# bdrmap: Inference of Borders Between IP Networks

Matthew Luckie
University of Waikato
mjl@wand.net.nz

Amogh Dhamdhere
CAIDA / UC San Diego
amogh@caida.org

Bradley Huffaker
CAIDA / UC San Diego
bradley@caida.org

David Clark
MIT
ddc@csail.mit.edu

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

## ABSTRACT

We tackle the tedious and unsolved problem of automatically and correctly inferring network boundaries in traceroute. We explain why such a conceptually simple task is so hard in the real world, and how lack of progress has impeded a wide range of research and development efforts for decades. We develop and validate a method that uses targeted traceroutes, knowledge of traceroute idiosyncrasies, and codification of topological constraints in a structured set of heuristics, to correctly identify interdomain links at the granularity of individual border routers. In this study we focus on the network boundaries we have most confidence we can accurately infer in the presence of sampling bias: interdomain links attached to the network launching the traceroute. We develop a scalable implementation of our algorithm and validate it against ground truth information provided by four networks on 3,277 links, which showed 96.3% – 98.9% of our inferences were correct.

With 19 vantage points (VPs) distributed across a large U.S. broadband provider, we use our method to reveal the tremendous density of router-level interconnection between some ASes. In January 2016, the broadband provider had 45 router-level links with a Tier-1 peer. We also quantify the VP deployment required to observe this ISP's interdomain connectivity, with 17 VPs required to observe all 45 links. Our method forms the cornerstone of the system we are building to map interdomain performance, and we release our code.

## Keywords

Internet topology; Router ownership



*IMC 2016, November 14–16, Santa Monica, CA, USA.*
 ISBN 978-1-4503-4526-2/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2987443.2987467

## 1. INTRODUCTION

Every Internet researcher and operator is familiar with traceroute, but not everyone is aware of the inferential challenges in interpreting traceroute output, the diverse barriers to developing a more accurate Internet path inference capability, and the resulting impediments to a range of research and development efforts. Despite these impediments, researchers have relied on traceroute to support measurement and analysis of Internet router-level topologies for two decades.

Internet router-level topology discovery and inference is tedious and error-prone for at least five classes of reasons. Most obviously, the TCP/IP architecture has no notion of interdomain boundaries at the network layer. In fact, the architecture does not even have any mechanism to identify a complete router – an IP address identifies only a single interface on a router, and routers may have hundreds of (or thousands of virtual) interfaces.

Second, the measurement tool used for IP topology discovery implements a 30-year old clever hack in which an end host sends customized probe packets along a forward path toward a destination to trick each router into revealing one of their interface IP addresses to the end host. Constructing an Internet-scale topology requires superimposing millions of such measurements from multiple sources, onto an interface topology graph, and then applying heuristic techniques to infer which IP addresses are interfaces on the same physical router, i.e., alias resolution. Failure to accurately resolve such aliases for the same router will result in an inflated inference of the number of links between networks.

Third, operator address assignment and router implementation practices limit the accuracy of the canonical approach of mapping an IP address observed in traceroute to the organization that announces the longest matching prefix for that IP address in BGP [44]. Fourth, traceroute measurements repeatedly sample links close to the vantage point [22], resulting in topologies where links farther from the network are less likely to be observed. This sampling bias can reduce the accuracy of router ownership inferences because there are fewer topological constraints available.

Finally, operators could overcome all of these issues by assigning hostnames to router interfaces that identify the router itself, but concerns about revealing network topology information to competitors, or to prospective attackers, align operator incentives away from transparency about their internal network topologies.

These challenges impede many Internet research and development efforts, from realistic network modeling, protocol design, to assessment of real-world network properties such as interdomain congestion, infrastructure resiliency, and identifying vulnerabilities due to inadequate security policies. In light of these challenges, this paper makes the following five contributions:

**(1) We introduce a scalable method for accurately inferring the boundaries of a given network, and the other networks attached at each boundary.** Our approach efficiently infers forward IP paths from traceroute measurements, resolves IP aliases to construct a router-level topology [15, 40, 21], and then uses this topology as well as topological constraints inferred from BGP data [25] to narrow the possible set of links and associated IP addresses that represent borders between networks. Our approach builds on generally accepted assumptions about industry practices, such as numbering border router interfaces with IP addresses that belong to an upstream provider's network. Our method explicitly accommodates known limitations of traceroute, e.g., sometimes a border router responds to a traceroute probe using a source address belonging to a third-party AS, i.e., one that maps to neither network constituting the border. Even when a neighbor network does not respond to our probes, our algorithm can still pinpoint where they interconnect.

**(2) We develop an efficient system to allow deployment of our method on resource-limited devices.** We show that the probing our algorithm uses obtains a complete and accurate router-level interdomain map, but the densest measurement infrastructure deployments use resource-limited devices that cannot maintain state for the entire map themselves. We extend our measurement tools to support low-resource devices by interactively offloading data and state to a centrally-operated system.

**(3) We validate our algorithm's correctness using ground truth from four network operators as well as databases of IXP address use.** We used ground truth provided by a research and education network, a large access network, a Tier-1 network, and a small access network. For these networks, our algorithm correctly identified interdomain links and corresponding ASes for 96.3% – 98.9% of the 3,277 links inferred, finds nearly all (92% – 99%) AS neighbors observed in public BGP data, and correctly identifies interdomain links not observed in public BGP data.

**(4) We demonstrate the utility of our algorithm by analyzing the topology of a large access ISP to understand modern interconnection agreements.** Data we collected in January 2016 shows the presence of an astounding 45 interdomain links with one of the ISP's Tier-1 peers, geographically distributed throughout the ISP's network, with 25 of these links only visible from a vantage point (VP) in specific regions. For 73% of measured prefixes, we observed 5 – 15 distinct border routers carrying probe traffic.

**(5) We publicly release the source code implementation.** To promote further validation and use of our network border mapping measurement and analysis algorithm, which we call `bdrmap`, we publicly release our source code implementation as part of scamper [23].

Our method infers all interdomain links directly connected to and visible from the network hosting a single VP. Accurately inferring the parties involved in all interdomain links observed in traceroute requires overcoming the natural sample bias in traceroute [22], i.e., poorer visibility into distant networks, which limits our ability to assemble constraints. We build on years of prior work in topology discovery (e.g. [10, 2]), alias resolution (e.g. [15, 40, 5, 21]), AS relationship inference [25], and active probing systems (e.g. [23]). Our contributions are a scalable system that synthesizes this prior work, and a set of novel heuristics that correctly infer router-level interdomain links and the involved parties.

## 2. MOTIVATION

**Network Modeling and Resilience:** Early models of Internet topology considered each AS a node, and an interdomain link an edge between two ASes [43]. While this simplistic model does not reflect the complexity of Internet interconnection, it has reflected the inferential capabilities of the research community [43]. Our work enables the construction of a router-level map of interdomain connectivity, which will empirically ground efforts to accurately model AS topology evolution.

The capability to correctly identify the interdomain links of a network also enables analysis of network resiliency and robustness in a way not previously possible. For instance, we can use comprehensive traceroutes (from archives or targeted) to estimate which routers, links, and interconnection facilities carry traffic to a significant fraction of the Internet [37], and the potential of an attack or outage to disrupt connectivity.

**Interdomain congestion:** Exploding demand for high-bandwidth content (e.g., streaming video) has fueled recent peering disputes, creating tension among ISPs. Although in the past such disputes have sometimes led to de-peering, the Internet's critical role today inhibits such behavior by networks. Instead, stalled negotiation about who should pay to upgrade interdomain links may lead to congestion as traffic grows beyond capacity, affecting not just the networks in dispute but the sources and destinations of all traffic crossing the link. The public policy community has growing interest in identifying persistent congestion on interdomain links, and its potential harms to consumers. In 2015, the U.S. Federal Communications Commission (FCC) attached

conditions to the AT&T-DirectTV merger [12] that require AT&T report to the FCC performance measurements of its interconnections.

Recent work has explored various methods to identify interdomain congestion: sending a time series of probes to the near and far side of an interdomain link [24]; network tomography [36]; and crowd-sourced throughput measurements [27]. Remarkably, with each of these techniques, the greatest measurement challenge is not detecting the presence of congestion, but identifying interdomain links to probe, and associating observed evidence of congestion to specific interdomain links [24]. Further, placing any isolated signal of congestion in an appropriate context requires a more comprehensive network-wide view of interdomain links and their congestion state – ideally, some kind of weather map of congestion. The surprisingly primitive state of Internet measurement tools renders such an ambition a grand challenge in our field. Accurate identification of network borders is essential to progress on this goal.

# 3. RELATED WORK

Measurement and analysis of the Internet topology has been an active area of research for over two decades. The literature spans the AS, router, and physical topology, with recent focus on the rich connectivity at IXPs (e.g., [3, 1, 14]) and understanding physical infrastructure facilitating connectivity (e.g., [11, 13]). While our work overlaps with research seeking to understand how interconnection is facilitated, our objective is to infer interconnection topology details that enable accurate interpretation of routing and performance measurements.

**Techniques for resolving IP addresses to common routers.** The process of mapping IP addresses to routers is known as alias resolution; it is a critical step in counting and characterizing interdomain connections between networks. Although researchers have made significant progress, this problem is not entirely solved. A survey of existing alias resolution techniques and implementations, including utilizing reverse DNS information, record route data, and IGMP (multicast) protocol messages, is available in [19]. Early techniques [31, 15, 20] probe unused ports on routers and collect the resulting error messages. Probing one interface and getting an error response from a different interface is evidence that the two interfaces belong to the same router.

