A Topological Analysis of Monitor Placement[*]

Alden W. Jackson, Walter Milliken, César A. Santiváñez, Matthew Condell and W. Timothy Strayer

BBN Technologies

10 Mouton Street

Cambridge, MA 02138

{awjacks, milliken, csantiva, mcondell, strayer}@bbn.com

Abstract

The Internet is an extremely complex system, and it is essential that we be able to make accurate measurements in order to understand its underlying behavior or to detect improper behavior (e.g., attacks). The reality, however, is that it is impractical to fully instrument anything but relatively small networks and impossible to even partially instrument many parts of the Internet. This paper analyzes a subset of the general monitor placement problem where the goal is to maximize the coverage of the entire universe of potential communication pairs (i.e., source and destination are randomly distributed in the routable Internet address space). This issue arises, for example, when trying to detect/track a distributed attack. We present results from a simulation, seeded with data from skitter *and* RouteViews, *that indicate we can monitor a packet with a high probability by monitoring relatively few points in the Internet. Our analysis suggests that the preferred strategy to place monitors should be to instrument one or two specific inter-AS links per AS for many ASes rather than deeply instrumenting a subset of the largest ASes.*

1. Introduction

Increasingly, passive monitoring systems are used to collect data to give insight into the underlying behavior of Internet, but fully instrumenting any but the smallest of networks is impractical and it is impossible to even partially instrument large parts of the Internet. Furthermore, not all participants in the Internet will want their parts of the network monitored, and if monitoring does occur, the data is rarely shared. Even for networks willing to cooperate, full instrumentation may be prohibitively expensive or the deployment may be incremental.

Faced with these realities, those that study network behavior must be able to answer the following questions in order to best use the limited resources available to instrument their networks: Where should monitors be placed? How many monitors are needed? Given a fixed cost, is one placement scenario better, worse, or equivalent to another scenario? Is it possible to compensate for networks either unwilling or unable to cooperate and at what cost?

This paper analyzes the problem of how to place monitors such that they see as much traffic as possible between sources and destinations randomly distributed in the routable Internet address space. Note that we are interested in maximizing the coverage among *all potential* communication pairs, as opposed to maximizing the fraction of Internet traffic observed. Knowing the most effective placement of monitors is essential for the detection of user's misbehavior and deliberate (distributed) attacks, where the emphasis is not on the "good" traffic but the anomalous.

This paper describes the design and development of an AS-level simulation that, when seeded with real routing data gathered from the Internet, provides insight into the best placement of monitors. Our results indicate the traffic between randomly distributed source and destination addresses can be monitored with a high probablility by instrumenting relatively few points in the Internet.

2. Related Work

There has been considerable interest in the placement of both active and passive monitors in networks. Jamin *et al.* [5] discuss algorithms for the effective placement of an active form of Internet instrumentation called Tracers, supporting the IDMap project, which provides a distance estimation service for Internet hosts. Their work focuses on evaluating heuristic algorithms for placing a fixed set of tracers in generated topologies. While their work mentions diminishing improvements as the number of Tracers is increased, it does not quantify the effect with respect to tracer placement.

Horton *et al.* [4] show that computing the optimal number of required active monitors, called beacons, in a network using a BGP-like routing policy is NP-hard and, in the worst case, approximately $n/3$ beacons are needed in a n node network. They then introduce a heuristic based on the topology of the public Internet that suggests a relatively small set of beacons is needed to cover the Internet.

Barford *et al.* [1] show that once a small number of active measurement sites are present, the marginal utility of additional active measurement sites declines rapidly. Furthermore, for active measurement, they show the utility of adding destinations is constant, indicating that it is more important to add more destinations than to add more sources.

The consensus of the above work is that a relatively small number of active monitoring sites are needed to achieve an accurate picture of the network topology, which is consistent with our conclusion that a small number of passive monitors is sufficient to achieve high monitor coverage.

In the literature, the passive monitoring problem has been described as only monitor placement or a combination of monitor placement and sampling control. For this study, we assume that once a monitor is deployed on a link, all flows carried by the link are fully monitored.

