UCLA Computer Science Department - Techical Report TR-060017-2006

Placing BGP Monitors in the Internet *

Ricardo Oliveira †
rveloso@cs.ucla.edu

Mohit Lad †
mohit@cs.ucla.edu

Beichuan Zhang ‡
bzhang@cs.arizona.edu

Dan Pei §
peidan@research.att.com

Daniel Massey ¶
massey@cs.colostate.edu

Lixia Zhang †
lixia@cs.ucla.edu

Abstract

BGP Monitoring projects such as RouteViews and RIPE RIS provide valuable data for networking research. Prior efforts, such as understanding BGP dynamics, have mined the BGP data collected by RouteViews and RIPE to make general inferences about the Internet routing. Ideally one would like to have the collected BGP data covering the entire Internet. However various operational constrains limit the number of monitors to be used for the data collection, and up to this point, the selection of monitors has been based on intuitive judgment among available candidates. In this paper we use link coverage, defined as the set of links observed by all the monitors, as the monitor selection criterion and evaluate the current monitor selection by RouteViews project. We show that a simple greedy strategy can achieve much better link coverage than that achieved by the same number of monitors selected by RouteViews. Our results also show that edge nodes of the Internet provide more link coverage than the tier-1 nodes in general.

1 Introduction

The Border Gateway Protocol (BGP) is a path vector protocol used to propagate routing information globally in the Internet. RouteViews [10] and RIPE [14] collect BGP data from multiple operational routers (called monitors) throughout the Internet and provide public BGP data archives, which have been used in various BGP studies such as analyzing protocol dynamics, study of routing convergence, route flap dampening, and root cause identification, to name a few.

These public archives are a valuable source of data for understanding routing issues in the Internet, yet there has been no systematic study on the completeness of the BGP data collection. Ideally one would like to collect a complete set of BGP logs that cover the entire Internet, however blindly increasing the number of BGP monitors would not be an effective way to reach this goal. For example adjacent routers may report highly redundant data, which increases the size of the data archive with little additional value but can make the data mining effort more difficult. Furthermore, adding more monitors can also add substantial operational cost[8]. Thus one would want each new BGP monitor to add new information to the existing set.

One simple metric to measure the information added by each monitor is *AS link coverage*. The Internet consists of various administrative domains called autonomous systems (AS), and the BGP routing table at any router contains paths containing AS-AS links. The AS link coverage is defined as the number of unique AS-AS links seen in a BGP routing table. Whenever a link goes down, any routes going through that link must be changed and new BGP updates generated. Covering more links enable the monitors to collectively capture routing activity in a larger portion of the Internet. We first evaluate the current set of monitors by measuring the total number of AS links they covered, and then pose the question of "How can one place monitors to increase AS link coverage?". Note that our goal in monitor placement is not to *discover* new AS links, but rather to *cover* as many AS links as possible by placing a given number of monitors. We propose an heuristic for selecting monitors and use simulations on an Internet scale topology to compare its coverage to the current coverage achieved by the 60 monitors of RouteViews.

Our results show that our heuristic can cover the same number of AS links as the RouteViews monitor set by using only one third of the number of monitors; if we use the same number of monitors, we can cover almost 4,000 more AS links than the RouteViews set. We also find that our heuristic picks monitors from the edges of the Internet, compared to the existing data sets which draw monitors from close to

---

*This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No N66001-04-1-8926 and by National Science Foundation (NSF) under Contract No ANI-0221453. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA or NSF.

†Computer Science Department, University of California, Los Angeles.
‡Computer Science Department, University of Arizona.
§ATT Labs Research.
¶Computer Science Department, Colorado State University.

the core. By breaking down the AS links into different relationships such as customer-provider, sibling and peer, we observed that the peer links are the most difficult to capture. We believe these findings not only can help improve the selection of monitors for BGP data collection, but also can provide insightcvs on what the collected data could miss.

The rest of the paper is organized as follows. Section 2 reviews background information and introduces the coverage problem. Section 3 applies the greedy heuristic on a collected Internet topology and compares the results with RouteViews selection. Section 4 deals with some practical issues regarding the AS link coverage. Section 5 presents related work and Section 6 summarizes the paper and discusses future work.

2 Background and Link Coverage

In this section, we present some relevant background on Internet routing and BGP operations. We discuss the problem of placing monitors and the practical constraints of monitoring.

