Ingress Point Spreading: A New Primitive for Adaptive Active Network Mapping

Guillermo Baltra, Robert Beverly, and Geoffrey G. Xie

Naval Postgraduate School, Monterey, CA
{gbaltra,rbeverly,xie}@nps.edu

Abstract. Among outstanding challenges to Internet-wide topology mapping using active probes is balancing efficiency, e.g. induced load and time, with coverage. Toward maximizing probe utility, we introduce Ingress Point Spreading (IPS). IPS utilizes ingress diversity discovered in prior rounds of probing to rank-order available vantage points such that future probes traverse all known paths into a target network. We implement and deploy IPS to probe ∼49k random prefixes drawn from the global BGP table using a distributed collection of vantage points. As compared to existing mapping systems, we discover 12% more unique vertices and 12% more edges using ∼50% fewer probes, in half the time.

1 Introduction

Accurate and complete maps of the Internet topology are important to both security and networking research. As a piece of critical infrastructure, understanding network structure, interconnectivity and vulnerabilities is a first step toward protecting the Internet and making it more robust. Further, topology data is essential to network research that creates new protocols, performs modeling, designs clean-slate architectures, or examines Internet evolution and economics.

However, obtaining Internet topologies remains a challenging task [4]. The sheer size of the network implies that the accuracy of collected topologies can depend on the number, location, and probing rate of available vantage points (VPs) [16]. Topological inferences of paths, aliases, and structure can be brittle or lead to false conclusions [19]. Compounding the measurement difficulty, the Internet is non-stationary and dynamic. While mapping systems such as Archipelago (Ark) [10], Rocketfuel [17], and iPlane [13] have achieved Internet scale and produced important research insights [20][5], recent research, e.g. [2][18][7], shows that their performance, particularly in terms of probing efficiency as measured by the return of topological data per probing packet, can benefit significantly from an *adaptive* approach where the source and destination of each probe packet are judiciously chosen based on knowledge gained from prior probes and an understanding of network provisioning.

In this paper, we propose a new adaptive interface-level network mapping technique which we term "Ingress Point Spreading" (IPS). Underlying IPS is the observation that a target autonomous system (AS) is typically multi-homed and multi-connected. According to two 2010 studies [12, 6], the number of these

peering links, and thus the number of distinct ingress router interfaces for external traffic to enter the AS, are on the rise. We henceforth call these interfaces the *ingress points* of the target network. Intuitively, two probes would likely reveal more of a target network's topological structure if the probes were to enter the network via distinct ingress points. IPS aims to increase probing efficiency by first inferring the number of ingress points for a target network and then, for each new probe, selecting the VP with the highest likelihood to traverse an ingress point that has not yet been covered.

To evaluate the performance of IPS, we implement and deploy an Internet mapping system that integrates IPS with another recently proposed adaptive mapping primitive (subnet centric probing [2]). The system uses one day's worth of prior probing results to infer potential ingress points at different notional network boundaries for each target prefix. Rather than being agnostic to network structure, our system is designed to discover: i) the degree of subnetting within edge networks through an iterative interrogation process; and ii) sources of path diversity into networks by finding and exploiting the target's ingress points. This paper therefore makes the following three primary contributions:

1. Design and implementation of an Internet mapping system that integrates IPS with a complementary adaptive primitive originally proposed in [2].
2. Real-world deployment of the new mapping system. Specifically, we probed a sample set of 49,000 random destination prefixes in December, 2013.
3. Compared to data collected by a popular, currently deployed mapping system in the same time period for the same set of prefixes, our system finds more interfaces and edges, using only half of the total number of probes. This result is in contrast to prior efforts that demonstrate probing savings, but at the expense of lower topological recall.