Chaudet *et al.* [3] study the problem of minimizing the number of—and finding optimal locations for—both passive and active monitors. They formulate heuristic solutions, based on a combinatorial view of the problem, and simulate the performance of their algorithms on synthetic topologies inferred by the Rocketfuel tool [9]. While these networks are much smaller in scale that the Internet, the authors do show that it can be very cost effective, in the number of monitors, to cover 95% of the traffic.

Suh *et al.* [10], consider the intersection of minimum cost monitor deployment and maximum traffic coverage problems. After proving that their problems are NP-hard, they propose greedy heuristics and simulate their performance on synthetic and Rocketfuel-inferred topologies.

Both Chaudet and Suh need to perform traffic matrix estimation, as in Medina *et al.* [6], to create a traffic load on the generated topologies. Since we focused on maximizing the fraction of *potential* source-destination pairs covered—as for example detect attackers employing hijacked user hosts—we avoid the need to model traffic matrices of "normal" or well-behaving traffic. Instead, we randomly select terminal nodes from the routable IP address space and identified if a monitor was in the path between any two terminals.

Recently, Cantieni *et al.* [2] offered a reformulation of the passive monitor problem, to consider, in a network where all links can be monitored, which monitors to activate and at what sampling rate to use to achieve a given measurement task. In particular, the authors show how their framework can be used to select monitors and sampling

rates to estimate the amount of traffic flowing between a set of origin-destination pairs across a real backbone network. While we do not address sampling in our monitors, we deal with the problem of distributed monitoring in internetworks, where the capability to monitor every link cannot be assumed.

3. Building the Simulation Model

In this paper we characterize the effect of incomplete monitor placement on the results of an idealized monitoring system. The system is considered to be "effective" if the network path between the sources and destinations of interest includes at least one link that is monitored.

To accomplish this, we developed a simulation environment to test the ability to monitor flows in large-scale partial deployment scenarios. The driving requirement for such a simulator is that it has to display reasonable microscopic realism even while using a macroscopic model of the Internet. Specifically, the topological model, routing behavior, and address distribution has to be sufficiently close to that of the Internet. A packet-level simulation is not necessary because our purpose is only to note if a monitor on a specific link saw a specific flow.

While it might be ideal to use a router-level model of the entire Internet, this granularity is also too fine and complex (over a million routers), and accurate routing and topology data at this level is typically confined to each ISP and held as proprietary.

Fortunately, a model of the Internet at the autonomous system (AS) level is sufficient: the topology spans the Internet and is readily available from the Internet registries. For randomly chosen source/destination pairs, the probability is extremely high that at least one AS boundary must be crossed by the path, so we can concentrate on these border points. This leaves the issue of routing. First we must be able to obtain an accurate view of the AS topology at a given point in time: CAIDA's *skitter* project [8] provides that. Second, we have to know what the state of the BGP routes through the AS topology at that point in time. The University of Oregon's RouteViews project [7] offers data on this. Picking a particular day, and acquiring both skitter and RouteViews data for that day, offers a reasonably good representation of the real routing within the Internet for that day.

The immediate benefit from using this AS-level model is that the simulator now has to deal with only about 15,000 ASes rather than a million or so routers, which is a much more manageable scale for running experiments. The downside, however, is that the results are achieved with a loss of resolution—instead of individual routers, we treat all the routers inside an AS as a single entity in our model, and assume any intra-AS traffic will not be detected.

Use of BGP routing data gave us a nearly full set of IP address prefixes used for routing, satisfying the realistic address distribution requirement for the simulation. But even with the BGP routing data, many routes were not known. For those ASes that did not have a RouteViews entry describing the route between them, we used a standard shortest-path-first (SPF) routing algorithm to complete their routes. While this means that our routing tables in the simulator are not totally accurate, we assert that they are representative of a large scale Internet and do meet the realism criteria for the model. The effects of using SPF at the AS level are discussed in Section 3.2.