In 2002, Spring *et al.* developed a method and tools to map individual ISP networks using focused traceroutes and BGP routing tables [40]. Their *Ally* tool was the first to infer that two addresses are aliases if probe packets sent to them produce responses with increasing but appropriately proximate IP ID values, since the IP ID field increments with each packet sent from some routers. RadarGun [5] refined this method by looking for similarities in IP ID time series collected from multiple addresses. More recently, Sherwood *et al.* developed Discarte [39] which uses the IP record-route option, traceroute, and disjunctive logic programming to accurately derive a router-level topology from the IP-interface topology. APAR [16] and kapar [19] used sophisticated graph analysis techniques to infer subnets linking routers, and from that, aliases. Sherry [38] describes iPlane's use of the IP prespecified timestamp option to infer aliases. MIDAR [21] expanded on the IP velocity techniques of RadarGun by implementing an extremely precise ID comparison test based on monotonicity rather than proximity, integrating multiple probing methods from multiple VPs, and using a probe scheduling algorithm that scales to millions of IP addresses. While these efforts have contributed great progress to the field of router-level topology inference, they have focused on structural properties of individual ASes; our work here focuses on the accurate inference of router-level interdomain connectivity between ASes.

**Inference of AS-level connectivity.** Despite its limitations, traceroute can provide a source of AS-level connectivity information, and compensate for a fundamental limitation of interdomain routing (BGP) data. Specifically, BGP routers usually propagate to other routers only their best path to a given destination, leaving a rich database of connectivity knowledge available only to the network's operator. A dense deployment of traceroute VPs does not rely on cooperation of network operators, allowing each VP to observe the best path to a given destination. However, deriving AS-level connectivity from traceroute requires some level of heuristic inference, and caution when applying it. The canonical approach to convert IP-level traceroute output to an AS-level path is to map each observed IP address to the AS originating a BGP announcement of the longest prefix containing that address. But this approach can induce substantial false AS-link inferences, because some routers respond to traceroute packets with a source IP address belonging to a different network [44]. Chen *et al.* proposed a set of robust heuristics to distill missing AS-level links from traceroute data [8], although they did not attempt to attribute router ownership. The primary motivation and focus of these efforts has been extending coverage of the AS-level graph by conservatively supplementing BGP-based AS-level topology information. In contrast, our work focuses on efficient, scalable, and automatable inference of router ownership at network boundaries, although we must navigate all the same pitfalls associated with IP- and AS-level measurement data.

The closest prior works are two efforts led by Mao to build an accurate AS-level traceroute tool [29, 28], a study by Huffaker *et al.* correlating AS- and router-level connectivity [17], and a study by Chandrasekaran *et al.* assessing the performance of paths between servers operated by a large content distribution network [7]. Mao's "AS traceroute" [29] correlated BGP and traceroute views from the same VP, as well as DNS names and WHOIS data to adjust IP-AS mappings so that the traceroute-derived and BGP-observed AS paths were

more congruent. In follow-on work, Mao *et al.* used a dynamic programming technique to adjust IP-AS mappings at a /24 prefix granularity using co-located BGP and traceroute views [28]. However, private interconnection between networks usually uses /30 or /31 prefixes (rather than /24s) to use address space efficiently. The source code for both systems is not publicly available. Huffaker *et al.* evaluated router ownership heuristics based around router alias resolution, relationships, and degree [17]. The best performing heuristic was validated to be correct 71% of the time. Neither of these works attempted to identify interdomain connectivity at the router level. Our work does not require a correlated BGP view or DNS data to correctly infer border routers. Finally, Chandrasekaran *et al.* developed a method for inferring ownership of interfaces observed in pair-wise traceroutes between servers operated by a large content distribution network [7]. Their heuristics are similar to heuristics in this work. However, they acknowledge that the collected IP-level paths are sparse and lack ideal constraints; as a result, not all addresses had an owner inferred. Further, there was no opportunity to use alias resolution to infer a router-level graph as the paths included historically collected data. Our method deliberately collects paths towards every routed prefix to obtain an ideal set of constraints, performs alias resolution to extract routers, and focuses on interdomain links attached to the network hosting the VP.

Concurrent to our work, Marder *et al.* proposed the MAP-IT algorithm to infer router ownership [30]. Similar to Chandrasekaran *et al.* [7], their method works on an interface-level graph, and infers the operator of all addresses observed in the middle of a traceroute path, using IP-AS mappings of adjacent addresses observed before and after an address in a path. Half the interdomain links in our inferences are at the end of paths, with no adjacent addresses in neighbor address space. They validate their method against a published Internet2 topology (100% correct) and using DNS strings on interfaces from Level3's address space (95.4% correct).

# 4. CHALLENGES

Inferring a router's owner is surprisingly complicated because the obvious inference – the origin AS of the longest matching BGP prefix covering the IP address on a router interface – may be incorrect for at least the following seven reasons, many of which are covered in [44, 24]. Yet, lack of a better method leaves researchers using simple but error-prone IP-AS mappings.

**1. The router interface's IP address may be from a neighbor's address space.** When two ASes interconnect with a point-to-point link, they typically assign the link a subnet (usually a /30 or /31 in IPv4) from address space held by one of the two networks. In a customer-provider relationship, the provider usually supplies the address space. When crossing a provider-customer link, the customer's router will usually use

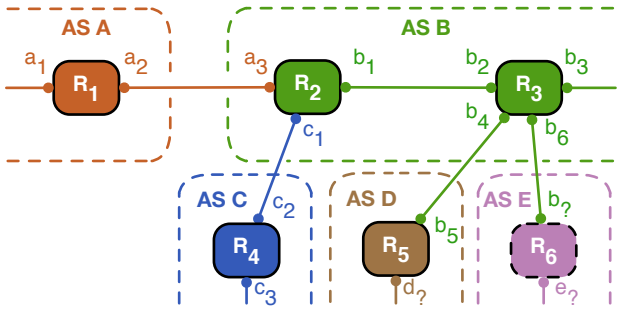


**Figure 1: Responses to traceroute probes depend on the router software implementation and placement in the network. A response from $R_2$ may be naively interpreted as coming from a router operated by AS A, B, or C.**

an address from the provider's space when responding to a traceroute probe, so the first hop in the customer's network in traceroute will usually use an address routed by the provider. There is no convention who supplies the address space in peer-peer relationships. Figure 1 illustrates this challenge: router $R_2$ may respond with address $a_3$ originated by AS A, but be operated by B.

**2. Border routers may use a third-party address when responding to traceroute probes.** A third-party address is an IP address corresponding to an AS that is not on the path toward a destination. A third-party address arises from IETF advice to implementers that a router use the source address of the interface that transmits the response [4]. If a border router's best route to the VP is via a third-party AS, and that AS provides the address space for the interdomain link, then the source address of response will map to the third-party AS. In figure 1, $c_1$ is a third-party address on router $R_2$ in a $R_1 - R_2 - R_3$ path.

Sometimes $R_2$ will respond using $c_1$ regardless of the interface it uses to transmit a response; for example, the source address of an ICMP echo response is the destination address of the corresponding ICMP echo request. We therefore avoid using the source address of an ICMP response that matches the destination address probed, as the position of that address on a router does not necessarily correspond to the interface a traceroute probe arrived at or departed from. For example, if we observed an IP path segment $a_1 - c_1$ in traceroute towards $c_1$, a naive IP-AS interpretation would incorrectly infer an interdomain link between ASes A and C with border routers $R_1$ and $R_2$.

**3. Border routers may be configured to firewall traceroute probes.** Due to security concerns, operators of enterprise networks may configure their border routers to discard packets that do not match a permitted flow at the edge of their network. In figure 1, $R_5$ will respond to traceroute probes with a TTL expired message from address $b_5$, but will not allow subsequent probes into the network which will reveal IP

addresses routed by D. Therefore, the only address observed by traceroute on a router operated by D on a path to D may be $b_5$, originated by B. Similarly, $R_6$ is both configured to discard packets that do not match a permitted flow, and to not send ICMP messages, including the TTL expired message. Therefore, we may not have the ability to observe a border router in traceroute for some neighbor networks.

**4. Virtual routers may use a different responding interface.** Operators can use virtual router functionality to isolate individual routing tables. Each virtual router uses one IP address to form a BGP session with a single neighbor. When the router responds to a traceroute probe, it uses an address from the virtual router that would have forwarded the packet had the TTL not expired, even if that interface does not send the response. In figure 1, if $R_3$ has a virtual router connected to AS D with address $b_4$, and a virtual router connected to AS E with $b_6$, then the router will respond to packets whose TTL has expired towards these ASes with addresses $b_4$ and $b_6$ respectively. Therefore, we require the ability to infer that $b_4$ and $b_6$ belong to the same router $R_3$, although not all routers are responsive to alias resolution probes.

**5. Sibling AS behavior confuses attempts to infer connectivity between organizations.** Different ASes under the same administrative control (siblings) may originate different prefixes. WHOIS-based inference of siblings [6, 18] suffers from limitations in raw WHOIS data, which is not only inconsistently formatted across regions, but also becomes stale if not updated as mergers occur. The only public sibling inferences are derived at three-month intervals, with recognized false and missing inferences [18].

**6. IXP-owned addresses appear inconsistently in paths.** Most interconnection links are automatically established between ASes at an IXP using the IXP's route server [14]. To promote public peering, IXP operators provide a shared peering fabric and associated IP subnet for participants to use. The IXP's own AS may or may not originate this subnet, and/or an IXP member ASes may inadvertently announce it, misleading inferences based on IP-AS mapping.

**7. Multiple ASes may originate a prefix into BGP.** Some prefixes are originated by multiple ASes, which might be siblings or distinct organizations. The more ASes originating a prefix, the more challenging it is to interpret the appearance of a matching IP address in a traceroute path, as the address could be on a router operated by any of the originating ASes.

In prior work, we discussed how a subset of these challenges impacted our ability to measure performance of interdomain links at scale [24]. In this paper, we build and validate a system for mapping the interdomain connectivity of a hosting network. This system supports the CAIDA/MIT congestion project [9], monitoring interdomain links for congestion using 40 VPs in 28 networks as of May 2016.