2 Methodology

At the heart of our methodology is discovering network ingresses and predicting the ingress through which traffic from an available VP will enter a target network. Our intuition is straightforward: by ensuring that our probing uses all available ingresses, we more completely explore the target network, as well as exercise diverse paths to reach the target. As an additional benefit, a focus on ingress diversity matches an explicit higher-level goal of understanding topological connectivity, mapping disjoint paths, and characterizing ways in which portions of the network can become disconnected.

This section first describes probing properties that motivate a focus on ingress points, then details modifications to existing algorithms to support an ingress-centric approach. We then provide our algorithm to rank-order VPs on a per-destination network basis in order to maximize each probe's topological coverage.

2.1 Vantage Point Importance

It is well-known that the VPs used in active probing strongly influence the inferred topology [16]. A natural question is why we focus on the *order* of VPs

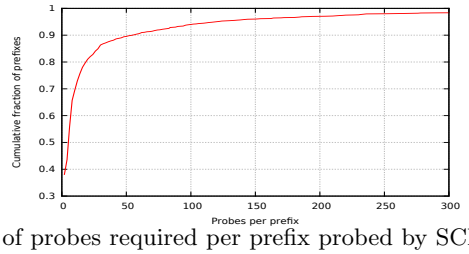


Fig. 1. Distribution of probes required per prefix probed by SCP. Because more than half of the prefixes are probed fewer than 10 times, VPs selection is important.

employed when probing a particular destination network. If all VPs are used, then the order in which they are used assumes only small importance. Instead, we consider two situations that commonly arise in topology probing: i) the set of VPs is large; or ii) the system must balance coverage and efficiency. For example, we may wish to use a subset of the VPs that result in the most topological coverage while exploring the destination network – thereby saving needless probing.

To characterize VP importance, we examine the popular CAIDA Ark system [10]. Ark divides the entire routed address space into logical /24 subnetworks, and in each "cycle," probes a random address within each /24 using a random VP. Ark assimilates the union of 21 of probing to obtain a high resolution map. For N cycles and M VPs, the expected number of unique VPs that explore a given /24 prefix (Y) in Ark is given by:

$$E[Y] = M - \frac{(M-1)^N}{M^{N-1}} \tag{1}$$

Examining one team of CAIDA probing from June, 2013, we see that $M = 18$ VPs were used. Thus, on average, each /24 in the union of $N = 21$ cycles is explored by: $E[Y] = 12.6$ VPs, and not all VPs are utilized even though $N > M$.

As a second example, the Subnet Centric Probing (SCP) algorithm of [2], which we also employ in our complete system, uses a variable number of probes per prefix in order to balance efficiency and coverage. To better understand the implications of SCP on VP selection, we used SCP with 60 Ark VPs to probe 1500 prefixes selected at random from the global Routeviews BGP view [14]. Figure 1 shows the number of probes per prefix versus the cumulative fraction of prefixes when using SCP. We observe that over half of the prefixes are probed fewer than 10 times, while $\approx 90\%$ of the prefixes see 50 or fewer probes.

This exploratory analysis of CAIDA's data and SCP support two observations. First, even when the number of probes is larger than the number of VPs, using randomly selected VPs is sub-optimal. Second, for systems such as SCP that attempt to maximize efficiency, the number of VPs used is frequently less than the total available. Thus, *the order in which VPs are employed matters.*

2.2 Recursive Subnet Inference

Intelligent selection of VPs, described in detail in the next subsection, is only a partial topology mapping solution. Just as important is the selection of destinations to probe. To this end, we take inspiration from the SCP algorithm

proposed in [2]. However, our practical experience in implementing SCP directly revealed two impediments. First, per-flow load-balancing, as commonly found in the Internet, perturbs SCP's stopping criterion by artificially influencing the path edit distance. Second, SCP's dependence on edit distance requires pair-wise comparisons between probes that originate at the same VP – and thus prevents the full utilization of multiple VPs.

