP packet too big message (PTB) with an MTU field smaller than the size of the packets solicited from it. In this work, we send 1300-byte ICMP echo request packets; when we receive 1300-byte echo replies, we send the router a PTB message with an MTU of 1280 bytes. If the router follows the IPv6 protocol [9], it will subsequently send (induced) fragmented echo replies to our hosts. In our dataset, 32.1% of interfaces we probed sent fragmented echo replies with incrementing ID field values; 17.9% sent responses with a random value in the ID field, and 50.0% did not send fragments because they either did not respond to the echo request (30.2%), or appeared to ignore PTB messages (19.8%). Section 4.4 reports on the marginal gains possible by probing interfaces from multiple VPs; a different VP may receive fragmented responses from a router where another VP received no response. To better understand IPv6 fragmentation behavior implemented in various hardware, we obtained hardware ground truth from several commercial service providers via email communication. In our testing, routers manufactured by Cisco, Huawei, Vyatta, HP, and Mikrotik all return sequential fragment identifiers; only Juniper routers, originally based on BSD, return random identifiers. Inevitably an IPv6 router-level map will be composed using multiple complementary inference techniques, as with IPv4.

Soliciting fragments requires us to send large (>1280 byte) probe packets, compared with IPv4 where 28-byte probe packets are sufficient. The requirement to send packets 46 times larger than sent in IPv4 restricts the rate at which packets can be sent. However, we have found it rare for routers to fragment traffic in their normal operation. When we began our work, 28% of probed interfaces replied with an initial fragment ID value of zero or one [4], suggesting they had not sent any fragmented traffic since booting. Figure 2 shows the distribution of IPID values derived from a counter: 80% of the samples occupy 0.00002% of the sample space (i.e. have an ID < 1000 where the ID is a 2^{32} bit value).

Except for responses to our alias resolution probes, routers do not currently source fragmented traffic and so there is no background velocity. It seems unlikely that there will be background velocity in the future. ICMP error messages are limited to 1280 bytes in size so they do not require fragmen-

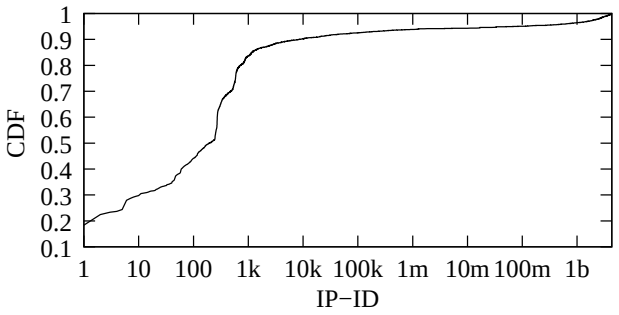

Figure 2: The first IPID value observed from April 2013 interfaces where a counter was inferred. Because there is no natural velocity to the ID counter in IPv6, 80% of the IPID values are less than 1000.

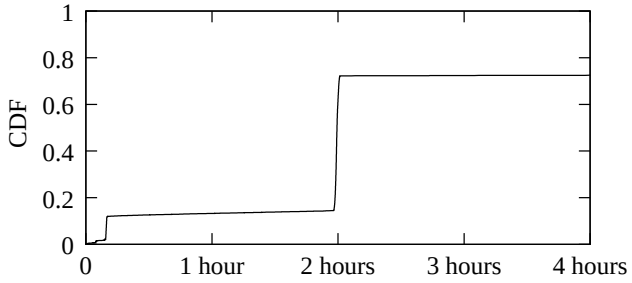

Figure 3: Length of time between sending a PTB and the final fragmented packet received from interfaces that assign IDs from a counter. 78% of interfaces send fragments for at least two hours; 8.7% send fragments for 10 minutes, and 0.8% send fragments for 5 minutes.

tation [7], and it is unlikely that routing protocol traffic will be fragmented in the future if it is not currently.

The minimum MTU in IPv6 is 1280 bytes; it is possible to send 5TB before the 32-bit field contains a duplicate value. Therefore, we can conduct Internet-scale alias resolution with a moderate probing rate without impairing our ability to accurately infer aliases. Routers are required to cache the Path-MTU when they receive a PTB. How long a router caches the Path-MTU depends on the implementation; RFC1981 [20] requires a system to cache for at least five minutes and recommends at least ten minutes. Figure 3 shows that a small fraction of routers cache for the recommended length of time; 78% of interfaces send our host fragments for at least two hours after we sent it a PTB message. This caching helps because it reduces the number of PTB messages sent to routers during the alias resolution process.