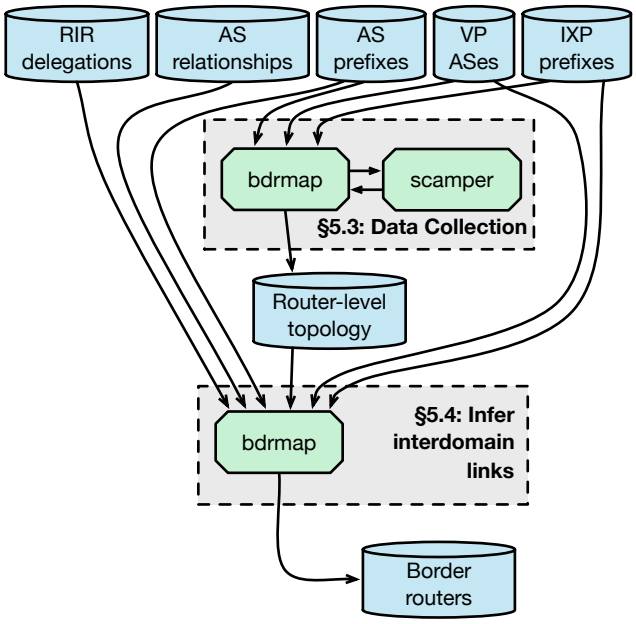


**Figure 2: The `bdrmap` system collects raw data to build an interdomain router-level map for the hosting network (§5.3), and applies heuristics to infer its border routers (§5.4).**

# 5. BORDER MAPPING METHOD

Figure 2 summarizes our approach to border mapping. Our approach begins with assembling routing and addressing data used to inform data collection and analysis. Then, we deploy an efficient variant of traceroute to trace the path from each VP to every routed prefix observed in the global BGP routing system. We apply alias resolution techniques to infer routers and point-to-point links used for interdomain interconnection. We use this collected data to assemble constraints that guide our execution of heuristics to infer router ownership. §5.1 discusses our approach to developing our system, and §5.2 outlines the input data the system requires. §5.3 describes our preliminary construction of a router-level map, and §5.4 explains how we apply heuristics to the collected data to infer routers and their owners. §5.5 presents some limitations of our algorithm, §5.6 reports on our validation, and §5.7 compares our inferences to the public BGP view. Finally, §5.8 discusses how we addressed systems challenges.

We developed a specialized measurement utility that we call `bdrmap` to drive data collection (§5.3) and infer border routers (§5.4). The goal of `bdrmap` is to obtain as much information available about the links observed from a given network toward every other network, in order to constrain our subsequent border router inferences. We implemented `bdrmap` as a driver to *scamper* [23], a parallelized measurement system that efficiently gathers raw traceroute and alias resolution data.

## 5.1 Development Approach

The goal of our system is to correctly identify owners of border routers, with minimal manual work, so that our system will support applied research of network behavior. In §5.6, we report that the system produced inferences for four networks that validated well – 96.3% to 98.9%, depending on the network. However, we emphasize that we did not develop our algorithm with ground truth. Anticipating difficulty obtaining ground truth, we developed our data collection and heuristic methods iteratively, over the course of a year, without validation data. We used DNS-naming, where available, to infer if our methods appeared to yield correct inferences, as well as manual investigation of inferred routers and their neighbors; e.g., border routers with high out-degree to routers in a single neighbor AS usually implied an incorrect inference. We could not perform automated validation using DNS-based heuristics, as we found interdomain links labeled incorrectly as well as links labeled with organization names, rather than AS numbers.

## 5.2 Input Data

We seed our measurements with four data sources: public BGP data to obtain origin ASes for each routed prefix as well as to infer AS relationships between networked organizations; a list of known IXP prefixes; delegation files published by RIRs, and a list of sibling ASes for the networks we measure.

**Public BGP data:** We obtained BGP data from routing table snapshots collected by the Route Views (RV) and RIPE's Routing Information Service (RIS) projects [35, 34]. For each IPv4 prefix of size at least /8 and no smaller than a /24, we recorded the origin ASes we observed in BGP paths to those prefixes. We also used the process described in [25] to infer AS relationships for the same BGP data. This algorithm annotates each AS-link observed in the BGP data with either a peer-peer (p2p) or customer-provider (c2p) label.

**RIR delegation files:** Because some networks do not advertise all prefixes used to number their interfaces, we use the public datasets supplied by the five Regional Internet Registries (RIRs) that report address blocks they have delegated to networks. Some RIRs provide an opaque ID that allows researchers to group prefixes that are delegated to a single organization, although the ID cannot be directly tied to an AS.

**List of IXP prefixes:** We compiled a list of IXP prefixes from database snapshots provided by the PeeringDB and Packet Clearing House (PCH) projects [33, 32]. PeeringDB is a database that allows Internet exchange point operators (IXPs) to record information such as the IP prefixes used to establish public peerings at their IXP, and allows network operators to record the IP addresses they have been assigned by IXP operators to establish peering. PCH records IP subnets, as well as pairs of IP addresses and ASNs used by BGP routers to establish peering at PCH-operated route collectors. Because not all PeeringDB records are correct (they may be entered erroneously and may become stale) and many IXPs are missing from the database, we combined both PeeringDB and PCH data to produce lists of network prefixes used by IXPs to establish peering. Where available, we used IP addresses recorded by operators to validate our ownership inferences (§5.6).

**VP ASes:** For each VP we probed from, we assembled a list of sibling ASes the network hosting the VP uses to organize its routing. We seeded our manual inference with CAIDA's public AS-to-organization mapping file [18] which is derived from information encoded in WHOIS databases, manually added missing siblings, and removed spurious siblings. Sibling inferences are the only input data that requires manual oversight.

## 5.3 Construction of Router-level Topology

Our inference of the router-level topology of the hosting network builds on years of previous work in topology measurement, and proceeds as follows.

**Generate list of address blocks to probe:** We begin by assembling address space blocks that each AS routes. If X originates 128.66.0.0/16, and Y originates 128.66.2.0/24 (a more specific subnet of the /16), we associate the 128.66.0.0 – 128.66.1.255 and 128.66.3.0 – 128.66.255.255 blocks with X, and the 128.66.2.0 – 128.66.2.255 block with Y. As our goal is to infer interdomain connectivity, `bdrmap` does not include any blocks originated by the network hosting the VP.

**Gather traceroutes:** We use the Paris traceroute method [2], sending ICMP echo packets toward each address block in the list, probing each target AS one block at a time to minimize the impact on target ASes. To reduce run-time, `bdrmap` probes multiple target ASes at a time in parallel. For each traceroute, we record the first IP address originated by an external network, and then supply these addresses (the stop set [10]) to other traceroutes involving the same target AS to prevent subsequent traceroutes toward that AS from probing beyond the first interdomain link in the path that has been seen before. For each address block, `bdrmap` first probes the first (.1) IP address. If `bdrmap` does not observe any IP addresses in the traceroute path that map to an external network, or if the only address observed outside the VP's network was the address probed, then `bdrmap` tries the next address in the block, up to five addresses per block, to avoid interpreting potential third-party addresses as neighbors (see §4).

**Resolve IP address aliases to routers:** We use alias resolution techniques to reduce the interface-level graph to a router-level graph reflecting the underlying physical topology. This reduction allows us to include constraints collected for all paths traversing a given router. As `bdrmap` proceeds, it assembles sets of IP addresses that might belong to the same router (candidate alias sets) and probes candidate pairs of IP addresses using alias resolution methods *Ally* and *Mercator*. Ally [40] infers two IP addresses are aliases if the

IP-ID values in responses from the two IP addresses suggest they were derived from the same central counter; we use UDP, TCP, ICMP-echo, and TTL-limited probes to maximize our ability to infer aliases when routers are unresponsive to specific types of probes. Mercator [15] infers two IP addresses are aliases if the source address of ICMP port unreachable responses is the same.

**Infer point-to-point links:** We use the *prefixscan* algorithm [26] to try to confirm that we observe the inbound interface of a router in traceroute, rather than a third-party address. The prefixscan algorithm assumes common peering practice: routers connected with point-to-point links often use /30 or /31 subnets between them. Prefixscan thus infers if an address in a traceroute path corresponds to the interface on a router that received the packet (the inbound interface), by attempting to infer if its /30 or /31 subnet mate is an alias of the previous hop. We also used the Ally and Mercator techniques to infer the same alias pair.

**Limit false aliases:** Because Ally infers false aliases if two IPID time series from different central counters temporarily overlap, we repeat Ally measurements five times at five minute intervals. We only include alias inferences if further Ally measurements do not reject a shared counter hypothesis. We use MIDAR's test [21] that requires non-overlapping IPID samples to strictly increase, rather than be within a fudge factor.

**Build router-level graph:** We use the alias resolution data collected with Ally, Mercator, and Prefixscan to collapse the interface graph to a router graph. When building a router using Ally and Prefixscan inferences through transitive closure (i.e., if $x_1$,$x_2$ and $x_2$,$x_3$ are pairs of inferred aliases, then $x_1$ and $x_3$ are also aliases) we only used pairs of IP addresses where none of the measurements suggested a pair of IP addresses were not aliases. For each router, we then identified which interfaces we observed in ICMP time exceeded messages in traceroute probes, as we focus on these interfaces when making ownership inferences: ICMP time exceeded messages are less likely than other ICMP messages (such as echo reply and destination unreachable) to have third-party source IP addresses. For example, a router will use the destination IP address from an echo request as the source address of an echo reply, providing no indication whose router the address is on.

Parts of our data collection process are similar to Rocketfuel's process [40]. However, Rocketfuel's goal was to map topologies of networks from outside. To gain efficiency, Rocketfuel collected paths to customer prefixes of each network, as these are more likely than other prefixes to cross the network. `bdrmap` infers interdomain connectivity for the network hosting the VP, and uses doubletree's stop set concept to gain efficiency. `bdrmap`'s run-time depends on the diameter and complexity of the hosting network: at 100pps, the shortest run-time we observed was for a research and education network at ≈12 hours. `bdrmap`'s run-time on large U.S. broadband provider networks takes ≈48 hours.