3.1. Simulator Function

The simulator uses skitter data describing how ASes are connected to build a topological model of the Internet at the AS level. The skitter's AS link data identifies whether a skitter monitor, while performing a trace, found a direct or indirect link between ASes (or AS sets or multiple ASes). A direct AS link, AS_from–AS_to, occurs when the trace contains an IP address from AS_from's address space directly followed by an IP address from AS_to's address space. An indirect link occurs when an IP address from AS_from's address space is followed by one or more unknown IP addresses and then an IP address from AS_to's address space. The length of the unresolved gap is also recorded.

The simulator then populates this AS-level model with BGP routing information taken from RouteViews data. The RouteViews data contains a mapping of IP prefixes to AS paths. We used 5 different BGP dumps from the same day the skitter data used was collected, which contain close to 10 million routing entries. The simulator creates forwarding trees for all of the terminal nodes used for the study. From this, the simulator then calculates a unidirectional path for each pair of terminal nodes.

Due to the limited availability of AS-specific BGP routing data (there was a significant amount of overlap in the routing entries), only a fraction of the forwarding trees could be completely populated with routing information from the RouteViews datasets. Therefore, in order to complete the forwarding trees for all the destinations, the simulator employed a modified shortest-path-first algorithm (over the topology defined by the skitter data[1]) to populate routing information in the ASes that are not seen in the available BGP data. The algorithm respects the routes directly installed from the BGP routing data, and uses them as a basis to calculate the routes for the remaining ASes. We believe this approach produces routing trees that are a sufficient representation of the Internet routing trees, but before

we could fully assert this claim we had to test it. This test is discussed in Section 4.5.

The IP addresses of the terminal nodes were selected from routable IP addresss space. We randomly chose 200 terminal IP addresses, but 20 of the routing trees contain gaps (i.e., are unreachable from certain sources[2]) when used as destinations. As a result, 180 terminal IP addresses were used in the final simulation, resulting in $32,220 (= 180 \times 179)$ unidirectional paths.

Additional functions allow the simulator to determine the home AS associated with an IP prefix (either a source or destination), and to mark the path of a packet from a source AS through the forwarding tree. These additional capabilities of the simulator permit experiments to be devised exploring the characteristics of different monitor deployment strategies at the AS level. Once all the paths have been generated, the simulator contains a complete AS-level forwarding tree for the preselected destinations. These trees are saved and restored for use in multiple experimental trials, which amortizes the costly SPF computation across multiple experimental runs.

It should be noted that the terminal nodes chosen by this method are as representatively distributed across the globe as are the ASes from which they are drawn. There are terminal nodes on every continent except Antarctica. Also, AS selection for source and destination addresses is biased by size—that is, ASes with larger address space are more likely to be chosen as either a source or a destination AS. Finally, it is possible that the same AS is chosen as source and destination, that is, both terminal nodes chosen belong to the same AS, so intra-AS traffic can occur. In fact, in our experiments, 92 out of our 180 terminal nodes shared their home AS with at least some other terminal node. The largest set of terminal nodes sharing the same (large) home AS had a cardinality of 7. Obviously, when two nodes from that set are chosen as source and destination, their intra-AS traffic is not detected by our system. The total effect, as reflected in our simulation results, is a small reduction (less than 1%) in the detection probability.

3.2. Simulator Limitations

While we assert that the simulator is sufficient to provide an understanding of performance with repect to monitor placement, we know that it is not a truly accurate model of the Internet at the AS level. In particular, we removed or hand-repaired a fraction of the links in the skitter topology data because they indicated various types of multi-link or multi-path interconnections between ASes, or networks with multiple AS labels.

---

[1]Ties are broken in lexicographical order with respect to the ASes ID.

[2]Due to limitations on the skitter probing mechanism, the skitter-defined topology is not complete.