2.1 BGP Operations and Monitoring

The Internet consists of a large number of networks called Autonomous Systems (AS), and BGP (Border Gateway Protocol [13]) is the protocol used to exchange routing information between these ASes. Destinations are in the form of prefixes, where each prefix represents a network space. For example, a prefix 131.179.96.0/24 represents a network space of 2^8 addresses. As a path vector routing protocol, BGP lists the entire AS path to reach a destination in its routing updates. Monitoring projects like RouteViews and RIPE have a set of boxes called *collectors* that passively receive updates from participating routers, called *monitors*. Each collector acts as a regular router, except that it does not propagate any BGP update. The collectors log BGP updates received from the monitors and periodically dump a snapshot of the routing table of each monitor.

While one would want to collect data from as many sources of the Internet as possible, operational constrains place limitations on addition of new monitors. One of the main constrains are the man-power resources needed to manage the set of collectors of the monitoring infrastructure [8]. Besides day-to-day maintenance, unexpected events such as fiber cuts, software/hardware failures and loss of network connectivity demand a quick human response to assure there is no gap in the collected BGP data ([9] has a complete list of such events). These limitations motivate the search for a good strategy for monitor selection.

2.2 Monitor Placement & AS link coverage

BGP monitors can be placed in the Internet to achieve various objectives. One particular objective for placing monitors can be to capture BGP dynamics. For example, studies have been done on understanding routing instabilities [7], [11], convergence problems [6], and BGP behavior under stress events like worm attacks [16]. One would want to place monitors so as to capture more BGP dynamics. For example, if two monitors are selected such that one is single homed customer of the other, then the routing tables can be expected to overlap completely and the two monitors will most likely observe the same events. On the other hand, if two monitors are selected such that their routing table entries minimally overlap, then one can expect them to capture different events. This ability to capture different events would help in studies on understanding BGP dynamics, where inferences are made about the state of the Internet. We capture this aspect of widening the coverage using a metric called link coverage. The link coverage from a set of monitors is defined as the unique number of links present in the routing tables of all the monitors, and more the link coverage the better. With link coverage as the objective metric, our task is to pick a set of monitors so as to cover as many links, and thus, as much of the Internet as possible.

2.3 Topology and methodology

In order to evaluate strategies for placing BGP monitors we used an Internet scale topology collected from BGP tables and updates from RouteViews and RIPE, and several other sources such as looking glasses, route servers and routing registries [17]. Our topology has 22,478 nodes and 63,883 links, and represents an inferred snapshot of the Internet has of February 15^{th} 2006. We used the PTE algorithm proposed in [3] to infer the relationship between each pair of connected ASes. Then we computed the routing paths with no-valley policy [3] from each source to each destination using the three phase algorithm proposed in [4]. We observed that only 59,772 from the total number of links were used by some node's routing tree, therefore this is the maximum number of links we expect to cover if we add all nodes as monitors.

In the next section, we propose a greedy heuristic to place monitors, and evaluate this heuristic on the topology described above. We also use the same topology to compare the placement of the greedy heuristic to that of RouteViews. As of February 15^{th} 2006 RouteViews peers with 100 routers from a total of 60 autonomous systems.

3 A greedy heuristic

We now look at the coverage from the set of monitors from RouteViews and compare it to the coverage of our proposed greedy heuristic. We also understand the difference in coverage and provide insight into why the greedy selection provides more coverage and what types of links are difficult to capture.

3.1 $Greedy$ and $RouteViews$ Coverage

We first look at the link coverage provided by the set of 60 monitors from Routeviews. We order the monitors in the same chronological order they were selected by RouteViews and call this selection process $RouteViews$. With this ordering, we can see the value in terms of *new links* that each new monitor from RouteViews added.

Figure 1 shows the coverage of $RouteViews$. In order to project the coverage of $RouteViews$ beyond its 60 monitors, we approximate $RouteViews$ placement as a random selection of monitors, a process we call $Random$. In $Random$, a monitor is picked from a pool of 22,478 candidates representing the total number of ASes in our topology, with the condition that the same AS cannot be picked more than once.