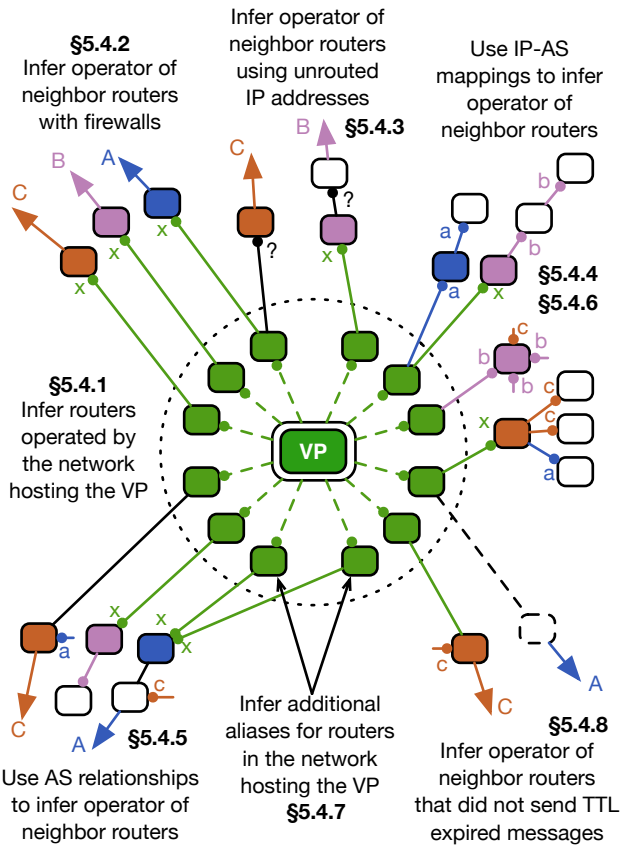


**Figure 3: Conceptual mapping of heuristics we use to infer interdomain routers (§5.4.) Heuristics are numbered in the order that `bdrmap` evaluates them for a given router.**

## 5.4 Algorithm to infer interdomain links

We traverse each router in the graph structure, in the order of observed hop distance from the VP, and apply a set of heuristics (conceptually mapped in figure 3) to infer the owner of each router. The overall approach of `bdrmap` is to first infer the routers operated by the network hosting the VP, and then use as much information as possible to make inferences for routers operated by neighbor networks. `bdrmap` evaluates the heuristics in the order we present them; the heuristics for inferring neighbor routers are ordered by available constraints.

First, (§5.4.1) we try to infer if the router is operated by the AS hosting the VP, as this may allow us to infer an adjacent router with interfaces numbered from address space originated by the network hosting the VP is actually operated by a neighbor AS. Because we traverse the graph in order of observed hop distance, we identify routers operated by the AS hosting the VP (the near side of an interdomain link) before we infer routers operated by neighbor ASes (the far side of an interdomain link). We infer the near side of an interdomain link using only this first step, and all subsequent heuristics infer owners for the far side. In this step,

bdrmap also estimates ownership of address space not originated in BGP by the network hosting the VP. We have used VPs in several networks who do not announce some of their own address space; fortunately these networks usually announce other infrastructure addresses that bdrmap observes nearby in a traceroute. When bdrmap observes an address in traceroute originated in BGP by a VP AS, it assumes all previous addresses in the traceroute path back to the VP were delegated to the network hosting the VP, and identifies the missing address blocks by finding the match for each IP address in the RIR-published delegation files.

Second, (§5.4.2) if we observed no other interfaces adjacent to a router, and we only observed interfaces numbered from address space originated by the network hosting the VP, then we reason about ownership based on the destination networks probed, as we have no other constraints. Third, (§5.4.3) if we visit a router where the address space it uses is unrouted, then we reason about the router based on adjacent networks and destination networks probed, as we have no other constraints. Fourth, (§5.4.4) if we visit a router where two consecutive hops are routed by the same external AS, we reason that the addresses do not represent a third-party AS and infer the router is operated by the external AS. Fifth, (§5.4.5) if we know of an existing relationship from BGP, then we infer that router is likely operated by that network, with exceptions made for third-party addresses also inferred using BGP-derived relationships. Sixth, (§5.4.6) we reason about ownership using IP-AS mappings as we have exhausted better methods (§5.4.1-§5.4.5). Seventh, (§5.4.7) we use our interdomain link inferences to infer additional aliases for near-side routers of interdomain links where we infer a point-to-point link was used to establish connectivity to their neighbor.

Eighth, (§5.4.8) we reason about border routers that did not send TTL-expired messages, as we can now place these routers into a topological context using our inferred router-level graph. Until §5.4.8, we only consider router addresses from ICMP TTL-expired messages when we infer router ownership. As discussed in §4, the source address of an ICMP echo reply is the destination address probed, which can be any of the interfaces on the router, whereas ICMP time exceeded messages usually identify ingress interfaces [26].

Finally, for each router, bdrmap defines *nextas* as the most common provider AS of all destination ASes it probed through that router from that VP, if the router appears in paths to multiple destination ASes. In the first three steps, we use *nextas* as a candidate owner AS for these routers, reasoning that the AS may be providing transit to the ASes reached through these routers.

**5.4.1 Infer routers operated by the network hosting the VP** (figure 4): Interfaces subsequent to a router $R_1$ that are also routed by the network hosting the VP (AS X) usually imply $R_1$ belongs to X. Therefore, in step 1.2, if the IP addresses bdrmap observes are originated by X (as for $R_1$), and bdrmap observes other

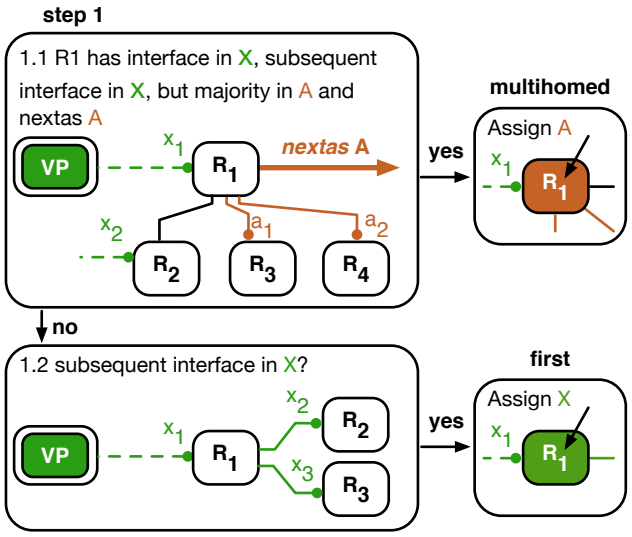


**Figure 4: (§5.4.1) Interfaces subsequent to $R_1$ that are also routed by the network hosting the VP usually imply $R_1$ belongs to the VP.**

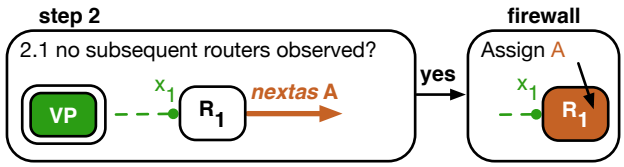


**Figure 5: (§5.4.2): It is not common for an address from A to appear in a traceroute toward A, because a firewall usually discards packets at the edge of A.**

IP addresses originated by X subsequent in the path (e.g., $x_2$ and $x_3$ on routers $R_2$ and $R_3$), then bdrmap infers $R_1$ is operated by X. An exception is when neighbor A is multihomed to X via routers adjacent to each other – step 1.1 in figure 4, routers $R_1$ and $R_2$. If bdrmap observed those routers as $x_1$ and $x_2$ in a traceroute path, and bdrmap observed addresses originated by A also adjacent to $R_1$, we infer A operates both $R_1$ and $R_2$. To limit false inferences, we consider owner AS inferences we would have made for routers subsequent to $R_1$: if any is a customer of X, but not a known neighbor of A, then we infer X operates $R_1$.

As a result of these heuristics, we also infer that any other router bdrmap observes with addresses originated by the network hosting the VP is operated by a neighbor network. That is, the hosting network provided the address space for the interconnection link to the neighbor. In §5.6 we show this logic is nearly always correct.

**5.4.2 Infer operator of neighbor routers with firewalls** (figure 5): It is not common for an address from AS A to be recorded in a traceroute path toward A, because a firewall usually discards probe packets at the edge of A. Therefore, the last router observed by bdrmap

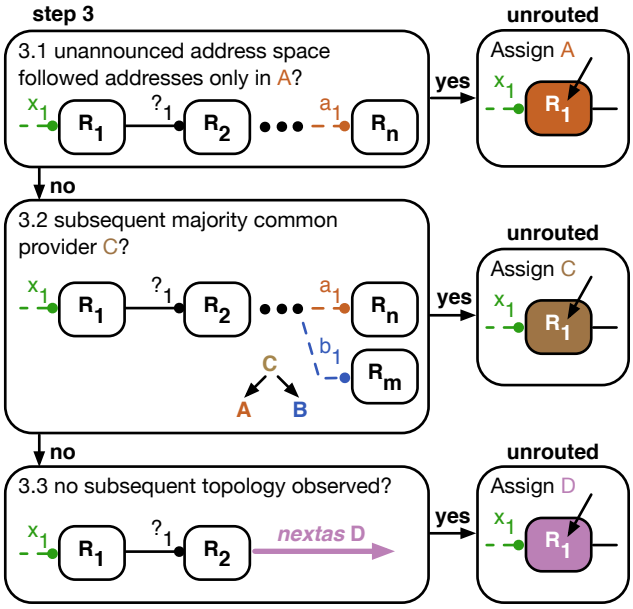


**Figure 6: (§5.4.3): Some operators do not route infrastructure IP addresses in BGP, so we infer their routers based on subsequent routed addresses in traceroute paths.**

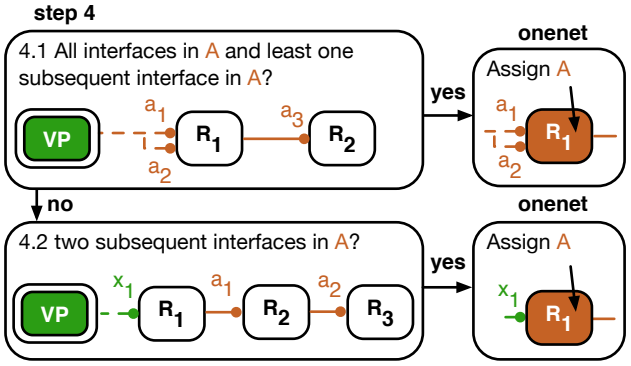


**Figure 7: (§5.4.4) IP-AS mappings can lead to incorrect inferences in the presence of third-party addresses, but subsequent interfaces routed by the same network suggest $a_1$ is not a third-party address.**

in a traceroute path to A (and A's siblings) is usually A's edge router, as is the case for router $R_1$ operated by A. If `bdrmap` observes $R_1$ with interfaces originated by the network hosting the VP (X), and no adjacent interfaces in A, it assumes X provided the address for interconnection to A and infers $R_1$ is operated by A.