Because of this, and also due to the limitations of skitter and its probing techniques, not all routable ASes found in the BGP data were connected in the skitter topology graph. We eliminated these ASes from the model and re-homed their IP prefixes to neighboring ASes on the BGP routes. Although this eliminated nearly half of all the ASes in the data, nearly all of these are leaf ASes. The loss of the leaf ASes, which are usually connected to the Internet through a single intermediate AS, would have very little effect on monitoring systems primarily implemented in the major backbone ASes, which formed most of our experimental scenarios.[3] Furthermore, the only relevant effect of rehoming those IP prefixes is to cause some inter-AS communication (e.g., communication among two directly connected leaf nodes) to appear as intra-AS communication. Since intra-AS communications are, in our system, not observable, the total effect of removing the leaf ASes is to underestimate the monitoring capability of the system. Further, considering that the percentage of source-destination pairs that belong to two directly connected leaf ASes is much smaller than the total number of potential source-destination pairs, we believe that the removal/rehoming of leaf ASes had little effect on our results.

The simulator also has a naive notion of inter-AS links. It sees only a single link between any pair of ASes, despite the fact that there may actually be dozens of peering points between a given AS pair, in order to provide geographic offloading (i.e., hot-potato routing). In our results, this would increase the number of monitors required on all "links" by roughly an order of magnitude, but otherwise has little effect on the comparative results. On the other hand, the simulator treats public peering points such as NAPs as N^2 inter-AS links if there are N ASes present at the interchange.

The use of the SPF routing algorithm at the BGP/AS level of the Internet is also clearly a simplification. SPF routes all traffic to the same next hop node along the same path, i.e., it depends only on the link topology and the binding of IP prefixes to specific routers, and does not distinguish between prefixes other than to determine their home node (home AS in this case). However, BGP routing between ISPs is really policy-based, and may produce different routing results for different prefixes, even if they share the same home AS. To get an accurate picture of BGP routing would require either the ability to examine the BGP forwarding tables in every BGP router, or to see the routing updates output by every BGP router and know the BGP policies implemented in each one. RouteViews provides only a very limited subset of this data, just the BGP routes advertised by a modest number of border routers, and it provides no visibility into the policy at all. However, the actual BGP routing in the Internet still produces a simple forwarding

tree for each IP destination prefix. We only needed a representative forwarding tree for the placement evaluation, so this simplification does not affect the results.

4. Experimental Results

4.1. Experiment Setup

In this section we describe our monitor placement algorithm, which uses two input parameters besides the AS-level topology: the minimum out-degree for an AS to be instrumented, and the maximum number of inter-AS link instrumented at each AS.

To begin, the ASes are ranked in order of out-degree (their connectvity to other ASes), highest to lowest. Then, for every AS, its inter-AS links are ranked according the the out-degree of the neighboring AS, i.e., connections to larger ASes are ranked higher than those to smaller ASes. The result is an out-degree-ordered list of ASes, and each AS has an ordered list of its inter-AS links, from largest AS to smallest. The rationale behind this ordering is that ASes with higher node degree (e.g., core ASes in the Internet, concentration "hubs") are more likely to be part of the shortest path between any source-destination pair, and therefore instrumenting them provides more value (i.e., coverage) than instrumenting ASes with smaller out degree.

For a given value of minimum out-degree, all the ASes with the same or higher out-degree were selected. However, not all their inter-AS links were instrumented. Instead, for a given AS, its inter-AS links were instrumented one by one following the aforementioned ordered list, until all of its links were instrumented or the maximum number of instrumented links per AS (the algorithm's other parameter) is reached.

Once the set of inter-AS links to instrument is determined, the flow detection probability is computed as the percentage of the 32,220 paths that cross any of these instrumented links.

We conducted experiments where the minimum out-degree for an AS to be instrumented ranged from 1 (i.e., all ASes) to 1182 (i.e., only the first AS in the ordered list), and the maximum number of inter-AS link per AS ranged from 1 (i.e., only the link to the biggest neighboring AS) to 500 (i.e., all the links[4])

4.2. Breath-first vs. depth-first placement

Consider the relationship between two deployment configurations in terms of their performance and cost. Performance is measured as the percentage of the 32,220 flows

---

[3]The famous skitter map used a similar reduction of ASes in its construction.