Instead, we implement the Recursive Subnet Inference (RSI) technique which takes inspiration from SCP. The input to RSI is a network prefix, i.e. network and subnet mask. Rather than simply splitting the prefix into its constituent /24 subnetworks, as is done with e.g. Ark, RSI attempts to discover the internal subnetting structure of the given prefix. Abstractly, RSI performs a binary search over the address space represented by an input prefix, pruning those branches of the tree that do not reveal new topology information.

To interrogate a prefix, RSI uses the same Least Common Prefix (LCP) principle as defined in [2]. Given an input prefix and mask p/m, LCP splits the prefix into two halves and probes a center address of each from a different VP. More formally, $LCP(p/m) = (d_1, d_2)$ where the two destination addresses are:

$$d_1 = p + 2^{32-m-2} + 1 \tag{2}$$

$$d_2 = p + 3\left(2^{32-m-2}\right) + 1 \tag{3}$$

Note that LCP readily adapts to 128bit IPv6 prefixes in the future. We term the initial two probes to the two halves of the input prefix the "parent probes." For each input prefix, RSI maintains the set of discovered interfaces within the destination AS. By only considering those interfaces within the destination AS, RSI is agnostic to which VP issues the probes, thereby accommodating IPS.

Let I denote the set of all unique router interfaces discovered that belong to the AS of the target prefix. Let P_i denote the set of router interfaces within the target's AS discovered by the i'th probe. Then, RSI splits an input prefix into two halves and recursively operates on those two smaller prefixes (which leads to additional probing using different VPs) if the following condition holds:

$$|P_i \setminus I| \geq \tau \tag{4}$$

where we set $\tau = 1$ such that probing terminates for a prefix only if no new interfaces are discovered. The interface set is then updated: $I = I \bigcup P_i$.

2.3 Ingress Point Spreading

Given our analysis of the importance of VPs (§2.1), and a probing strategy that may use fewer probes than VPs (§2.2), we turn to implementing a primitive that extracts the most benefit from each probe via intelligent VP selection.

At a high-level, we assume M VPs that will explore X destinations within a prefix (p/m), where it is frequently the case that $X < M$. The problem is to select the VP for each of the X destinations to be probed. Practically, we view RSI as requiring a pool of VPs to serve as the origin of RSI's probes, where we rank-order the VPs to provide maximum per-probe topological coverage.

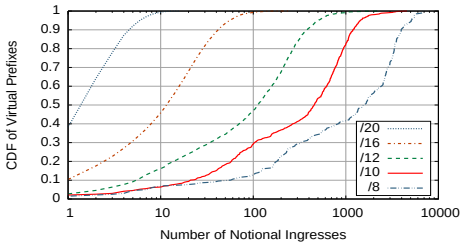


Fig. 2. Distribution of ingresses into prefixes of different logical size, as discovered during a prior round of probing. By expanding the size of the notional prefix, all VPs can be rank-ordered by their path diversity.

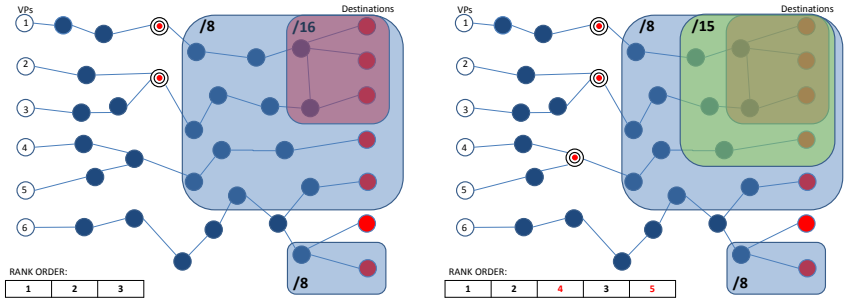
The Ingress Point Spreading (IPS) algorithm computes a per-destination network rank-ordered list of VPs based on *prior rounds* of probing. IPS seeks to utilize all of the ingress points discovered in prior rounds of probing such that future probing can induce probe traffic to flow through each of these known ingresses, thereby exploring more of the destination network's topology. By utilizing specific VPs, IPS spreads probes across ingresses. By spreading the probes across ingresses, RSI explores diverse paths, thereby preventing its early termination (as per the stopping criterion in eq. 4).