**5.4.3 Infer operator of neighbor routers that use unrouted IP addresses** (figure 6): Some operators do not advertise routes to the IP addresses on some of their routers. This practice can hamper inference of border routers because the origin AS can provide constraints to narrow down the owner of the router. There are two related scenarios that `bdrmap` addresses.

A neighbor router $R_1$ might have addresses that our VP-hosting network (X) originates in BGP, but the addresses observed on subsequent routers (e.g., $R_2$) might be unrouted. Or, not illustrated in figure 6, $R_1$ might have unrouted addresses but be connected to a router we have previously inferred to belong to X. In both cases, `bdrmap` assembles the set of ASes that originate the first routed interfaces in traceroute paths after $R_1$, and uses these interfaces to infer $R_1$'s operator. If there is only one AS (step 3.1), `bdrmap` infers that AS operates $R_1$. If there are multiple (step 3.2), for each AS in the set `bdrmap` identifies the providers of the AS using BGP-derived relationships, and infers that $R_1$ is operated by the most frequent provider AS among the provider AS set, reasoning that this AS provides transit to networks observed by `bdrmap`. If `bdrmap` does not observe any routed addresses in traceroute paths after the border router, it infers that *nextas* operates $R_1$.

**5.4.4 Use IP-AS mappings to infer operator of neighbor routers** (figure 7): Using IP-AS mappings to infer ownership can be error-prone because a router may respond with an IP address that represents a third-party (see §4). However, we hypothesize that we are unlikely to observe two third-party addresses in a row, so if we observe addresses originated by the same AS at two consecutive hops, we infer that AS is the interconnecting party.

Therefore, if all interface addresses `bdrmap` observes on router $R_1$ map to the same origin AS A in BGP, and at least one adjacent router $R_2$ subsequent in a traceroute path also has an address in A, then we infer A operates $R_1$ (step 4.1). Similarly, if `bdrmap` observes a border router $R_1$ operated by neighbor A using addresses that the network hosting our VP (X) originates in BGP, and `bdrmap` observes two consecutive routers $R_2$ and $R_3$ with interface addresses originated in BGP by external network A, then we also infer A operates $R_1$ (step 4.2).

**5.4.5 Use AS relationship inferences to infer operator of neighbor routers:** (figure 8): If we do not observe two hops with addresses from the same AS (§5.4.4), we have less router-level information to reason about router ownership. Therefore, we use AS relationships (§5.2) to guide router operator inference.

We first infer if the IP-AS mapping of a router interface is a third-party AS mapping, as follows. If `bdrmap` observed an address on $R_2$ that A originates in BGP, but `bdrmap` only observed $R_2$ on paths toward B, it is possible the address `bdrmap` observed on $R_2$ is a third-party address. If A is a provider of B (per our BGP inference), then we infer $R_2$ used a route from their provider to respond to traceroute (a third-party address) and that AS B operates $R_2$. If (step 5.1) `bdrmap` only observed addresses that the network hosting our VP (X) originated in BGP on a router $R_1$ preceding $R_2$, then we also infer that AS B operates $R_1$. Similarly,

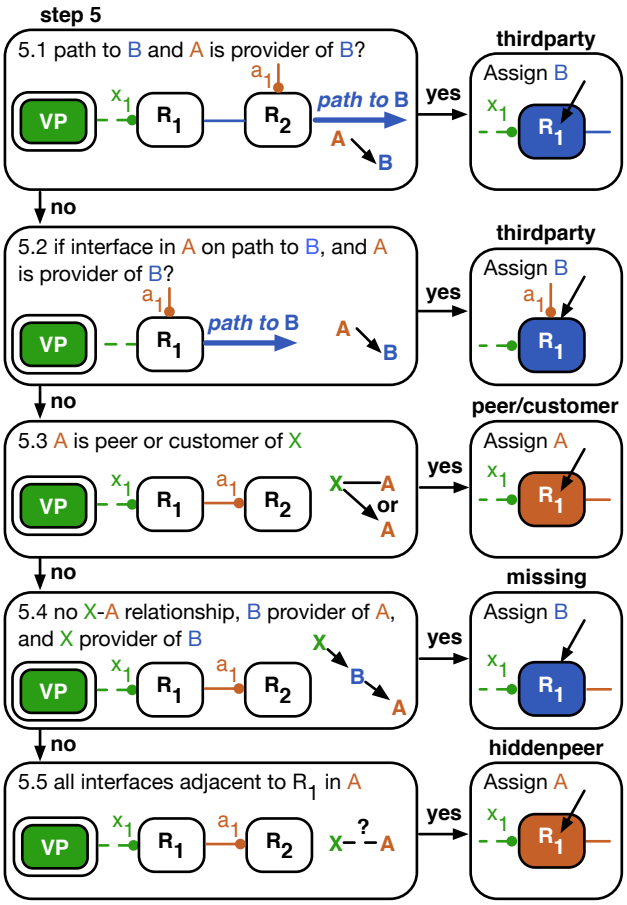


**Figure 8: (§5.4.5) AS relationship inferences derived from public BGP data assist in identifying the operator of a router responding with a third-party addresses, as well as known peers and customers of the network hosting the VP.**

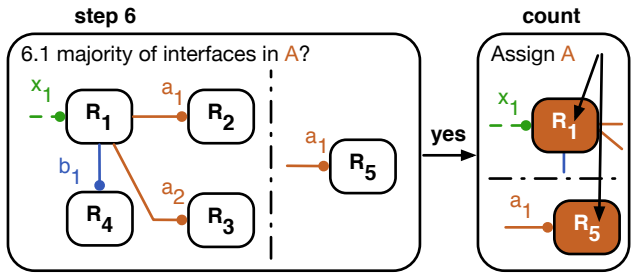


**Figure 9: (§5.4.6) If there are multiple possible IP-AS mappings, we infer the neighbor router is operated by the AS with the most subsequent interfaces.**

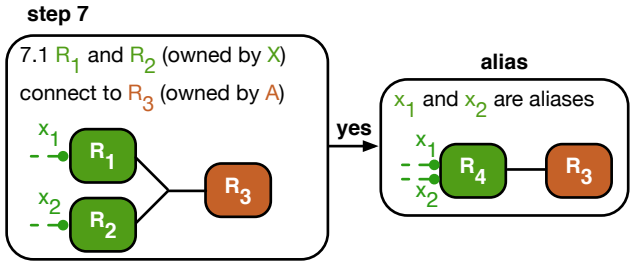


**Figure 10: (§5.4.7) Interdomain links are usually point-to-point links between two routers, so multiple apparent IP links to the same neighbor router are likely to be caused by IP aliases.**

if (step 5.2) `bdrmap` instead observed $R_1$ with address space originated by A only on paths to B, then we infer B operates $R_1$.

For steps 5.3, 5.4, and 5.5, we start with $R_1$ which we observed using an address from the network hosting our VP (X). If (step 5.3) adjacent interfaces only have addresses originated in BGP by a known peer or customer A, then we infer that AS A operates $R_1$. We make these inferences after detecting third-party addresses because a neighbor might use a third-party address that happens to be a known peer or customer of X. If (step 5.4) adjacent interfaces only have addresses originated in BGP by a network A which is not an inferred peer or customer of X, but B is a provider of A and X is a provider of B, then `bdrmap` infers AS B operates $R_1$; sibling AS relationships (§4) can cause this scenario, where the same organization operates ASes A and B. If none of the above hold, and (step 5.5) subsequent interfaces are originated in BGP by a single AS A, then we infer that A operates $R_1$.

**5.4.6 Use IP-AS mappings to infer operator of neighbor routers in ambiguous scenarios** (figure 9): Some neighbor routers are also border routers to other networks. This impacts analysis of traceroute, as multiple adjacent IP addresses can be originated in BGP by different ASes. In figure 9, $R_1$ precedes interfaces $a_1$ and $a_2$ on $R_2$ and $R_3$ routed by A, and $b_1$ routed by B on $R_4$; both $R_1$ and $R_4$ could be border routers operated by different networks. When (step 6.1) we observe a neighbor router $R_1$ using an address from the network hosting our VP (X) to form the interdomain link with, and multiple adjacent IP addresses originated in BGP by different ASes, then `bdrmap` infers the operator of $R_1$ to be the AS with the most adjacent IP addresses. If there is a tie, we select the first AS with a known relationship (per our BGP inference) to the VP. Otherwise, if the addresses `bdrmap` observes on a border router $R_5$ are originated by a different AS, then `bdrmap` infers the operator of $R_5$ to be operated by that AS.