We will now derive an analytical expression for the *expected* number of links covered by $Random$. We define the random variable $X_{i,n}$ as follows:

$$X_{i,n} = \begin{cases} 1 & : \quad \text{if link } i \text{ is covered after placing } n \text{ monitors} \\ 0 & : \quad \text{otherwise} \end{cases}$$

The expected value of $X_{i,n}$ then becomes

$$\begin{aligned} E[X_{i,n}] &= 0 \cdot Pr[X_{i,n} = 0] + 1 \cdot Pr[X_{i,n} = 1] \\ &= Pr[X_{i,n} = 1] \end{aligned} \tag{1}$$

We are interested in obtaining the expected value of the number of links covered with n monitors, $R(n)$:

$$\begin{aligned} R(n) &= E[\sum_{i=1}^{L} X_{i,n}] \\ &= \sum_{i=1}^{L} E[X_{i,n}] \\ &= \sum_{i=1}^{L} Pr[X_{i,n} = 1] \end{aligned} \tag{2}$$

where L is the total number of links. If p_i is the probability that we cover link i with a randomly picked monitor, then $Pr[X_{i,n} = 1] = 1 - (1 - p_i)^n$ and the number of covered links after n monitors $R(n)$ is given by:

$$R(n) = L - \sum_{i=1}^{L} (1 - p_i)^n \tag{3}$$

We will now show that the probability p_i is given by $\frac{S_i}{N}$ where N is total number of nodes in the topology and S_i is number of nodes that have link i in the routing tree. Furthermore, this probability does not change after the placement of each monitor, as we will see. Let $Z_{i,n} = 1$ if link i is picked at step n and $Z_{i,n} = 0$ otherwise. Y_n represents the node that is randomly picked at step n. Then we can write

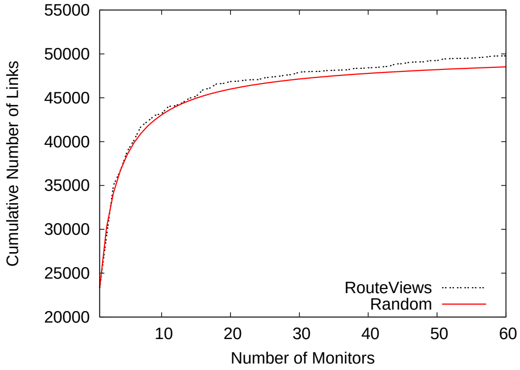

Figure 1: $RouteViews$ and $Random$ coverage on Internet topology.

$$\begin{aligned} p_i &= \sum_{j=1}^{N} Pr[Z_{i,n} = 1 | Y_n = j] \cdot Pr[Y_n = j] \\ &= \frac{1}{N} \sum_{j=1}^{N} Pr[Z_{i,n} = 1 | Y_n = j] \\ &= \frac{S_i}{N} \end{aligned} \tag{4}$$

Note that as the number of monitors n increases, $R(n)$ converges to the total number of links L. (3) is valid for $n < N$, since once we reach the last monitor N, we know for sure we are covering all the links, and therefore $R(N) = L$. Figure 1 shows that the expected coverage of $Random$ given by (3) is close to $RouteViews$ coverage. The gap between the curves is due to the fact that $RouteViews$ selection is not purely random ($RouteViews$ picked mainly high degree nodes belonging to the core tiers, as we will see in the following subsection). However, $Random$ still provides a reasonable approximation to $RouteViews$ coverage.

Algorithm 1 describes our proposed greedy heuristic for link coverage. Function $cover(u)$ returns the set of links covered by node u, i.e., the set of links present in the routing tree of node u. In each of the n steps, the greedy algorithm picks a node that contributes the highest number of new links. In case of tie between nodes for the highest number of new links added, the greedy heuristic picks the node with the lowest numeric ID.

Algorithm 1: Greedy($G(V, E), n$)

```
1. Initialize $C = \{\}$;
2. while $|C|$ != $n$ do
    Find $u \in V$ such that $|cover(u) \cup cover(C)|$ is maximum;
    $C = C \cup \{u\}$;
```

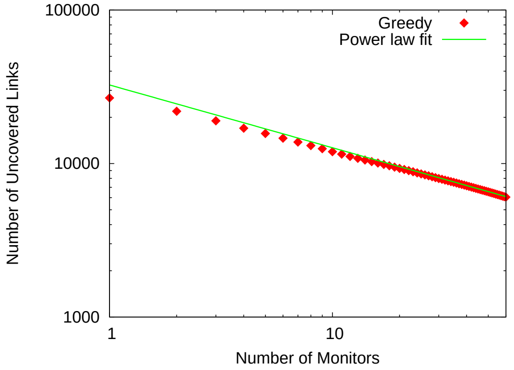

Figure 2: Greedy coverage as a power law process.