3.2 Speedtrap algorithm

We use the same MBT test as is used by MIDAR: for two interfaces to be aliases, the IPID sequence received from non-overlapping probes must strictly increase. Our probing strategy is tailored to determine aliases using the minimum number of probe packets necessary given that the ID field has no velocity except that caused by our probing. Briefly, we (1) determine the set of interfaces in our set that send responses with an IPID derived from a counter, (2) determine which interfaces appear to share a counter, (3) try to

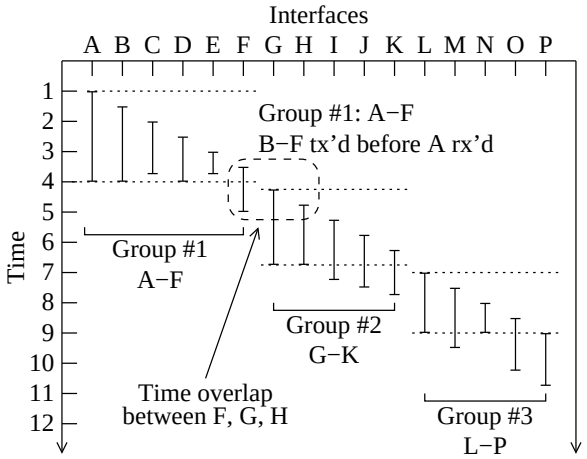

Figure 4: Grouping interfaces during step 2. The vertical bars represent the time a probe is outstanding in the network: between when we transmit and then receive a reply. Each group includes only probes transmitted before a response to the first was received. These groups can then be probed in parallel if there is no overlap between members of the groups in time: #1 and #3 can be probed simultaneously, but #2 cannot be probed at the same time as #1 because G and H (in #2) were transmitted before F (in #1) was received.

cause the counters of distinct routers to diverge, and (4) confirm aliases with pairwise probing. Given the absence of velocity in ID counters and the large probes required for the technique to work, we probe at a low rate of 20pps from a single VP, producing 26Kbps of traffic. We evaluate the performance of our technique in section 4.

Step 1: determine IPID behavior of interfaces. We send each interface six echo requests one second apart. We probe interfaces in a sliding window: probing at 20pps means we test 20 different interfaces at a time. We infer the interface is deriving fragment ID values from a counter provided we receive at least three responses and the difference in ID values for adjacent responses is less than 65,535. In practice, 99.8% of interfaces return perfectly sequential IPID values during this step if they use a counter. This step prevents the following stages from sending probes that will not help resolve router aliases.

Step 2: solicit a sequence of non-overlapping fragments from all interface-pairs. To test if interfaces A and B are aliases, we obtain a sequence of ID values A-B-A or B-A-B where the probes soliciting the ID values were not overlapping. We break this step into three rounds. In each round, we solicit a single ID value from every interface that we inferred to be deriving ID values from a counter; if we do not receive a fragment (and thus no ID value), we probe the interface up to two further times after waiting at least one second.

In the first round, we solicit a single fragmented response from all interfaces, probing in parallel to obtain samples quickly. Because this first round of probing is conducted in parallel, many ID samples are taken while multiple probes were in flight concurrently. To obtain the non-overlapping sequence necessary to infer if two interfaces might share a

counter, we assemble groups of interfaces where the previous samples were overlapping in the network. In the second round, we solicit a single fragmented response from each interface in the group one at a time. Figure 4 illustrates the grouping process: we visit samples in order of their transmission time and assemble a group by including all probes subsequently transmitted before the response to the first probe in the group was was received. To achieve time efficiency, we probe interface groups in parallel provided no members overlap between the groups in time. In Figure 4, group #2 cannot be probed at the same time as #1 because of overlap between samples from F, G, and H; however, we can probe group #3 at the same time we probe group #1 because no samples were taken concurrently. In the third round, we solicit a single fragmented response from all interfaces in parallel as we did in round one. On completion, we have obtained packet triplets (A-B-A or B-A-B) with non-overlapping probes for all pairs of interfaces (A, B), which allows us to test whether the interfaces might share a counter.