**5.4.7 Infer additional aliases for border routers** (figure 10): We undertake a final analytical alias resolution step to address the cases where we were unable to resolve likely aliases because the routers did not respond favorably to our alias resolution probes. That is, the routers did not respond to UDP probes with a common source-IP address, and did not assign IP-ID values to responses from a single central counter. We assume

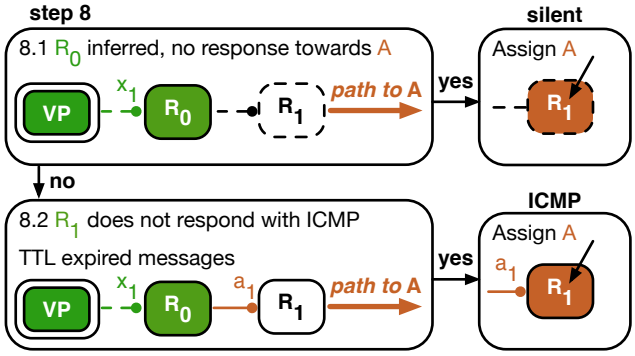


**Figure 11: (§5.4.8) If a known neighbor (per BGP) firewalls selected probes from entering their network, but paths towards that AS always visit the same VP border router $R_0$, we infer the neighbor is connected to the VP router.**

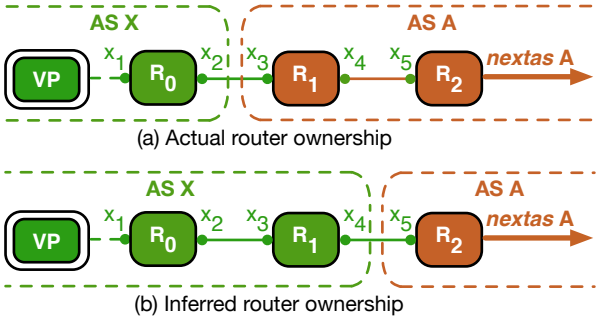


**Figure 12: If an AS uses provider-aggregatable address space from their provider on interfaces on their internal routers, `bdrmap` may incorrectly infer the position of interdomain link.**

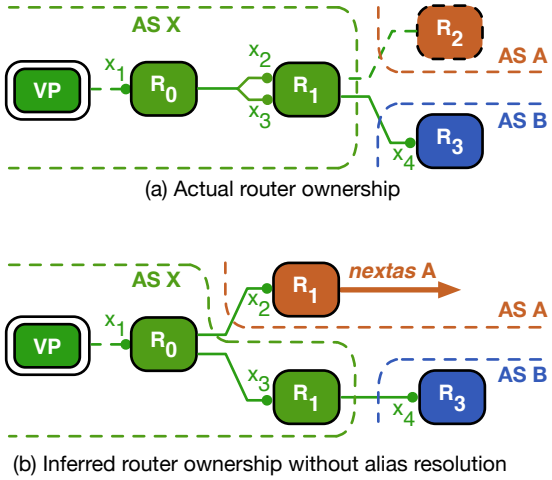


**Figure 13: If router $R_1$ responds with different IP addresses depending on the destination probed, and those addresses are not inferred to be aliases, `bdrmap` may incorrectly infer the position of an interdomain link.**

that a neighbor router $R_3$ operated by A connects to a single router in the VP's AS by point-to-point link, and (step 7.1) collapse single interface routers $R_1$ and $R_2$ we previously inferred to be operated by X (in step 1.2) into into a single border router $R_4$.

**5.4.8 Infer operator of neighbor routers without TTL expired messages** (figure 11): Some operators configure their routers to never send TTL expired messages, so these routers are not processed by previous heuristics. Some routers respond to probes using other messages, such as ICMP echo replies and destination unreachables. The remainder of them remain silent. We distinguish these cases from routers that rate limit their response (are periodically responsive) by comparing the set of VP neighbors we inferred borders for, from those known to exist through public BGP data.

First, we assemble a list of neighbors observed in BGP for the network hosting the VP for which we have not inferred any interdomain links, and the traceroutes toward those ASes. We then process the traceroutes for each AS as a set. If the final router observed by `bdrmap` in the network hosting the VP was always the same router (step 8.1), and `bdrmap` observed no other interfaces after that router when tracerouting toward a single AS, then we infer the neighbor AS connects to that router. In this scenario, the AS has disabled ICMP time exceeded messages and blocked our probe packets. While we cannot identify this (silent) router, we can identify where it connects to the VP. Otherwise, if the final router `bdrmap` observed in the network hosting the VP was always the same router (step 8.2), and `bdrmap` observed ICMP echo reply or destination unreachable messages with a source address that maps to that neighbor AS, then we infer that neighbor AS connects to the specific VP router. In this scenario, the neighbor's border router has firewalled our probes from entering the network, but sends specific ICMP messages in response to our probes.

## 5.5 Limitations

`bdrmap` relies on the router-level map providing adequate constraints so that our heuristics may correctly infer border routers and their owners. However, not all inferences we make are correct, as there are multiple possible explanations for the topological arrangements observed by `bdrmap`. In this section, we focus on topological limitations that can result in incorrect inferences in where the network hosting the VP ends.

A provider may delegate provider-aggregatable (PA) address space to their customer, and the customer may configure part of that address space on their router interfaces. In figure 12, AS A operates routers $R_1$ and $R_2$, and uses PA address space from provider X on those routers. When `bdrmap` observes these interfaces, it infers an interdomain link between routers $R_1$ and $R_2$ operated by X and A, respectively, instead of correctly

| | R&E network | | | | Large access network | | | | Tier-1 network | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cust | peer | prov | trace | cust | peer | prov | trace | cust | peer | prov | trace |
| Observed in BGP | 30 | 2 | 1 | | 652 | 26 | 5 | | 1644 | 70 | 0 | |
| Observed in `bdrmap` | 28 | 2 | 1 | 82 | 599 | 26 | 5 | 65 | 1602 | 58 | 0 | 58 |
| Coverage of BGP | | 93.9% | | | | 92.2% | | | | 96.8% | | |
| 1. Multihomed to VP | | | | | 0.4% | 2.0% | | | | | | |
| 2. Firewall | 51.4% | | | 36.5% | 60.4% | 5.9% | | 39.1% | 64.7% | 9.2% | | 62.2% |
| 3. Unrouted interface | | | | 1.0% | 0.1% | | | 3.0% | 0.5% | 5.0% | | 8.5% |
| 4. IP-AS (onenet) | 8.6% | 100% | 100% | 31.2% | 3.9% | 39.2% | 87.5% | 26.3% | 6.7% | 36.9% | | 2.4% |
| 5. Third party | 8.6% | | | 2.1% | 0.4% | | | 5.3% | 0.2% | | | 15.9% |
| 5. AS relationship | 20.0% | | | | 29.4% | 41.2% | | | 20.8% | 34.0% | | |
| 5. Missing customer | | | | | | | | | 0.2% | | | |
| 5. Hidden peer | | | | 24.0% | | | | 24.1% | | | | 4.9% |
| 6. Count | | | | 4.2% | 0.6% | 7.8% | 8.4% | | 0.8% | 7.1% | | 2.4% |
| 6. IP-AS | 2.9% | | | 1.0% | 0.5% | | 4.2% | 2.3% | | 2.1% | | 3.7% |
| 8. Silent neighbor | 8.6% | | | | 2.7% | 3.9% | | | 4.0% | 5.0% | | |
| 8. Other ICMP | | | | | 1.5% | | | | 2.0% | 0.7% | | |
| Neighbor routers | 35 | 2 | 3 | 96 | 775 | 51 | 24 | 133 | 2088 | 141 | 0 | 82 |

**Table 1: Evaluation of `bdrmap` heuristics against BGP observations and AS relationship inferences for three networks. We validated the R&E and large access network inferences against ground truth (§5.6). For these networks, between 92.2% and 96.8% of BGP-observed links had a neighbor border router inferred by `bdrmap`. `bdrmap` also inferred interdomain links that were not BGP-visible, and these links and border routers are reported in the "trace" columns. For all three networks, the firewall heuristic inferred most customer routers, i.e., the last interface observed by `bdrmap` was the ingress interface address assigned by the network hosting our VP on their border router.**

inferring the interdomain link between routers $R_0$ and $R_1$. This occurs because `bdrmap` first infers X operates $R_1$ as adjacent interface $x_5$ on $R_2$ implies $R_1$ is operated by X (§5.4.1, figure 4). Note that `bdrmap` correctly infers $R_2$ is operated by A using the firewall heuristic in §5.4.2 or the customer heuristic in §5.4.5, but the inferred location of the interdomain link is incorrect.

Similarly, a router may respond to traceroute with different IP addresses, particularly if there are multiple load balanced paths involving the router, or if the operator configures virtual routers to establish BGP sessions with neighbors and the router uses a single interface from each virtual router to respond to traceroute (§4). In figure 13, $R_1$ is owned by AS X, and $R_1$'s interfaces $x_2$ and $x_3$ were observed in traceroute paths preceding routers $R_2$ and $R_3$, respectively. However, if `bdrmap` did not infer $x_2$ and $x_3$ to be aliases, and if `bdrmap` only observed $x_2$ in paths towards A, `bdrmap` incorrectly infers $x_2$ to belong to a router operated by AS A. `bdrmap` correctly infers $x_3$ to belong to a router operated by X, as adjacent interface $x_4$ implies $x_3$ is on a router operated by X (§5.4.1, figure 4). If `bdrmap` had correctly inferred $x_2$ and $x_3$ to be aliases, it would have correctly inferred the existence of a silent router $R_2$ operated by A, and that $R_1$ is operated by X.

## 5.6 Validation against ground truth

During the validation phase, we contacted 10 networks seeking ground-truth, and received data from 4: a research and education (R&E) network, a large access network, a Tier-1 network, and a small access network. When we received a response declining to provide validation, the response highlighted commercial sensitivity, as commercial networks view their interconnection as proprietary, particularly at the router-level. We only asked about interdomain links we observed, as we assumed the networks would not provide ground truth on links we had missed.

We were able to validate all our neighbor router inferences for the R&E and small access networks; the R&E network supplied us a sanitized router configuration dump, and the small access network operator manually checked inferred adjacencies. Validation for the larger networks was challenging, both because the engineers we spoke with only had detailed visibility into part of the network, and because sometimes the address we observed in traceroute was not the address that the interconnecting party used on the interface peering with the network offering us validation.