Figure 3 shows the accumulated link coverage of $Greedy$ after the addition of each monitor. We only show the coverage up to only 60 monitors, since we want to be able to compare $Greedy$ coverage with $RouteViews$. One can see that initially the link coverage increases rapidly, but beyond the first few monitors the contribution of each monitor leads to much a smaller increase. We analyzed this increase in more detail to find that $Greedy$ could be modeled as a power law process. More precisely, the number of links left uncovered by $Greedy$ at each stage, $u(n)$, can be approximated by the power law $u(n) \simeq u(1) \cdot n^{-\beta}$, where n is the number of monitors, $u(1)$ is the number of links left uncover by the first monitor and β is a constant. Figure 2 shows this relation.

The power law can be also obtain from the following difference equation:

$$G(n+1) - G(n) \simeq \beta \cdot \frac{L - G(n)}{n+1} \tag{5}$$

where $G(n)$ is the total number of links covered by $Greedy$ at step n, L is the total number of links and β is the constant of the power law. If we let n can have small increases Δn, ΔG are the links added by monitor n, (5) can be rewritten as

$$\frac{\Delta G}{\Delta n} \simeq \beta \cdot \frac{u}{n + \Delta n} \simeq \beta \frac{u}{n}$$

and since $\Delta G = -\Delta u$, we have

$$\frac{\Delta u}{u} \simeq -\beta \frac{\Delta n}{n}$$

and integrating both sides will yield the power law:

$$u(n) \simeq u(1) n^{-\beta} \tag{6}$$

From (5), $G(n)$ will converge to the total number of links L as the number of monitors increases. Furthermore since $G(0) = 0$, $G(1) = \beta L$ and therefore we can estimate β knowing $G(1)$. We found a good approximation using $\beta = 0.44$. The solid line in Figure 3 shows the link coverage given by (5), which is very close to the actual coverage of $Greedy$ represented by the dashed line.

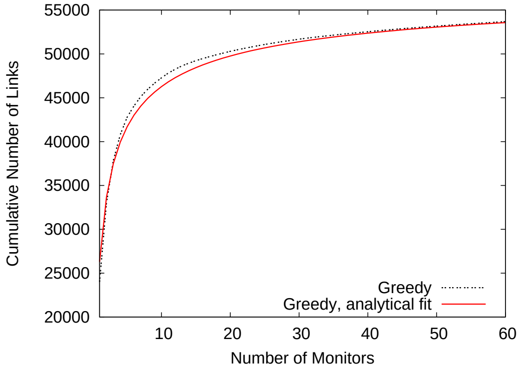

Figure 3: Greedy coverage on Internet topology.

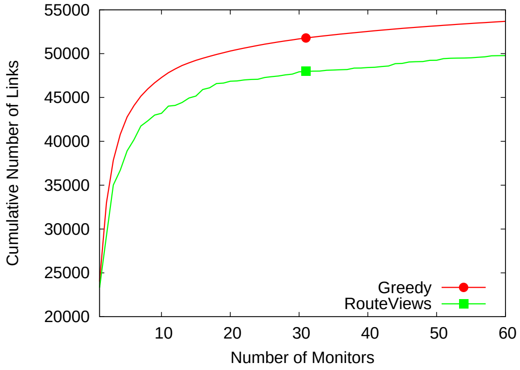

Figure 4: $Greedy$ and $RouteViews$ selection on Internet topology.

Figure 4 shows the coverage of $RouteViews$ and $Greedy$ with 60 monitors. The link coverage of $Greedy$ outperforms $RouteViews$, achieving a gap of almost 4,000 links after the last monitor is placed. $RouteViews$ covers 49,778 links with 60 monitors, while $Greedy$ achieves the same coverage with just 17 monitors.

We now extend the above comparison beyond the initial 60 monitors using the analytical approximations previously described. Figure 5 shows a comparison between $Greedy$ and $Random$ for 1,000 monitors using (5) and (3) respectively. We observed that both placements reach a knee point at about 100 monitors, after which there is a linear increment in the number of links covered. This figure also shows that the benefit of using a greedy strategy is not restricted to a small set of monitors.