Step 3: distill candidate routers. We use the data acquired in step 2 to produce sets using a transitive closure (TC) of all interface-pairs that individually passed the MBT using the samples obtained in step 2. Because each closure can contain multiple distinct routers, we probe the interfaces in each closure that holds more than three interfaces, to try to force counters on different routers to diverge. Given a closure (A, B, C, D, E) we try to cause a divergence by sampling an interface between every other interface in the set, e.g. A, B, A, C, A. While the algorithm scales $O(N!)$ worst case, only the subset of interface-pairs that passed the MBT in step 2 will require testing. In addition, in this step we only have to interleave A where the previous sample was monotonically increasing; if a sample for B did not monotonically increase, then we can probe C without interleaving A because B could not have shared a counter with A and caused A's counter to increment. Because the closures represent distinct routers, we can probe separate closures in parallel without inducing rate limiting. After probing each set, we process the interfaces to form smaller closures where the MBT has ruled out shared counters between interfaces.

Step 4: pair-wise testing of candidate routers. The final step is to test each candidate pair of interfaces in each closure that remains after step 3. This step is necessary because there can be large time gaps between previous samples. For a pair of interfaces (A, B) we probe (A, B, A, B, A) and declare aliases if the MBT suggests a shared counter. Because the closures represent distinct routers, we again probe separate closures in parallel. The output from this process is a set of routers and their associated interface addresses.

4. RESULTS

This section provides our results. We first detail our real-world alias success rate, and properties of the aliases we infer. Second, we report our validation results, which relied on a combination of methods and out-of-band interaction with four providers. Of the inferred alias we could validate, $> 99\%$ of them were correctly assigned. Third, we evaluate the scalability of *speedtrap* and suggest future avenues of improvement to the algorithm. Fourth, we quantify the marginal benefit of performing *speedtrap* from multiple VPs to mitigate ICMP6 filtering and identify those networks likely to be filtering.

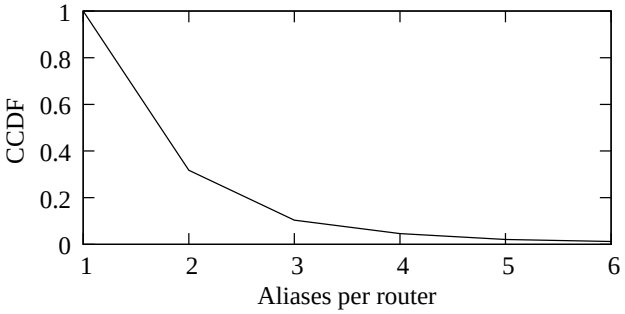

Figure 5: CCDF of number of interfaces observed per router, inclusive of interfaces that assign ID values from a counter. Our dataset contains a single interface for most routers.

Step		Packets	Time
1	IPID behavior	317,814	5:35:44
2	Non-overlapping sequence	80,017	1:15:31
3	Distill candidate routers	34,659	1:15:43
4	Pair-wise testing	63,765	1:01:12
Total:		496,255	9:08:10

Table 1: Packets and time required to complete each step evaluating 52,969 interfaces at 20pps. More than half of the time is spent determining the IPID behavior of interfaces in the set.

4.1 Inferred IPv6 Aliases

Overall, 17,002 interfaces (32.1%) sent echo replies with an incrementing IPID value. Speedtrap inferred 11,181 routers involving these interfaces; figure 5 shows the number of aliases inferred for each router. We observed a single interface on 68% of routers in our set, and two interfaces on 21%. Our validation in section 4.2 suggests that we only observed a single interface on those routers with traceroute, and not that interfaces belonging to the same router use independent counters. The largest router we observed contained 25 aliases. We are confident all aliases we inferred are true aliases because of the cleanness of the data collected in pair-wise testing: 11,083 of 11,086 pairs tested had perfectly sequential ID values in step 4. Table 1 shows the number of packets sent and the time taken to complete. Most time is used and packets sent in step 1; in section 4.3 we show increasing the probe rate and using multiple VPs reduces the time required.