[4]Except for the largest AS, which has 1182 inter-AS links, and therefore we limited its instrumentation to the first 500 links only.

Up to 500 top inter-AS links (depth-first)			**Only** the top inter-AS link (breadth-first)		
# of ASes Instrumented	Total # of links Instrumented	% of Flows Detected	# of ASes Instrumented	Total # of links Instrumented	% of Flows Detected
27	2329	91	521	521	91
12	1701	82	195	195	81
8	1411	68	91	91	70
5	1078	55	29	29	55
1	500	31	12	12	35

Table 1. Percent of flows detected for breath- vs. depth-first placement scenarios.

between the 180 target nodes that the scenario detects. Cost is the number of inter-AS links required to be instrumented to achieve the performance.

Columns 1, 2, and 3 of Table 1 show the percentage of flows detected if up to 500 inter-AS links are instrumented per AS for 1, 5, 8, 12, and 27 ASes (depth-first placement). We found that for 91% of the flows to be detected by at least one monitor, 27 of the top ASes need to be fully (up to 500 links) instrumented. This requires instrumenting 2329 inter-AS links.

Columns 4, 5, and 6 of Table 1 show the percentage of flows detected if *only* the top inter-AS link is instrumented for 12, 29, 91, 195, and 521 ASes (breath-first placement). We found that for 91% of the flows to be detected, 521 ASes need to have their top inter-AS link instrumented. (This requires 521 inter-AS links).

These results suggest that instrumenting only the top inter-AS link on N ASes is much more cost effective than multiple inter-AS links on a smaller set of ASes ranked by the number of its neighbors.

4.3. Cost effective deployments

We compare a set of cost-effective deployments using the two scenarios previously described. Figure 1 shows the deployment cost, in number of inter-AS links monitored, for detecting a given percentage of flows. It can be seen that for detection probabilities in the range from 22% to 35%, the breadth-first approach outperforms the depth-first approach by more than an order of magnitude.

Even when taking into account that an inter-AS link maps to multiple peering points (physical links) and that the inter-AS links chosen by the *breadth-first* algorithm tend to be the biggest, most complex ones, the difference in the number of inter-AS links instrumented by the two approaches is so large that it outweights any other criteria.[5]

---

[5]The number of physical links/peering points between two ASes—while unknown due to proprietary nature of the information—is arguably in the order of 10s rather than 100s, and this should scale the number of links instrumented for both cases roughly equally.
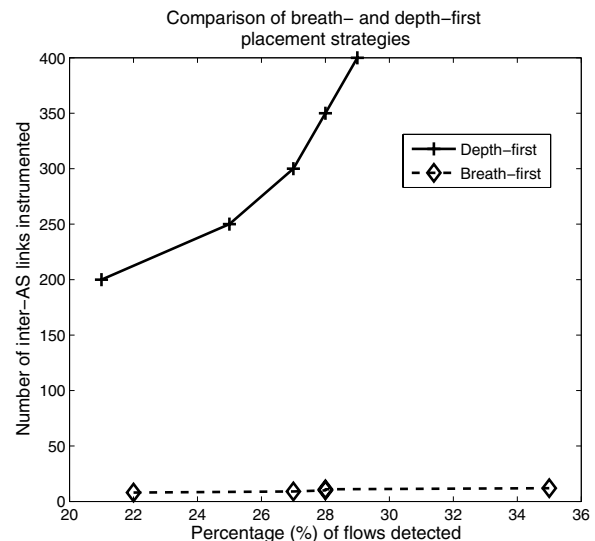

Figure 1. Number of inter-AS links that need to be instrumented for a given probability of detection, for both approaches. On the region that maximizes cost-effectiveness (probability of detection over instrumentation cost) breadth-first outperforms depth- first by an order of magnitude.