3.2 Understanding the difference between $Greedy$ and $RouteViews$

Given Figure 4, our objective is to understand why $Greedy$ covers more links than $RouteViews$. In particular, we studied the difference in characteristics between the two sets of monitors picked by $Greedy$ and $RouteViews$ as well as the

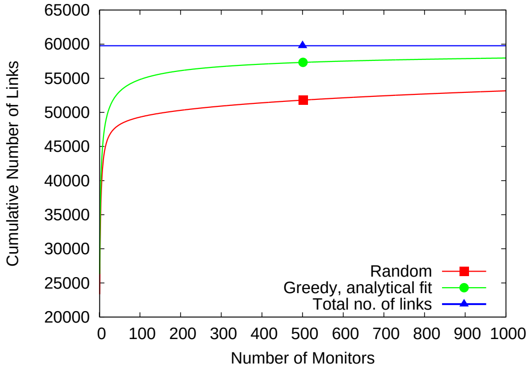

Figure 5: *Greedy* and *Random* coverage with 1,000 monitors.

	Total links	Covered by *Greedy*	Covered by *RouteViews*
Customer-provider	49,957	48,444	47,685
Peer	9,067	4,502	1,353
Sibling	748	739	740
Total	59,772	53,685	49,778

Table 1: Links covered by *Greedy* and *RouteViews* .

links covered by both the sets.

To analyze the monitor sets, we sort nodes into three different classes belonging to different levels in the Internet hierarchy. The *Tier1* class consist of ASes that do not have providers and form a full mesh within themselves. The *ISP* class consists of nodes that do not belong to *Tier1* and provide transit to other nodes (Internet Service Providers). The *Stub* class consists of ASes that do not provide transit to any AS, they just originate prefixes. Figure 6 shows the classification of the 60 monitors in *Greedy* and *RouteViews* into these three categories. We also included in Figure 6 the expected number of monitors picked by *Random* in each category, e.g. if there are S stubs in our topology of N nodes, *Random* will pick 60 · $\frac{S}{N}$ stubs after 60 placements. While *RouteViews* contains a lot of *Tier1* and *ISP* nodes, the majority of the nodes in *Greedy* are stubs, i.e., nodes at the border of the network. In fact, *Greedy* does not contain a single *Tier1* node. Furthermore, we can observe that the distribution of monitors of *Greedy* is very similar to *Random* . This indicates that picking a stub randomly does not assure a good contribution for link coverage, except when this selection is guided by an heuristic like *Greedy* .

To better understand the difference between *RouteViews* and *Greedy* , we plot in Figure 7 the distribution of node degree of the monitors picked by both placements. We observe that monitors selected by *Greedy* have degree less than 172, while *RouteViews* selects nodes with very high degree (2,656 and bellow).

In order to better understand the difference between the two placements, we classified the links covered by *Greedy* and *RouteViews* into customer-provider, peer or sibling, as presented in Table 1. We know from previous results that *Greedy* captures more links than *RouteViews* and looking at Table 1 we see that this difference comes mostly from peer links. To explain why *Greedy* might capture more links, Figure 8 shows a scenario where AS A is a customer of AS X, and AS B is a customer of AS Y. The link between AS A and AS B is a peer-peer link, while other links from AS A and AS B are customer-provider links. Due to routing policies, AS A will use link (A, B) to reach B's customers, since a peer-peer link is preferred over a provider path through AS X. However, AS A will not advertise paths involving link (A, B) to its providers. Similarly, AS B will not advertise any paths involving link (B, A) to its providers. Thus the only way to cover link (A, B) is to select at least one of A, B, one of A's customers, or one of B's customers as a monitor. Since most of the monitors picked by *RouteViews* are nodes at the core of the network, such peering links are easily missed.