IPS employs an abstraction we term the "notional prefix ingress." A notional prefix is simply an expansion to a larger prefix aggregate containing the target prefix, while a notional prefix ingress is the first router interface hop that leads to a next hop whose IP is within the notional prefix.

IPS maintains a mapping of VPs to ingresses, i.e. which VP resulted in which ingress being traversed while probing the prefix. The notional prefix is important as there may be too few ingresses discovered from prior probing into the destination network. To obtain as much path diversity as possible, we perform an expansion to utilize ingresses into notional prefixes that represent a larger IP address aggregate containing the target prefix. Note that the notional prefix has no implied relationship to real-world BGP route aggregation; it is simply a means for IPS to expand its ingress search space for a given target network.

To provide intuition over the available notional ingresses as a function of the size of the notional prefix, we analyzed an entire cycle of probing data from CAIDA's Ark spanning June 2-4, 2013 in Figure 2. While using a /20 results in 99.4% of the notional /20 prefixes having 10 or fewer ingresses, expansion to a /16 provides more than 10 ingresses for more than half of the notional prefixes. Taken further, more than 60% of the notional /8's have 1,000 or more notional ingresses. Thus, by using our expansion technique combined with notional ingresses, IPS can adapt to best utilize any number or location of available VPs.

To illustrate, consider probing the prefix `205.155.0.0/16`. Figure 3(a) is a simplified example showing traces from prior rounds of probing from six VPs (numbered 1-6) to various destinations (the red colored nodes). The /16 prefix to be probed in the current round is shaded in red and encompasses three previous destinations and two ingresses into the /8 that lead to known paths into the /16 prefix. The VPs 1 and 2 are selected as the first two VPs in the rank order list



(a) Target /16 prefix with two ingresses (b) Expansion to find notional ingresses

Fig. 3. Ingress Point Spreading (IPS): Example where six VP are rank-ordered relative to the destination prefix on the basis of the notional ingresses the VPs traversed in prior probing.

depicted at the bottom of Figure 3, as these two VPs resulted in traversing the diverse ingresses in the prior round. Since VPs 2 and 3 share the same ingress router into the /8 prefix, the latter is included at the end of the list.

However, we wish to obtain a total order over all of the VPs (typically, many more than six). IPS then expands its ingress search space to include `205.154.0.0/15` as shown in Figure 3(b) (green shaded box). In this example, the expansion results in one additional destination and one more ingress. VP 4 then becomes the third in the rank-order as it traversed the diverse ingress into the notional prefix in the prior round. Following the same reasoning used for VP 3, VP 5 is included at the end of the list.

IPS continues to expand its search space, i.e. `205.152.0.0/14`, `205.152.0.0/13`, ..., `205.0.0.0/8`, where the larger aggregates are notional prefixes containing `205.155.0.0/16`, until all VPs are ordered. At each step, more notional ingress points may be identified which are used to rank order additional VPs to be used with RSI. RSI sends at least as many parent traces as there were notional ingresses to the original input prefix, but may send more.

3 Results

This section details our initial findings from deploying the combined RSI and IPS primitives described in §2. As a baseline, we implement the current Ark strategy[1] of subdividing the routed address space into /24's and select a random VP from which to probe a random address within the /24. Herein, we refer to the Ark method and resulting topology data synonymously as "Ark."

As part of the pre-probing process, we provide IPS with one day's worth of probing results as published by CAIDA [1]. We use CAIDA data as input to IPS to demonstrate that IPS can utilize not only prior rounds of our own probing, but also external sources of data (which, from a probing load perspective, are a sunk cost). To gain some initial understanding of IPS's sensitivity to training

[1] Direct comparison with published Ark data is not possible as we do not use "teams."