The interval between 22% and 35% is the "sweet spot" for the partial deployment system to operate. That is, it is the region where the ratio between the benefit (i.e., probability of detection) and the cost (i.e., number of inter-AS links instrumented) is maximized. Furthermore, while a 30% probability of detection may seem small, for many applications it is sufficient. For example, when the probability of detection of any source-destination flow is 0.3, and there is a malicious user sending k simultaneous attacks, then the missing probability (i.e., not being able to monitor/record *any* of the attacker's k attacks) is 0.7^k; k equal to 10 results
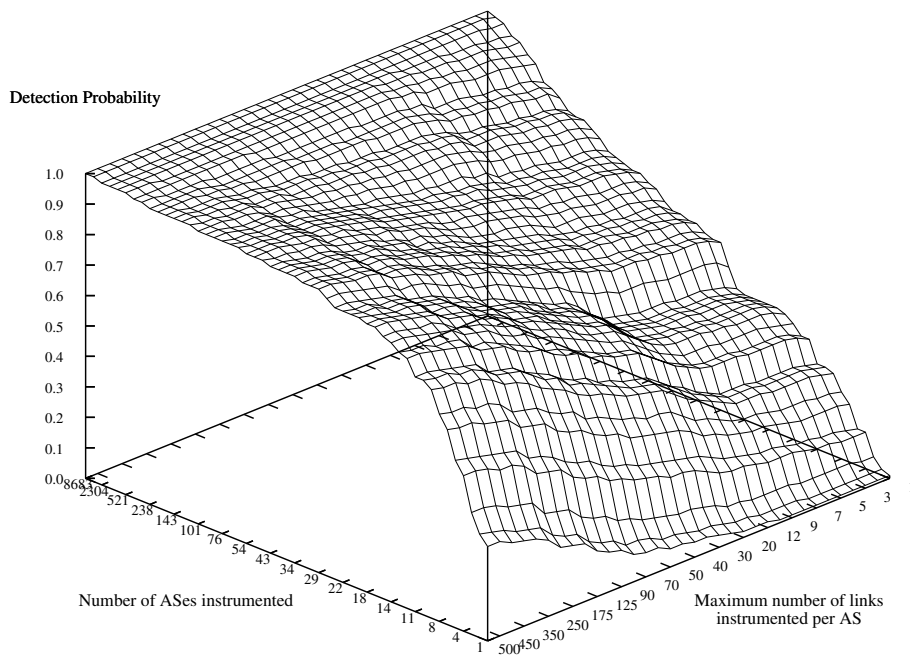
Figure 2. Probability of seeing a random connection.

in a miss probability of 0.03. This is not bad considering that only 10 inter-AS links had to be instrumented.

So, if the Internet behaves similarly to our model, even a relatively small breadth-first deployment can have significant effectiveness.

4.4. A larger view

Figure 2 shows the percentage of flows detected, or the probability of seeing a random connection, for all the combinations of both inter-AS links and ASes instrumented we explored. The x-axis is the number of instrumented inter-AS links for each instrumented AS. As before, the inter-AS links are always to the largest neighbor. The y-axis is the number of ASes with any instrumented inter-AS links. The y-axis sequence is the cardinality of the set of ASes whose number of inter-AS links is greater than or equal to some number. (The sequence, while monotonically increasing from 1 from a maximum of 8683, does not have values everywhere; one should not attempt to interpolate between the points.) An interesting feature of this graph is that the probability of seeing a flow between a random source-destination pair increases more quickly as the number of ASes instrumented grows than it does with increasing the number of instrumented links for a given AS.

Figure 3 shows the same data as Figure 2, except the de-

tection probability has been divided by the total number of links instrumented to give a cost-effectiveness indication of where the largest proportional gains in detection probability are for the least number of links instrumented per AS. This figure clearly shows that the preferred strategy should be to instrument more ASes with only one, or for increased robustness a few, links per AS. The ideal range for deployment is the area where the surface is maximized. This area of the plot (the "sweet spot" discussed in the previous section) covers a range of detection probabilities, enabling the network planner with a fixed set of resources to select both a cost effective and highly productive deployment of monitors. These results do not apply to traffic that has strong locality properties, e.g., traffic redirected to Internet content caches.