In order to better illustrate the difficulty of capturing peer links, we show in Figure 9 the result of the greedy heuristic exclusively on peer links. Thus, at each stage, the node that can cover the most number of peer links is selected, and

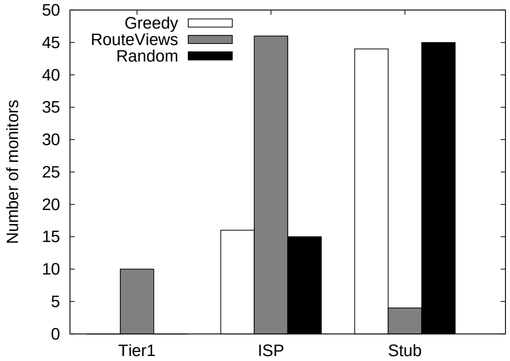

Figure 6: Location of monitor nodes.

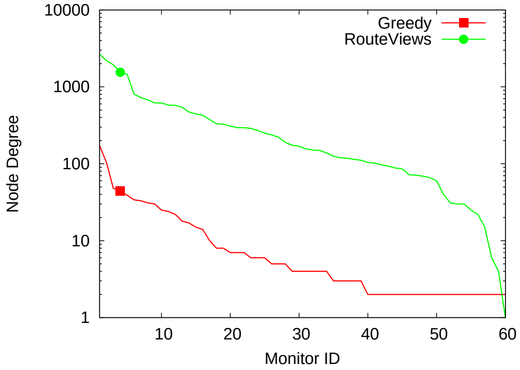

Figure 7: Degree distribution of monitors selected by *Greedy* and *RouteViews* .

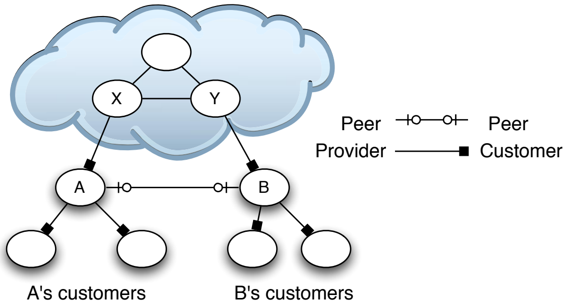

Figure 8: Covering peer links.

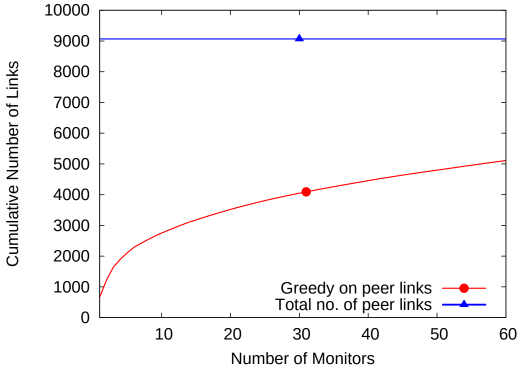

Figure 9: Greedy heuristic applied to peer links.

the total coverage of peer links is updated. After placing 60 monitors we covered 5,110 links, about 56% of the total number of peer links. From Table 1, we know that after placing 60 monitors $Greedy$ covers about 97% of customer-provider links, which indicates that the real challenge lies in covering the set of peer links.

Summarizing, in this section we show that a greedy heuristic can do significantly better than the existing $RouteViews$ set, and the set of nodes picked by greedy are mostly stubs. However, randomly picking nodes from stub sets results in a coverage similar to $RouteViews$. We also see that the difference between the two sets arises due to the difficulty to capture peer links.

4 Discussion

In previous sections, we showed how the greedy heuristic could be applied to the Internet topology to provide better coverage of AS links than the $RouteViews$. In reality, one does not know the complete AS topology in the Internet, and one AS may or may not be willing to serve as a monitor. In such situations, one cannot assume knowledge of entire entire topology and treat every single router as a candidate for monitoring. However, we can still use the insights from our results to do better than adding monitors by intuition. To start with, candidate monitors should be identified. In the case of RouteViews, this could mean asking for autonomous systems that are interested in monitoring. Each such candidate would then be needed to provide a routing table for one of their BGP routers. Routing tables for more than one time snapshot may be requested to ensure that snapshot is not taken during a major instability or route convergence process. If a sufficiently large pool of routers is received, then using their routing tables, we can apply the greedy heuristic to select the top n monitors based on the AS link coverage. The main difference from the previous section would be that we would consider a much smaller pool of potential candidates in this case compared to the general case where any AS could be picked as a monitor.