Table 1. Comparing RSI+IPS and Ark performance metrics. The same 49k random prefixes were probed in December, 2013.

Metric	**Ark**	**RSI+IPS** (Aug. 2013 trained)	**RSI+IPS** (Dec. 2013 trained)
Prefixes Probed	48,905	48,905	48,905
Vertices	464,544	521,513	520,903
Edges	906,680	1,024,295	1,034,101
Probes	4,041,289	2,056,562	2,052,842
Vertices (inside dest)	121,137	135,209	134,575
Vertices (intersection w/ ark)		309,997	309,971
Ingresses	31,138	38,532	39,020
Time	26h 55m	13h 38m	14h 47m

data and age, we perform two experiments, one where IPS is trained using data from Aug 28, 2013, and the second with Dec 18, 2013 training data. However, the probing itself was all performed between Dec 20-22, 2013.

From Routeviews [14], we randomly select 50,000 prefixes without regard to prefix size or origin AS. Of these original prefixes, we find 48,905 that were probed by both our IPS and Ark method. The natural changes in availability of VPs and routing require us to eliminate the 1,095 prefixes in order to fairly compare the two techniques. We probe these 48,905 prefixes using CAIDA's "topology-on-demand" service [9], where we have implemented RSI+IPS using 59 globally distributed VPs.

Table 1 summarizes our aggregate results. Two findings bear highlighting: not only is our combined RSI and IPS system significantly more efficient (using $\approx 50\%$ of the number of probes as compared to Ark and taking approximately half the time), we discover *more* topological information.

Examining the intersection of vertices, we observe that 309,997 vertices are common to both RSI+IPS (August) and Ark. RSI+IPS discovers 211,516 vertices not in Ark, while Ark discovers 154,547 vertices that RSI+IPS does not.

Figure 4(a) shows the distribution of the per-prefix difference of discovered vertex counts between our system versus Ark. Surprisingly, we find that our system performs worse than Ark for approximately 66% of the prefixes. Rather, RSI+IPS is significantly superior to Ark for a small number of prefixes, thereby contributing to the overall superior topological coverage. In contrast, Figure 4(b) shows that Ark and our system perform comparably in terms of edge discovery for approximately 80% of the prefixes, while we are superior for 10%. Again, the tail of the distribution is long – there are a small number of prefixes where we discover significantly more topological information. For future work, we will explore ways of refining the RSI stopping criterion as expressed in Eq. 4 to increase the percentage of networks for which our system has a better coverage. The fact that RSI+IPS performs better on some prefixes while Ark does better on others explains why a high number of interfaces and edges are uniquely discovered by each method.

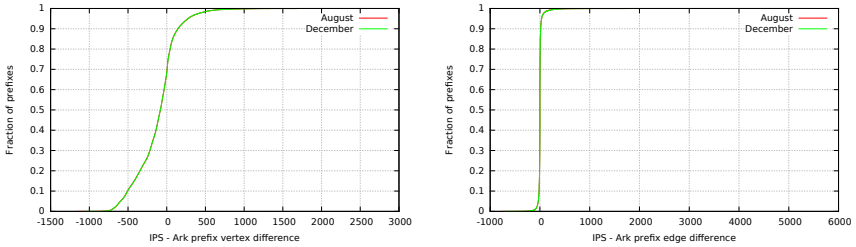
To understand the performance variations, we examine the distribution of the number of notional ingresses discovered for each prefix. Figure 5 shows two interesting phenomena. First, neither Ark nor our system discovers any ingresses for approximately 70% of the prefixes, as ICMP blocking and other forms of packet filtering may be prevalent, particularly for enterprise networks. However,