4.5. Vetting the analysis

When discussing the limitations of the simulator, we noted that our use of an SPF algorithm ignores the existence of BGP policy. We reiterate that the BGP dataset used does not contain policy information, only the routes advertised by a modest number of border routers. Thus, in the absence of the actual BGP policies, we must look elsewhere for ground truth.

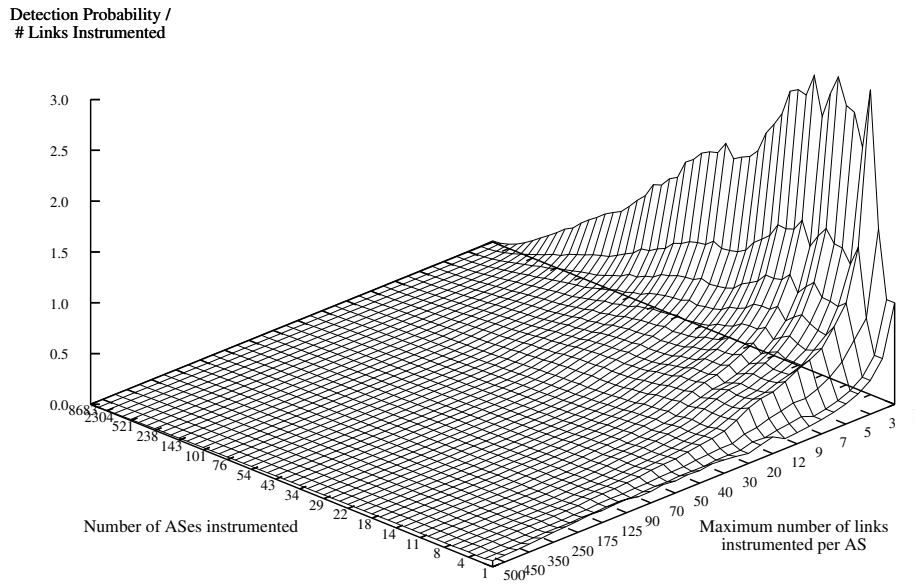The skitter logs contain the trace data for all IP addresses

Detection Probability /
Links Instrumented

Figure 3. Cost-effectiveness of instrumenting links.

probed by the skitter nodes. Since skitter does not use source routing, the network layer will always determine the path the probes take. These traces reasonably reflect the effect of any BGP policies implemented by the ISPs between skitter nodes and their destinations. Comparing the AS paths of the source-destination pairs from the skitter logs with those generated by the simulator should establish the accuracy of the SPF simplification in our network model.

In order to achieve fair comparison, we want to select skitter destination addresses that are close in prefix to the 180 random addresses used in the simulation study. Recall we saved the complete AS-level forwarding trees for the preselected destinations for purposes such as this. When the two addresses are close in prefix, then the likelihood of them belonging to the same AS is greater. The home ASes of the skitter sources are identified in the logs. We used the skitter logs from the same day as the simulation datasets for consistency.

The skitter project strives to achieve one monitored destination in each /24 network. There are over 16 million potential /24 segments in IPv4, about 4 million of them routable. At the time the data was collected, a skitter node could probe up to 800K addresses.

We performed a largest prefix match of an 80K sample of destination addresses from each of the 22 skitter nodes ($80K \times 22$) with each of the 180 IP addresses used by the

simulator. We expect that our hit rate should scale with the size of the sample, but we have not empiricly confirmed this assumption. Our hit rate for a /24 or greater match to one of the simulator's terminal node addresses in the was approximately 0.005%. Naturally, as the prefix becomes smaller, the hit rate of matches increases, 0.06% for a /20 or larger and 53% for a /10 or larger.