5 Related Work

We can find several efforts in the literature to passively discover AS-level topology based on BGP data at various monitors. [2] shows that the RouteViews snapshot misses a significant amount of connectivity information, and that additional AS connectivity information can be obtained from more vantage points (route servers and looking glasses). [17] explores an additional dimension of accumulating information from updates over time and obtained even more AS connectivity information. However, none of these approaches studies the monitor *placement* problem and the placement's performance. One closely related work is [12], which studies the new links covered by each additional monitors. This work showed evidence that more than 50% of the links in the AS topology are still to be unveiled, and that almost all these missing links are peer-peer links.

Many measurement and monitoring problems have been modeled as the facility location problem [15, 5]. These papers usually map their corresponding problems to some theoretical problems, provide NP-hardness proof, and use heuristics to approximate the optimal solutions. In contrast, our paper takes an empirical approach to study performance of the routing message monitors. Some of these efforts studies how to place the active measurement points for Internet tomography. [1] studies the active topology discovery problem, and showed that the marginal gain of adding additional probing sources declines rapidly after the second or third one.

6 Conclusion and Future Work

In this paper we showed that, without constraints on the candidates, the greedy approach can select one third of the number of monitors currently peering with RouteViews to achieve the same AS link coverage. In our topology, the set of links covered by $Greedy$ includes 97% of the total customer-provider links, but only 50% of the total peer links. Thus, it is clear that the challenge lies in capturing the peer links. We further examined the difference between the set of monitors picked by $Greedy$ and $RouteViews$ placements. We discovered that $Greedy$ picked most of its monitors from the edge of the network, while $RouteViews$ selected its monitors mainly from the core. However, randomly picking monitors from the edge is not as useful as $Greedy$. As we explained in Section 3, peer links are difficult to observe, and can only be captured by customers of the providers that establish the peering and the providers themselves.

A number of open questions still remain unanswered. Given one important usage of the collected data is for global routing research, how important is it to increase the coverage of peering links in order to understand global routing dynamics? What other criteria should one apply in selecting monitors? Finding answers to these questions is part of our planned future work in this area.

References

[1] P. Barford, A. Bestavros, J. Byers, and M. Crovella. On the marginal utility of network topology measurements. In *Proceedings of ACM IMW 2001*, October 2001.

[2] H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. Towards capturing representative as-level internet topologies. *Elsevier Computer Networks Journal*, 44(6):737–755, ”June” 2004.

[3] L. Gao. On Inferring autonomous system relationships in the internet. In *IEEE/ACM Transactions on Networking*, volume 9, pages 733–745, 2001.

[4] L. Gao and F. Wang. The extent of as path inflation by routing policies. In *Proceedings of IEEE Global Internet Symposium*, 2002.

[5] R. Kuma and J. Kaur. Efficient beacon placement for network tomography. In *Proceedings of ACM IMC 2004*, October 2004.

[6] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *Proceedings of the ACM SIGCOMM 2000*, August/September 2000.

[7] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. In *Proceedings of the ACM SIGCOMM ’97*, pages 115–26, Cannes, France, 1997.

[8] D. Meyer. Email exchange with authors on routeviews project. september 2005.

[9] U. of Oregon. List of RV Events. Available from: `http://www.routeviews.org/update.html` [cited 04/09/06].

[10] U. of Oregon. RouteViews Routing Table Archive. Available from: `http://www.routeviews.org/`.

[11] R. Oliveira, R. Izhak-Ratzin, B. Zhang, and L. Zhang. Measurement of Highly Active Prefixes in BGP. In *IEEE GLOBECOM*, November 2005.

[12] D. Raz and R. Cohen. The internet dark matter: on the missing links in the as connectivity map. In *Proceedings of the IEEE INFOCOM*, April 2006.

[13] Y. Rekhter and T. Li. A Border Gateway Protocol (BGP-4). *Request for Comment (RFC): 1771*, 1995.

[14] RIPE. Routing Information Service Project. Available from: `http://www.ripe.net/`.

[15] K. Suh, Y. Guo, J. Kurose, and D. Towsley. Locating network monitors: complexity, heuristics and coverage. In *Proceedings of the IEEE INFOCOM*, March 2005.

[16] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang. Observation and Analysis of BGP Behavior Under Stress. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop 2002*, 2002.

[17] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the internet as-level topology. In *ACM SIGCOMM Computer Communications Review (CCR)*, volume 35, pages 53–62, 2005.