4.2 Validation

We obtained validation data from four networks: a large access provider (AP), two small transit providers (STP-1, STP-2), and a Tier-1 network. Three of the four sets of validation data were obtained by deriving and then confirming an IP address naming convention used in DNS entries with operators of the networks. Additional networks were contacted but their naming conventions were not sufficiently clean to be used as validation data. The fourth set of validation data was obtained by extracting interfaces from a RANCID database [26]. Stale information was found in three of the four sources of validation data, including the RANCID database, which required in-depth discussion with the network operators involved. Table 2 shows the outcome of our

Validation name	STP-1	STP-2	AP	Tier1
Data source	RANCID	DNS	DNS	DNS
Routers				
Incr. IPID	43	40	86	50
Random IPID		43	85	98
No Fragments		11	84	77
No Echo replies			8	11
Mixed			4	3
Total Routers	70	94	267	239
Interfaces	151/750	85/279	138/1008	79/625
Correct	150/151	85/85	137/138	79/79

Table 2: Validation of IP to router assignments. Overall, 451 of 453 (99.6%) of assignments were correct. There exists surprising heterogeneity in router behavior in these networks.

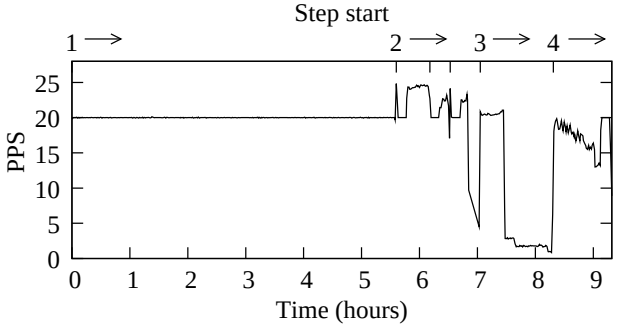

Figure 6: Data collection rate measured by packets-per-second (PPS) over time. Steps 1 (IPID behavior classification) and 2 (non-overlapping sequence) are limited by the configured PPS rate. Steps 3 (distill candidate routers) and 4 (pair-wise testing) are algorithmically limited). The reduction in PPS rate immediately before step 3 is due to $O(N^2)$ (pair-wise) IP-ID comparisons that distill the candidate routers.

validation exercise. Overall, 451 of 453 (99.8%) assignments were made correctly, and we validated 219 of 11,181 routers (2%). The two incorrect assignments were where a single interface was not correctly matched with its aliases, implying that we observed no aliases for the two interfaces. Both cases were due to our not receiving a response to probes sent in round 2 of step 2.

4.3 Scalability

Figure 6 evaluates the scalability of *speedtrap*, focusing on the PPS rate as data collection progressed. We configured scamper's probing rate at 20pps; for nearly the first seven of nine hours the experiment is limited by the configured probing rate; increasing the probing rate or using a distributed set of vantage points linearly reduces the time taken to complete these steps. Steps 3 and 4 are algorithmically limited; because we only probe one address in each candidate router at a time, the time these steps take to complete is limited by the size of the largest candidate router. At present, steps 3 and 4 probe all pairs of addresses in a candidate router. These steps can be improved by building a transitive closure as addresses are evaluated; if an incrementing sequence of IP-ID values is observed probing addresses A and B, then both A and B do not need to be tested against C. With a separate experiment, we confirmed that increasing the probing rate results in a linear reduction of the time to complete steps 1 and 2, and that steps 3 and 4 are currently algorithmically limited. As with MIDAR [16] and RadarGun [2], the number of packets required scales linearly with the number of interfaces to test.

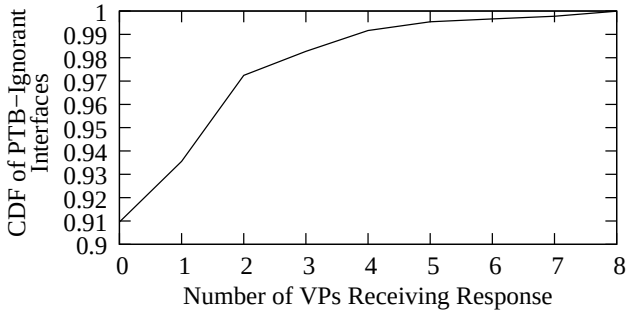

Figure 7: Dependence on vantage point (VP). We observed ≈9% of interfaces that were unresponsive from the first VP responding from at least one of eight different VPs. We observed diminishing marginal success with more vantage points.

