# AS Path Inference by Exploiting Known AS Paths

Jian Qiu and Lixin Gao
Department of ECE, University of Massachusetts, Amherst, MA 01003
{jqiu, lgao}@ecs.umass.edu

***Abstract*— Inferring AS-level end-to-end paths can be a valuable tool for both network operators and researchers. A widely known technique for inferring end-to-end paths is to perform traceroute from sources to destinations. Unfortunately, traceroute requires the access to source machines and is resource consuming. In this paper, we propose two algorithms for AS-level end-to-end path inference. The key idea of our algorithm is to exploit the AS paths appeared in BGP routing tables and infer AS paths based on the ones. In addition, our algorithms infer AS paths on the granularity of destination prefix instead of destination AS. That is, we infer AS paths from any source AS to any destination prefix. This is essential since routing in the Internet is determined based on destination prefixes instead of destination ASs. The validation results show that our algorithm yields accuracy up to 95% for exact match and accuracy up to 97% for path length match. We further extend our algorithm to infer a set of potential AS paths between a source AS and a destination prefix. We find that on average, 86% of inferred AS path sets are accurate in the sense that one of the paths in the set matches the actual AS path. Note that our algorithms require BGP routing tables only and do not require additional data trace or access to either sources or destinations. In addition, we demonstrate that the accuracy of this BGP-based inference approach cannot go beyond 90%.**

## I. INTRODUCTION

The Internet has evolved into the largest man-made distributed system in the world. The selection of end-to-end Internet paths depends on both routing protocols deployed and a large number of network operators involved. Yet, the ability to infer end-to-end Internet paths is essential for network operators as well as network researchers to perform traffic engineering, network diagnosis, and overlay network routing.

A well-known existing tool for inferring end-to-end paths is to perform traceroute from a source host to a destination host. However, traceroute is limited since it requires access to source hosts and it is resource consuming given a large set of paths are to be inferred. Access to a large collection of hosts in the Internet is challenging due to the distributed administration of Internet hosts. Although there have been around 1000 traceroute/looking glass servers around the world [5], the provided sources are still far too few given the enormous scale and heterogeneity of the Internet.

Alternatively, BGP (Border Gateway Protocol) [13] routing information collected at several public data repositories, such as ROUTEVIEWS [4] and RIPE RIS [3], provides a relatively comprehensive view of AS level topology. Although it is infeasible to query the AS level paths from the BGP routers of the more than 20,000 ASs on the Internet, the public BGP information has been shown to be able to capture relatively complete Internet AS-level topologies [6] and infer AS commercial relationships [7]. Hopefully, the information can be exploited to infer AS level end-to-end paths while no need to access either the sources or the destinations and perform the resource-consuming probing.

Nevertheless, the BGP-based AS Path inference is not trivial. Internet path selection largely depends on routing policies, which in turn are defined independently by network operators in each individual AS and are seldom publicly available. Mao *et al* [9] are the first to conduct an intensive study on inferring AS paths between any two ASs based on BGP routing information. Their algorithm assumes the shortest AS path routing policy among available paths. They claimed accuracy up to 60% in the sense that one of the inferred shortest paths matches with the actual AS path, and accuracy up to 62% in terms of path length match. They further propose a novel scheme to infer the first AS hop by assuming the access to destination hosts. The evaluation results show that given the first AS hop information, the accuracy of path length match can be improved up to around 70%.

The AS Path inference can benefit a broad spectrum of network applications. For instance, in the overlay networks, knowing the AS Paths between any two nodes helps to find the nearest adjacent peering nodes in the AS-level [10]. Meanwhile, the information of AS Path length is also useful. An example is that Tao *et al* [14] shows that the AS Path length change dominates the end-to-end delay performance and can be utilized to estimate the end-to-end delay in the overlay network to choose the best relay nodes. Also, Gao *et al* [8] find that the AS path length information can help conduct ingress interdomain traffic engineering. In addition, knowing all possible AS paths from a source AS to a destination prefix is also helpful. In the overlay networks, the overlay links should be disjoint so as to achieve independent underlay link performance [10], [14]. The knowledge of all potential AS Paths between nodes helps to explore the disjoint overlay links.

In this paper, we formulate two problems for AS path inference. One is the single AS path inference problem, which infers a single best AS path from an AS to a prefix. The other one is the potential AS path inference problem, which infers a set of AS paths from an AS to a prefix. Note that in contrast to [9], we infer AS-level end-to-end paths on the granularity of destination prefix instead of destination AS.