Focusing on the prefix matches greater than or equal to /24 (there were no matches greater than a /27), we have 50 individual skitter destination IP address that have /24 or better match to one of the 180 addresses used in the simulator. Comparing the AS paths extracted from the skitter data with that reported by the simulator we found the following:

- Six of the AS path pairs are identical

- Eight of the path pairs have the same length, but each path differs from the other in exactly one inter-AS hop, e.g., the AS path from skitter's trace data (19836, 10913, 3356, 8447, 6706) versus the simulator's AS path (19836, 7018, 3356, 8447, 6706)

- Four of the path pairs have the same length, but each differs from the other for exactly two consecutive inter-AS hops

- Nine out of the path pairs differ in length by 1 AS hop,

where the skitter AS path is longer and contains an extra AS hop for all but one pair of paths

- Fifteen of the path pairs differ in length by 1 AS hop, where the skitter AS path is longer and contains two AS hops that replace a single AS hop in the simulator's AS path. Neither the AS replaced nor the replacement ASes are the same for all the path pairs

- Two of the path pairs differ in length by 2 AS hops, where the skitter AS path is longer and contains two extra AS hops

- Two of the path pairs differ in length by 3 AS hops, where the skitter AS path is longer and contains a extra 3 AS hop sequence

- Four of the path pairs contain more than 3 differences in either length or AS path hops

Summarizing the above, three-fourths of the AS paths compared are virtually identical to each other. Of those that are not virtually identical, most contain significant overlap. Yes, there are small differences, as it is to be expected: we do not have a view into the BGP policies between the ASes. Our simulator's SPF would find the shortest path, which might not be the agreed path between ASes. However, our goal was to create a simulator that is representative, not an exact duplicate, of AS-level routing in the Internet. The data above vets our earlier assertion that our approach would produce routing trees that were a sufficient representation of the Internet routing trees.

5. Conclusions

This paper analyzed the problem of how to place monitors to see as much of the traffic as possible between randomly distributed routable sources and destinations. We described the development of a simulator that models the macroscopic topology, routing behavior and address distribution of the Internet. We vetted our supposition that the simulator is a sufficient AS-level model of the Internet using trace data from skitter. Our simulation results indicate that we can monitor a packet between a randomly-chosen source and destination addresses with a high probability by monitoring relatively few points in the Internet. This work also suggests that the preferred strategy to place monitoring systems should be to instrument only one (or two) logical inter-AS link(s) per AS for many ASes rather than deeply instrumenting the interconnection links of a subset of the largest ASes.

6. Acknowledgments

References

[1] P. Barford, A. Bestavros, J. Byers, and M. Crovella. On the marginal utility of network topology measurements. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 5–17, New York, NY, USA, 2001. ACM Press.

[2] G. R. Cantieni, G. Iannaccon, C. Barakat, C. Diot, and P. Thiran. Reformulating the monitor placement problem: optimal network-wide sampling. In *CISS 2006: Proceedings of the 40th Annual Conference of Information Sciences and Systems*, March 2006.

[3] C. Chaudet, E. Fleury, I. G. Lassous, H. Rivano, and M.-E. Voge. Optimal positioning of active and passive monitoring devices. In *CoNEXT'05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, pages 71–82, New York, NY, USA, 2005. ACM Press.

[4] J. D. Horton and A. Lopez-Ortiz. On the number of distributed measurement points for network tomography. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 204–209, New York, NY, USA, 2003. ACM Press.

[5] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. On the placement of internet instrumentation. In *INFOCOM (1)*, pages 295–304, 2000.

[6] A. Medina, N. Taft, S. Battacharya, C. Diot, and K. Salamatian. Traffic matrix estimation: Existing techniques compared and new directions. In *ACM SIGCOMM*, August 2002.

[7] RouteViews. U. of Oregon's RouteViews project. http://www.routeviews.org/.

[8] skitter. CAIDA's skitter project. http://www.caida.org/tools/measurement/skitter/.

[9] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, 2004.

[10] K. Suh, Y. Guo, J. Kurose, and D. Towsley. Locating network monitors: Complexity, heuristics, and coverage. *Computer Communications*, 29(10):1564–1577, June 2006.

IEEE
COMPUTER
SOCIETY