4.4 Dependence on Vantage Point

We examined the impact of ICMP6 and fragment filtering by focusing on the set of 10,497 interfaces (19.8%) that do not send fragmented packets after we respond to their echo reply with a PTB, a set we term *PTB-ignorant*. Since we cannot know whether the PTB message arrived at the destination, we instead study their responsiveness using eight Ark VPs distributed around the world (2 in North America, 4 in Europe, 1 in Asia, 1 in Australia). Figure 7 shows the cumulative fraction of the PTB-ignorant interfaces versus the number of VPs that received a fragmented response. Overall, 9,548 (≈ 91%) of these still did not receive fragments from any of the eight VPs, implying that filtering is close to the interfaces. 949 interfaces that were seemingly PTB-ignorant replied with fragments when probed from a different VP. Of the 949 interfaces, only 7.2% returned sequential IDs; we leave understanding why the behavior of these interfaces differs from the whole set to future work.

We then mapped interfaces to autonomous systems (ASes) using the longest matching prefix observed in BGP. For 3% of ASes, all of their PTB-ignorant interfaces were reachable from at least one of the eight VPs. However, for 87% of the ASes, none of their interfaces returned fragments when probed from any of our eight VPs. These findings suggest that various forms of ICMP6 filtering exist in the production IPv6 Internet, perhaps due to security concerns [18].

We therefore performed tomography to infer which ASes might filter PTBs. We performed a traceroute from each of our 8 VPs to infer the forward paths to the interfaces, and then used the global IPv6 BGP table as available from Routeviews [21] to infer the forward AS path. Thus, from each VP to each candidate interface, we have both the AS path, and an understanding of whether that path returned fragments. For all $(VP, interface)$ paths where we received fragments, we infer that no ICMP6 filtering is in place for the ASes along that path, and assemble a set of ASes that do not filter. Then, for all paths where we received unfragmented ICMP6 echo responses, we find the first AS along the

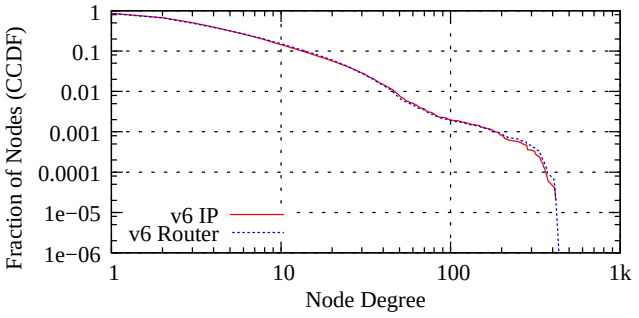

Figure 8: CCDF of IPv6 interface and router degree, inferred from 4.9M Ark traces (Apr, 2013).

path that is not in the unfiltered set. This first "closed" AS is likely the root of the filtering, since it appears in none of the AS paths where fragments are returned. We gathered the most prevalent ASes that likely filter ICMP6, which include: 5511 (France Telecom), 3265 (XS4ALL), 6327 (Shaw), 7843 (Time Warner), 10026 (Pacnet), and 3209 (Vodafone). Thus, performing *speedtrap* from multiple VPs can decrease the fraction of interfaces that do not return fragments, and increase the number of inferred aliases.

5. TOWARD AN IPV6 ROUTER MAP

We used *speedtrap* to resolve the IPv6 interface topology obtained from Ark during April 2013 into a router-level topology. The April 2013 data includes 4.9M IPv6 traces. Our analysis considered only the portion of the trace containing hops that sent ICMP6 time-exceeded messages. Using the *speedtrap*-inferred aliases, we reduced the graph from 49,542 distinct IPv6 interfaces connected by 165,832 edges (average degree of 6.69) to 42,929 nodes connected by 144,377 edges. 39,049 of the 49,542 interfaces (79%) have no known alias, which is similar to the corresponding ratio in a recent study of IPv4 aliases [12]. Figure 8 is a complementary CDF of the node degree distribution for the Ark IPv6 interface and *speedtrap* router-level topologies.