among those destination where probing within the target network is feasible, IPS finds significantly more ingresses than Ark. Based on the prefix's origin AS, we find that the top three prefixes for which IPS performs the best against Ark (as measured by additional vertices) are national ISP networks with hundreds of peering links while the bottom three prefixes belong to enterprise networks that have a small number of peering links. Furthermore, for those prefixes with at least one notional ingress, the relative performance of IPS has a medium correlation (with a Pearson correlation efficient of $\sim$0.35) with the number of ingresses discovered. The correlation is more significant ($\sim$0.45) if we consider only the set of $\sim$2000 prefixes with at least five discovered ingresses. Both of these findings confirm that IPS does a good job of leveraging available ingresses to increase probing efficiency. Of particular interest is that the size of the network prefix (in terms of IP address space) has a correlation of $\sim$0.52 to the performance difference, implying that our system performs better on smaller networks, while it often prematurely stops probing large prefixes. Adding a random component to RSI, or a lower probing threshold proportional to the network size may alleviate the performance differential of these prefixes. The results also suggest that the performance of IPS may be enhanced with a more effective network ingresses inference. We defer these to future work.

4 Related Work

Ever since the advent of network mapping research, emphasis has been placed on eliminating unnecessary probes to increase probing efficiency. Earlier notable efforts on this front include the development of the Rocketfuel [17], Doubletree [7] and DIMES [15] systems. These systems avoid traversing the same hops more than once by carefully choosing the starting point and the time to live (TTL) of each new traceroute packet used for probing. Follow-on studies [8, 2, 18] generalize and extend such techniques into a class of adaptive probing primitives, termed "set cover," characterized by a common requirement for determining a minimum number of probes to cover a set of previously discovered interfaces.

Recent efforts (e.g., [2, 16, 3, 11]) in developing adaptive mapping approaches center around leveraging information beyond the hops traversed before. The additional information considered includes knowledge of common subnetting practice, BGP route data, network latency characteristics, and potential path di-



(a) Vertex difference (b) Edge difference

Fig. 4. CDF of per-prefix coverage difference ($(RSI + IPS) - Ark$)

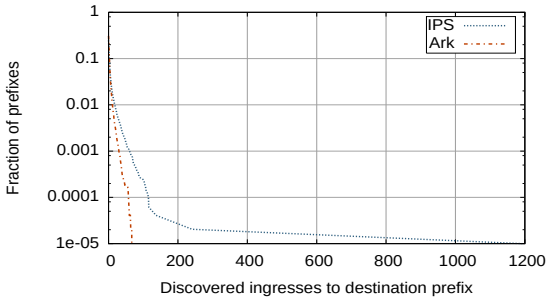


Fig. 5. CDF of per-prefix number of notional ingresses discovered

versity from different VPs. The work that is most related to this paper is the vantage point spreading (VPS) primitive proposed in [2]. IPS shares with VPS a similar intuition for increasing path diversity into target networks. The key difference is that IPS uses a more refined criterion of path diversity to drive the selection of VPs, and as such, is able to explicitly maximize the likelihood that a new probe will enter a destination network through a new ingress point. While IPS requires prior probing data in order to infer possible ingress points into each destination network, this data is naturally accumulated by production mapping systems as part of their functionality.

5 Conclusion

Significant prior work has considered the problem of balancing efficiency and coverage in active probing-based topology collection. We contribute to this body of work by explicitly taking into consideration target network ingresses, and the diversity of available vantage points (VPs) toward those ingresses, by developing the Ingress Point Spreading (IPS) algorithm. IPS rank-orders VPs for a given target prefix on the basis of ingresses discovered from prior rounds of probing. Thus, unlike prior approaches, IPS is not memoryless.

Via real-world probing of 49k randomly selected prefixes, we find that IPS not only reduces the probing load and time by approximately 50% as compared to CAIDA's Ark methodology, but also returns *more* vertices and edges. Crucial to many critical infrastructure questions, we also discover more ingresses.