**R&E network:** We obtained a sanitized router configuration dump from an R&E network. In January 2016, the network consisted of 17 routers with BGP sessions involving 48 ASes and three IXPs. Of the 45 interdomain links we inferred outside of the IXP, 44 correctly identified the presence of an interdomain link and the correct AS. We also correctly inferred the location of a further three interdomain links with ASes that disabled any form of ICMP response. Further, we validated the interdomain links established via route servers at the three IXPs where the network was present by using the IXP-published information on which ASes are present and the IP addresses they use. Of the 88 ASes for which we had validation data, we correctly identified the AS for 82, with 2 more identifying a sibling of the correct

AS, i.e. 84/88 (95.4%). Overall, we correctly identified 131 of 136 interdomain links (96.3%).

**Large access network:** We obtained a filtered snapshot of a router configuration dump for the backbone of a large access network. The backbone connects the network's largest peers and customers, while other parts of the network connect its enterprise customers. We sent an operator of this network a list of all IP addresses we believed to be on border routers, and received in return ground truth on whether each address was on an internal or interdomain link. For those that were an interdomain link, we received ground truth on which AS the link was with. We evaluated the correctness of inferences from three VPs within this network, where each VP observed between 188 and 198 interdomain links. We correctly inferred between 97.0% and 98.9% of interdomain links and associated neighbor networks with an AS reflecting the correct organization.

**Tier-1 network:** We sent an operator of a Tier-1 network a list of all IP addresses we believed to be on their neighbors' side of an interdomain link, and received validation data for 2691 interfaces. Using this data, we found 2584 of the 2650 routers we inferred were neighbor routers identified with an AS reflecting the correct organization (97.5%).

**Small access network:** An operator at a small access network manually validated our inferences for 14 routers with less than 12 interdomain links, and provided a sanitized dump of their interconnection partners for three routers at interconnection facilities. In total, they supplied validation data for 293 interconnections we inferred, with 283 (96.6%) reported as correct.

## 5.7 Comparing traceroute and BGP views

Table 1 reports the coverage of interdomain connectivity observed by one VP in each of three different networks. We also used `bdrmap` to infer border routers of 25 other networks, with similar results. Table 1 categorizes observed links into those observed in BGP (broken down by inferred AS relationship), and those inferred only in `bdrmap`'s traceroutes. `bdrmap` observed routers connecting between 92.2% and 96.8% of BGP-observed networks, with links for providers and peers the best represented (at least one neighbor router observed for all peers and providers for two of the three networks). The rows in Table 1 list the results of the heuristics in the order in which `bdrmap` applies them against an inferred neighbor router. `bdrmap` infers at least half of inferred customer routers with the *firewall* heuristic: i.e., `bdrmap` did not observe any interface originated by the customer in a traceroute. The challenge in inferring customers with traceroute is further illustrated by the relative use of the *onenet* and *silent* heuristics: only 3.9% – 8.6% of customers in these three networks had two consecutive interfaces in their network, but these heuristics inferred 36.9% of peers and provider routers. Similarly, `bdrmap` inferred that 2.7% – 8.6% of customers had disabled time exceeded messages and firewalled our probes.

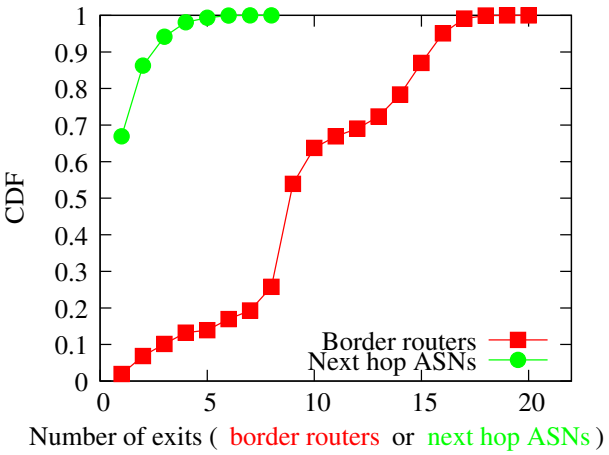


**Figure 14: Distribution of number of border routers and next hop ASes observed on paths to all routed prefixes from 19 VPs in a large access network. We found significant redundancy in the number of egress points towards a destination prefix. Most prefixes are routed via the same next hop AS irrespective of VP location.**

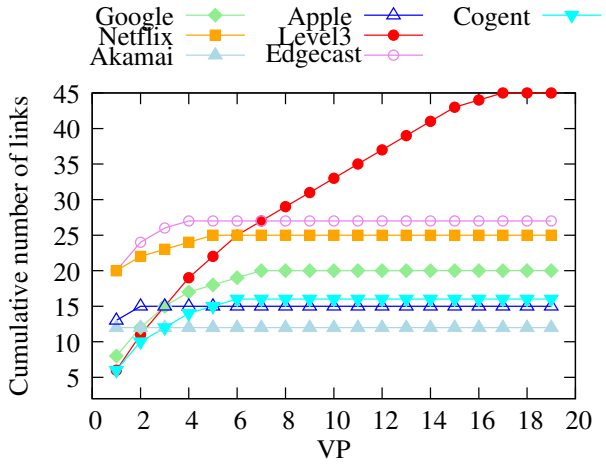


**Figure 15: Marginal utility of VPs in discovering interconnectivity of a large access network. Some networks require a dense VP deployment for `bdrmap` to uncover their links.**

## 5.8 Supporting resource-limited devices

Some of the densest measurement infrastructure deployments use extremely resource-limited devices. A mapping of IP addresses to originating ASes, stop lists, as well as state to guide alias resolution can require substantial CPU time and memory storage relative to resources available on some devices that could use our technique. For example, the most powerful RIPE atlas probes and SamKnows Whitebox measurement devices use a MIPS-based processor operating at 400Mhz with 32MB of RAM and 4MB flash, while `bdrmap` required approximately 150MB of RAM to operate.

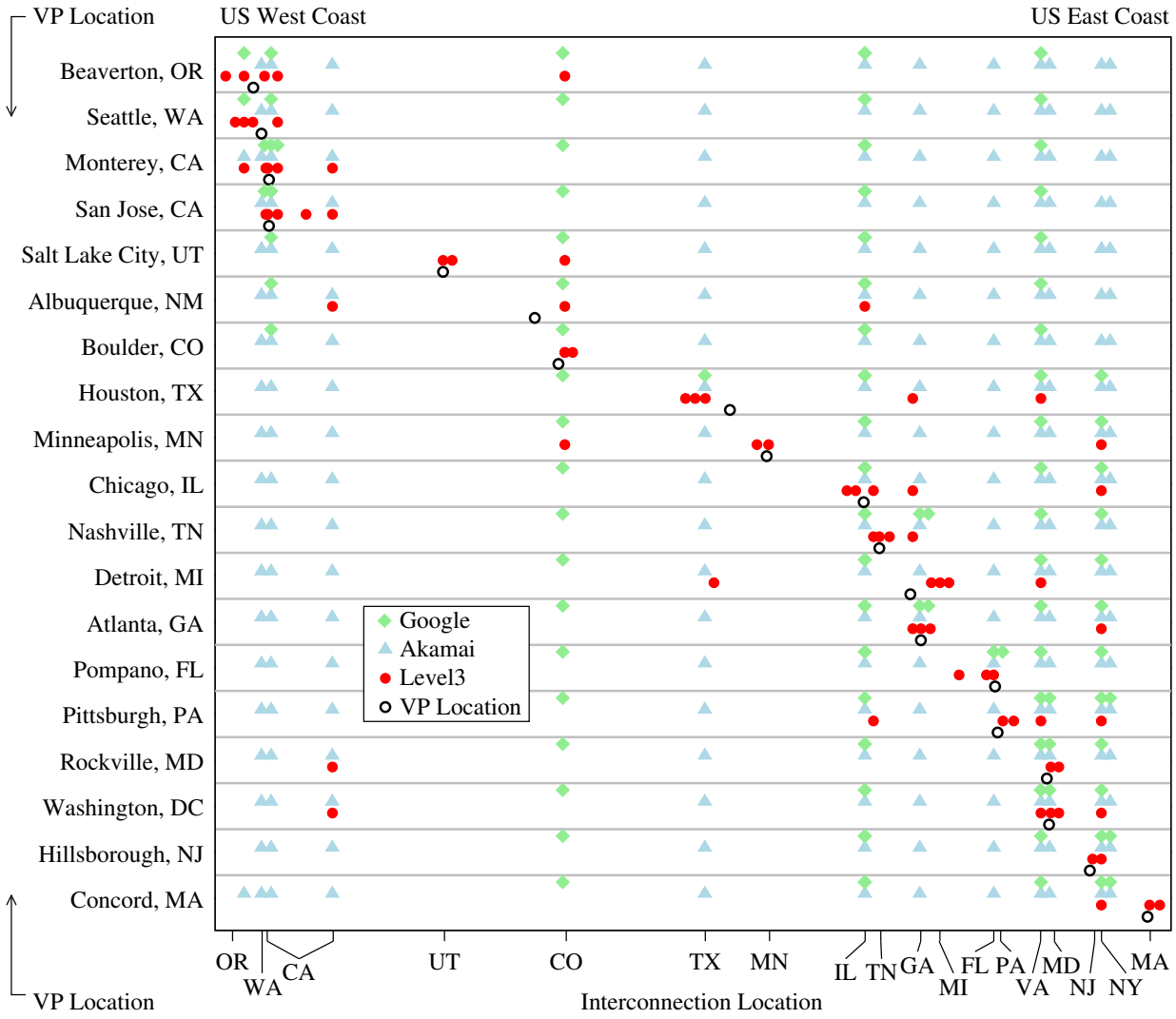


**Figure 16: The impact of the VP's geographic location (hollow circles) on the interdomain links (filled symbols) observed by the VP for a large access network. Each row plots the longitude of the VP-side of an interdomain link observed by a single VP in the access network. For Level3, the location of the VP strongly influences the interdomain links observed. Because Akamai originates some prefixes exclusively across individual interdomain links, all VPs observe all interdomain links.**

We extended our data collection tools so that the prober (scamper) can run on low-resource devices and call back to a centrally-operated system, which has access to greater compute resources and runs `bdrmap`. We used VPs from the BISmark project [41] to prove the system can feasibly run on other densely deployed measurement systems. Similar to the SamKnows systems, the BISmark systems are OpenWrt-based with a 450Mhz MIPS processor, 64-128MB of RAM, and 16MB of flash storage. During our measurement, the maximum CPU consumption used by scamper on the BISmark VPs was 3%, and it used 3.5MB of RAM, or 11% of the total memory on the RIPE atlas and SamKnows Whitebox devices. We use data we collected from BISmark in §6.