We manually investigated the 81 nodes with degree 100 or greater. 25 of 81 belong to the Tiscali group, including Tinet/Intelliquent, while 11 nodes belong to Level 3. Ten of these large degree nodes correspond to Hurricane Electric (HE) tunnel broker [13] servers, with `ams1.he.net` and `fra1.he.net` having 405 and 303 interfaces respectively. Within the IP-level topology and without alias resolution, `ams1.he.net` was inferred to be *different* nodes of degree 218, 194, 178, 189, 136, where some of the IPv6 interfaces corresponded to address space other than that owned by HE. This example illuminates the danger in using interface-level topologies, and the value in alias resolution for IPv6, where individual nodes may be disproportionately important at this stage of IPv6 evolution, or in the future. This finding is consistent with other recent measurements of BGP paths where HE, Level 3, and Tinet also appear as important parts of the IPv6 topology [10].

A limitation of our approach is that 17.9% of interfaces return random fragment identifiers, a function of the underlying router implementation. Unfortunately, MBT cannot make alias inferences over interfaces that return random identifiers. One potential enhancement to our technique is to use the PTB MTU as a nonce such that an alias of the interface to which the PTB is sent would return fragments of a distinguishing size. The ability to leverage MTU size for IPv6 alias resolution depends on the PMTU destination cache being shared across interfaces. In our testing, we find that Huawei, Vyatta, HP, and Mikrotik use a shared per-destination PMTU cache. However, we can already resolve aliases of these devices using *speedtrap*. Unfortunately, Juniper routers use a per-interface, per-destination PMTU cache. Thus, resolving the aliases of Juniper routers remains an open problem.

6. CONCLUSION

While researchers continue to make progress toward understanding the Internet's IPv4 topology, little is known about the IPv6 router-level topology. Relying on interface topologies such as those revealed by traceroute yields incorrect inferences and conclusions about the structure and resilience of the network [17, 30]. Alias resolution is crucial to producing a topology that represents actual equipment (routers and links). In IPv6, tunnels and virtual interfaces are common, and one cannot assume that nodes in the network core have low degree as might be dictated by physical router density constraints. Similarly, tunnels can directly connect distant IPv6 routers over many IPv4 hops. Understanding the IPv6 router-level topology is thus important both from security and Internet evolution perspectives [5].

We do not claim that our inferred IPv6 router-level topology is representative of the IPv6 Internet. In particular, our current research explores how to more efficiently and effectively perform active probes in IPv6 amid large amounts of topological sparsity encountered in random probes [3]. Further, combining graph analysis techniques such as those detailed in [15] will likely uncover additional router-level structure. However, our work represents an important step forward in more completely understanding the true IPv6 topology, and the first IPv6 router-level topologies.

In February 2013 the IETF updated the specification of the ID field in IPv4 so that the ID field is only set when a packet is fragmented [28]. If this RFC is followed, *speedtrap* will soon be required for IPv4 router alias resolution as well as IPv6. A related question is what fraction of IPv4 routers that set a constant (zero) ID field would set an incrementing IPID field if they were required to fragment responses.

Acknowledgments

We thank Owen DeLong, Aaron Hughes, and Brandon Ross for operational insight and ground truth, and the four anonymous operators of the networks (STP-1, STP-2, AP, Tier1) who confirmed our interpretation of their router naming convention which we used to validate our inferred router-level map. Bradley Huffaker supplied IPv4 degree distribution data for comparison purposes. Our shepherd, Lachlan Andrew, and the anonymous reviewers provided invaluable feedback. The work was supported by the NSF under grants CNS-1111445 and CNS-1111449, and by the U.S. Department of Homeland Security (DHS) Science and Technology (S&T) Directorate Cyber Security Division (DHS S&T/CSD) BAA 11-02 and SPAWAR Systems Center Pacific via contract number N66001-12-C-0130. This material represents the position of the author(s) and not necessarily that of NSF or DHS.

7. REFERENCES

[1] J. Abley, P. Savola, and G. Neville-Neil. Deprecation of type 0 routing headers in IPv6, Dec. 2007. RFC 5095.

[2] A. Bender, R. Sherwood, and N. Spring. Fixing Ally's growing pains with velocity modeling. In *ACM SIGCOMM IMC*, pages 337–342, Vouliagmeni, Greece, Oct. 2008.

[3] R. Beverly, A. Berger, and G. G. Xie. Primitives for active internet topology mapping: toward high-frequency characterization. In *ACM SIGCOMM IMC*, pages 165–171, 2010.

[4] R. Beverly, W. Brinkmeyer, M. Luckie, and J. P. Rohrer. IPv6 alias resolution via induced fragmentation. In *PAM*, Mar. 2013.