While we have demonstrated promising results by utilizing ingresses to our advantage, significant future work remains. We wish to scale our probing by one more order of magnitude to encompass all advertised prefixes on the Internet, and run continually. Our practical experience has shown that VPs are unreliable, yet IPS cannot simply use the next VP in the ordered list when the preferred VP is down, as the complete ordering is perturbed. In addition, we have found prefixes with significant topology that goes undiscovered by RSI due to the particular deterministic selection of destinations causing premature termination. We must accommodate all of these issues in future work.

Our hope is that this work contributes to the continual progress being made on topology mapping systems. Moving forward, we additionally plan to integrate IPS with recent advances in topology set coverage and change detection.

Acknowledgments

We thank Young Hyun, kc claffy, Justin Rohrer, Arthur Berger, our shepherd Bruce Maggs, and the anonymous reviewers for invaluable feedback and support. This work supported in part by the Department of Homeland Security (DHS) Cyber Security Division under contract N66001-2250-58231. Views and conclusions are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government or DHS.

References

1. The CAIDA UCSD IPv4 Routed /24 Topology Dataset (2013), `http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml`
2. Beverly, R., Berger, A., Xie, G.G.: Primitives for active Internet topology mapping: Toward high-frequency characterization. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. pp. 165–171 (2010)
3. Chen, M., Xu, M., Xu, K.: A delay-guiding source selection method in network topology discovery. In: IEEE International Conference on Communications (2011)
4. k. claffy, Hyun, Y., Keys, K., Fomenkov, M.: Internet mapping: From art to science. In: IEEE Cybersecurity Applications and Technologies Conference (Mar 2009)
5. Dainotti, A., Squarcella, C., Aben, E., Claffy, K., Chiesa, M., Russo, M., Pescap, A.: Analysis of Country-wide Internet Outages Caused by Censorship. In: Internet Measurement Conference (IMC). pp. 1–18 (Nov 2011)
6. Dhamdhere, A., Dovrolis, C.: The Internet is flat: Modeling the transition from a transit hierarchy to a peering mesh. In: Proceedings of ACM CoNEXT (2010)
7. Donnet, B., Raoult, P., Friedman, T., Crovella, M.: Efficient algorithms for large-scale topology discovery 33(1), 327–338 (2005)
8. Gonen, M., Shavitt, Y.: An O(log_n)-approximation for the set cover problem with set ownership. Inf. Process Lett. 109(3) (2009)
9. Hyun, Y.: On-demand IPv4 and IPv6 topology measurements (2012)
10. Hyun, Y., k. claffy: Archipelago measurement infrastructure (2013), `http://www.caida.org/projects/ark/`
11. Kardes, H., Gunes, M., Oz, T.: Cheleby: A subnet-level Internet topology mapping system. In: COMSNETS. pp. 1–10. IEEE (2012)
12. Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F.: Internet inter-domain traffic. In: Proceedings of ACM SIGCOMM (2010)
13. Madhyastha, H.V., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., Venkataramani, A.: iPlane: An information plane for distributed services. In: Proceedings of NSDI. pp. 367–380 (2006)
14. Meyer, D.: University of Oregon RouteViews (2013), `http://www.routeviews.org`
15. Shavitt, Y., Shir, E.: DIMES: Let the Internet measure itself. SIGCOMM Computer Communication Review 35(5), 71–74 (2005)
16. Shavitt, Y., Weinsberg, U.: Quantifying the importance of vantage points distribution in Internet topology measurements. In: IEEE INFOCOM (Mar 2009)
17. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with Rocketfuel. ACM SIGCOMM Computer Communication Review 32(4), 133–145 (2002)
18. Thomas, B., Friedman, T.: Efficient IP-level network topology capture. In: Proceedings of Passive and Active Measurement (PAM) Conference (2013)
19. Willinger, W., Alderson, D., Doyle, J.C.: Mathematics and the Internet: A source of enormous confusion and great potential. Notices of the AMS 56(5) (2009)
20. Wu, J., Zhang, Y., Mao, Z.M., Shin, K.G.: Internet routing resilience to failures: analysis and implications. In: Proceedings of ACM CoNEXT (2007)