## 6. INTERCONNECTION INSIGHTS

We used 19 VPs deployed in a single large U.S. access network to measure the diversity of its interdomain paths in January 2016. Specifically, we computed the number of border routers and next hop ASes traversed on paths from these VPs towards all routed prefixes (figure 14). This measure reflects network resiliency; many border routers and next hop ASes able to reach a destination implies many redundant paths available in case of network disruptions. The diversity in interconnection is remarkable: we inferred that fewer than 2% of prefixes left this access network via the same border router from each VP. For 73% of prefixes, we observed 5–15 distinct border routers, and 13% of prefixes had more than 15 exit points. These numbers suggest astonishing

resiliency and redundancy toward most of the IPv4 Internet. The AS-level density is lower: we inferred that most (67%) prefixes are routed via the same next hop AS regardless of VP location.

A driving motivation for this work is the ability to accurately measure interdomain congestion and network resiliency, which requires comprehensive coverage of interdomain links of a network being studied. We thus need to quantify how many VPs we need in a hosting network, and where we need them, to discover all router-level interconnections. Using the same large access network, we measured the marginal utility of additional VPs for discovering interdomain links with two large transit providers and five CDNs (figure 15). Akamai and Level3 appear to be at two extremes in terms of our ability to discover their interconnections from VPs in the access network, revealing an interesting difference in routing and interconnection strategies across these networks. Specifically, a single VP in the access network observes all the network's interconnections with Akamai, because Akamai advertises certain prefixes only at specific interconnection points. In contrast, each additional VP reveals progressively more interconnections with Level3, consistent with Level3 advertising most prefixes at each interconnection point so that the access network can hand off traffic toward a prefix at its closest interconnection point – hot potato routing [42]. We observed all 45 router-level interconnections this access network has with Level3 (as of January 2016), but required 17 VPs in diverse geographical regions across the U.S. to do so.

Figure 16 illustrates that it is not just number of VPs but their geographical diversity within the VP that affects the number of distinct interdomain links observed for a large access network. For this access network, we used the location information embedded in reverse DNS mappings for IP addresses on their border routers to infer their geographical location. Akamai's announcement policy allows a single VP anywhere to identify all points of interconnection to this network. Visibility of interconnections to Google requires west and east coast VPs, but visibility to all of Level3's interconnections requires VPs spread across the U.S. due to hot potato routing.

# 7. CONCLUSIONS

It would shock most people that something as basic as connections between TCP/IP networks remains so opaque to researchers and regulators, and the range of research that is handicapped by lack of this measurement capability. Although we have only taken the first step – identifying interdomain links directly connected to and visible from the network hosting a measurement vantage point – it is transformative for Internet mapping research. Our method uses targeted traceroutes, detailed knowledge of traceroute behavior, and codification of topological constraints in a structured set of heuristics, to correctly identify network boundaries at the router-level. We applied our method to reveal the tremendous density and diversity of router-level interconnection between some pairs of ASes. We explored the parameter space of this method by computing the marginal gains of VP deployment inside one large access ISP, and the geographical diversity of VPs required to achieve a full view of this ISP's interconnectivity. This topology measurement and analysis capability forms an essential cornerstone of the system we are developing to map interdomain performance measurements at Internet scale, and we publicly release our source code.

## Acknowledgments

We thank the operators who discussed aspects of their network's operations, the Research and Education Advanced Network New Zealand (REANNZ), Guilherme Martins (Princeton) who deployed scamper on BISmark nodes to support this work, and the anonymous reviewers for their feedback. BISmark was supported by NSF CNS-1422680 and CNS-1405781. This work was supported by NSF CNS-1414177 and CNS-1413905, by the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHS S&T/CSD) via contract number HHSP233201600010C, and by a grant from Comcast, but this paper represents only the position of the authors.

# 8. REFERENCES

[1] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a large European IXP. In *SIGCOMM*, 2012.

[2] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with Paris traceroute. In *IMC*, Oct. 2006.

[3] B. Augustin, B. Krishnamurthy, and W. Willinger. IXPs: Mapped? In *IMC*, 2009.

[4] F. Baker. Requirements for IP version 4 routers, June 1995.

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

[6] X. Cai, J. Heidemann, B. Krishnamurth, and W. Willinger. Towards an AS-to-organization map. In *IMC*, pages 199–205, Nov. 2010.

[7] B. Chandrasekaran, G. Smaragdakis, A. Berger, M. Luckie, and K.-C. Ng. A server-to-server view of the Internet. In *CoNEXT*, 2015.

[8] K. Chen, D. R. Choffnes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao. Where the sidewalk ends: Extending the Internet AS graph using traceroutes from P2P users. In *CoNEXT*, Dec. 2009.

[9] K. Claffy, A. Dhamdhere, M. Luckie, D. Clark, and S. Bauer. Mapping interconnection in the Internet: Colocation, connectivity and congestion. http://www.caida.org/funding/nets-congestion/.

[10] B. Donnet, T. Friedman, and M. Crovella. Improved algorithms for network topology discovery. In *PAM*, pages 149–162, Mar. 2005.

[11] R. Durairajan, P. Barford, J. Sommers, and W. Willinger. InterTubes: A study of the US long-haul fiber-optic infrastructure. In *SIGCOMM*, Aug. 2015.

[12] Federal Communications Commission. MB Docket No. 14-90), Memorandum Opinion and Order, FCC 15-94, July 2015. https://apps.fcc.gov/edocs_public/attachmatch/FCC-15-94A1.pdf.

[13] V. Giotsas, G. Smaragdakis, B. Huffaker, M. Luckie, and k claffy. Mapping peering interconnections to a facility. In *CoNEXT*, 2015.

[14] V. Giotsas, S. Zhou, M. Luckie, and k claffy. Inferring multilateral peering. In *CoNEXT*, Dec. 2013.

[15] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *INFOCOM*, pages 1371–1380, Mar. 2000.

[16] M. Gunes and K. Sarac. Analytical IP alias resolution. In *IEEE International Conf. on Communications*, pages 459–464, 2006.

[17] B. Huffaker, A. Dhamdhere, M. Fomenkov, and kc claffy. Toward topology dualism: Improving the accuracy of AS annotations for routers. In *PAM*, Apr. 2010.

[18] B. Huffaker, K. Keys, R. Koga, M. Luckie, and kc claffy. CAIDA inferred AS to organization mapping dataset. https://www.caida.org/data/as-organizations/.

[19] K. Keys. Internet-scale IP alias resolution techniques. *CCR*, 40(1):50–55, 2010.

[20] K. Keys. iffinder alias resolution tool, 2012. http://www.caida.org/tools/measurement/iffinder/.

[21] K. Keys, Y. Hyun, M. Luckie, and k claffy. Internet-scale IPv4 alias resolution with MIDAR: System architecture. *IEEE/ACM Transactions on Networking*, 21(2):383–399, Apr. 2013.

[22] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie. Sampling biases in IP topology measurements. In *INFOCOM*, Apr. 2003.

[23] M. Luckie. Scamper: a scalable and extensible packet prober for active measurement of the Internet. In *IMC*, pages 239–245, Nov. 2010.

[24] M. Luckie, A. Dhamdhere, D. Clark, B. Huffaker, and k claffy. Challenges in inferring Internet interdomain congestion. In *IMC*, Nov. 2014.

[25] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and k claffy. AS relationships, customer cones, and validation. In *IMC*, Oct. 2013.

[26] M. Luckie and kc claffy. A second look at detecting third-party addresses in traceroute traces with the IP timestamp option. In *PAM*, Mar. 2014.

[27] M-Lab Research Team. ISP interconnection and its impact on consumer Internet performance - a measurement lab consortium technical report. http://www.measurementlab.net/publications/, 2014.

[28] Z. M. Mao, D. Johnson, J. Rexford, J. Wang, and R. Katz. Scalable and accurate identification of AS-Level forwarding paths. In *INFOCOM*, Mar. 2004.

[29] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate AS-level traceroute tool. In *SIGCOMM*, pages 365–378, Aug. 2003.

[30] A. Marder and J. M. Smith. MAP-IT: Multipass accurate passive inferences from traceroute. In *IMC*, 2016.

[31] J.-J. Pansoit and D. Grad. On routes and multicast trees in the Internet. In *SIGCOMM*, 1998.

[32] Packet Clearing House. https://prefix.pch.net/applications/ixpdir/menu_download.php.

[33] PeeringDB. https://www.peeringdb.com/.

[34] RIPE RIS. http://www.ripe.net/ris/.

[35] U. Oregon Route Views Project. http://www.routeviews.org/.

[36] S. Roy and N. Feamster. Characterizing correlated latency anomalies in broadband access networks. In *SIGCOMM*, pages 525–526, Aug. 2013.

[37] M. Sanchez, F. Bustamante, B. Krishnamurthy, W. Willinger, G. Smaragdakis, and J. Erman. Inter-domain traffic estimation for the outsider. In *IMC*, Nov. 2014.

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

[39] R. Sherwood, A. Bender, and N. Spring. DisCarte: A disjunctive Internet cartographer. In *SIGCOMM*, pages 303–314, Aug. 2008.

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

[41] S. Sundaresan, S. Burnett, N. Feamster, and W. de Donato. BISmark: A testbed for deploying measurements and applications in broadband access networks. In *USENIX*, June 2014.

[42] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. In *SIGMETRICS*, June 2004.

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

[44] Y. Zhang, R. Oliveira, H. Zhang, and L. Zhang. Quantifying the pitfalls of traceroute in AS connectivity inference. In *PAM*, 2010.