[5] k. claffy. Tracking IPv6 evolution: data we have and data we need. *CCR*, 41(3):43–48, July 2011.

[6] Comcast. Comcast Launches IPv6 for Business Customers, 2013. `http://corporate.comcast.com/comcast-voices/comcast-launches-ipv6-for-business-customers`.

[7] A. Contra, S. Deering, and M. Gupta. Internet control message protocol (ICMPv6) for the Internet protocol version 6 (IPv6) specification, Mar. 2006. RFC 4443.

[8] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey. Measuring IPv6 adoption. Technical Report TR-13-004, ICSI, Aug. 2013.

[9] S. Deering and R. Hinden. Internet protocol, version 6 (IPv6) specification, Dec. 1998. RFC 2460.

[10] A. Dhamdhere, M. Luckie, B. Huffaker, K. Claffy, A. Elmokashfi, and E. Aben. Measuring the deployment of IPv6: Topology, routing, and performance. In *ACM SIGCOMM IMC*, pages 537–550, Boston, MA, USA, Nov. 2012.

[11] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *IEEE INFOCOM*, pages 1371–1380, Tel-Aviv, Israel, Mar 2000.

[12] B. Huffaker, M. Fomenkov, and k. claffy. Internet Topology Data Comparison. Technical report, Cooperative Association for Internet Data Analysis (CAIDA), May 2012.

[13] Hurricane Electric. IPv6 tunnel broker service, 2013. `http://tunnelbroker.net/`.

[14] Y. Hyun and k. claffy. Archipelago measurement infrastructure, 2013. `http://www.caida.org/projects/ark/`.

[15] K. Keys. Internet-scale IP alias resolution. *ACM SIGCOMM CCR*, 40(1):50–55, Jan. 2010.

[16] K. Keys, Y. Hyun, M. Luckie, and k claffy. Internet-scale IPv4 alias resolution with MIDAR. *IEEE/ACM Transactions on Networking*, pages 383–399, Apr. 2013.

[17] B. Krishnamurthy and W. Willinger. What are our standards for validation of measurement-based networking research? *SIGMETRICS Perform. Eval. Rev.*, 36(2):64–69, Aug. 2008.

[18] S. Krishnan. Handling of Overlapping IPv6 Fragments. RFC 5722 (Proposed Standard), Dec. 2009.

[19] M. Luckie. Scamper: a scalable and extensible packet prober for active measurement of the internet. In *ACM SIGCOMM IMC*, pages 239–245, 2010.

[20] J. McCann, S. Deering, and J. Mogul. Path MTU discovery for IP version 6, Aug. 1996. RFC 1981.

[21] D. Meyer. University of Oregon RouteViews, 2013. `http://www.routeviews.org`.

[22] S. Qian, Y. Wang, and K. Xu. Utilizing destination options header to resolve IPv6 alias resolution. In *IEEE Globecom*, Miami, FL, USA, Dec. 2010.

[23] S. Qian, M. Xu, Z. Qiao, and K. Xu. Route positional method for IPv6 alias resolution. In *ICCCN*, 2010.

[24] J. Sherry, E. Katz-Bassett, M. Pimenova, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. Resolving IP aliases with prespecified timestamps. In *ACM SIGCOMM IMC*, pages 172–178, Melbourne, Australia, Nov. 2010.

[25] R. Sherwood, A. Bender, and N. Spring. DisCarte: a disjunctive Internet cartographer. In *ACM SIGCOMM*, pages 303–314, Seattle, WA, USA, Aug. 2008.

[26] Shrubbery Networks, Inc. RANCID - Really Awesome New Cisco confIg Differ. `http://www.shrubbery.net/rancid/`.

[27] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM*, pages 133–145, Pittsburgh, PA, USA, Aug. 2002.

[28] J. Touch. Updated specification of the IPv4 ID field, Feb. 2013. RFC 6864.

[29] D. G. Waddington, F. Chang, R. Viswanathan, and B. Yao. Topology discovery for public IPv6 networks. *ACM SIGCOMM CCR*, 33(3):59–68, July 2003.

[30] W. Willinger, D. Alderson, and J. C. Doyle. Mathematics and the Internet: A source of enormous confusion and great potential. *Notices of the AMS*, 56(5), 2009.