The key idea of our inference algorithms is to exploit the known AS paths appeared in BGP routing tables. Using only BGP routing tables, our algorithm can yield an accuracy of up to 95% and 60% on average for exact match, i.e. the inferred best path exactly matches the actual path. In terms of path length match, our algorithm is able to yield an accuracy of

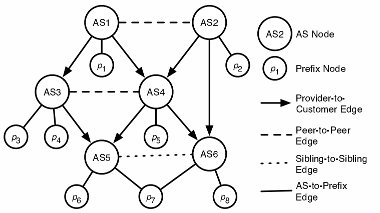


Fig. 1. Example of AS-Prefix Graph

up to 97% and 81% on average. Moreover, given the first hop AS information, the average accuracy can be improved to 78% for the exact match and 88% for path length match respectively. For the potential AS path inference problem, we find that by incorporating the path set stored at an AS node for a destination prefix, the inferred AS path set contains the actual AS path in 78% of our validation cases. Furthermore, we can extend our algorithm to achieve average accuracy of 86% in the sense that one of the paths in the inferred path set is the actual path although we limit the number of paths in the path set to no more than 10 only. Meanwhile, our experiments demonstrate that it is due to the incompleteness of AS graph derived from BGP routing tables that the inference accuracy cannot be higher than 90%. In this sense, our algorithm is able to capture 95% of the inferable AS paths.

We further perform intensive experiments to validate the robustness of our algorithms and examine the impact of the selection of the BGP vantage points, which provide the BGP routing tables for inference, on the accuracy. Besides that, despite the novel features that our algorithms have incorporated, they are still simple, efficient, and able to provide near real-time inference results.

The rest of the paper is organized as follows. The models and the problem formulation are introduced in section 2. Section 3 describes the details of the inference algorithms. In section 4, we evaluate the inference accuracy of the algorithms. Finally, section 5 summarizes the paper.

## II. MODEL AND PROBLEM FORMULATION

We model the Internet AS topology as an AS graph $G = (V, E)$ where $V$ is the set of ASs and E is the set of AS peering relationships. A path $P$ in the AS graph is a loop-free sequence of AS nodes, i.e. $P = (v_k v_{k-1} \dots v_1)$ where $v_i \neq v_j$ if $i \neq j$. $|P|$ represents the length of path $P$. We use $\phi$ to denote an empty path whose length is $0$. In the interdomain routing system, two neighboring ASs exchange paths according to their commercial agreements. Typically, the AS relationships can be classified into three categories: provider-to-customer, peer-to-peer and sibling-to-sibling. Each edge in the AS graph is labeled with the types of the commercial relationships. The AS path import and export policies complying with AS relationships lead to the *valley-free property* of the AS paths in an AS graph; that is, in a valid AS path, a provider-to-customer or peer-to-peer edge should be followed by a provider-to-customer edge only. In an AS graph with AS relationships (for simplicity, still called as AS graph hereinafter), a valid path must be loop-free and valley-free.

In an AS graph, two valid paths can be concatenated into one valid path, i.e. $(v_i \dots v_1) + (u_j \dots u_1) = (v_i \dots v_1 u_j \dots u_1)$ if the new path is both loop-free and valley-free; otherwise, the result is an empty path $\phi$. We can compute the shortest valley-free paths from an AS to another AS in an AS graph, as has been done in [9]. However, BGP routing tables provide the AS path information in the granularity of a source AS and a destination prefix. In order to infer the AS paths specific to destination prefixes, the topological information of the prefixes should be encoded into the AS graph. As shown in Figure 1, we also model the destination prefixes as nodes. The AS-prefix graph $G = (V, V_P, E, E_p)$ is composed of not only AS set $V$ and the AS edge set $E$ but also the prefix set $V_p$ and the AS-prefix edge set $E_p$, which comprises of links connecting ASs to their originated prefixes. Note that a prefix is attached to the ASs that originate it. Thus a prefix might connect to multiple ASs, as is known as MOAS (Multiple Origin AS) [15]. A path in an AS-prefix graph can append a prefix at the end of the path, but a prefix cannot appear at the other positions of the path. The length of a path only counts for the number of ASs that the path includes.

For a prefix $p$, an AS $u$ maintains a path set $rib\_in(u)[p]$ that contains all the feasible paths from $u$ to $p$ learned from its neighbors. Among these paths, $u$ will choose the best one, denoted by $path(u)[p]$, as its path to $p$ with a best path comparison procedure $bestpath_{u,p}()$ [1]. BGP is specified as a policy-based routing protocol. The route selection at each individual AS is determined by the local routing policies, which is usually seen as confidential and seldom published. However, the routing policies of each AS can be arbitrarily configured and largely independent of the configurations of the other ASs. Therefore it is challenging to infer the AS paths only relying on the public information.

The objective of the paper is to infer the exact AS path that an AS uses to reach a destination prefix based on the public BGP routing information, which is referred to as the single AS path inference problem. Besides this, we also propose an alternative of the single AS path inference problem, i.e. the potential AS path inference problem: to infer a set of AS paths from a source AS to a destination prefixes and some of these paths are potentially the actual paths that the source AS uses to reach the destination prefix.

## III. INFERENCE ALGORITHMS

We proposed two algorithms to solve the single AS path inference problem and the potential AS path inference problem. We exploit the BGP routing tables from several vantage points to infer the AS paths. Because we utilize BGP routing tables as the basis of our inference algorithm, we are able to provide near real-time inference results based on the most recent BGP table dumps. We have setup a website [11] to provide an online AS path inference service. Users are able to access the most up-to-date inference results through the provided interfaces. We hope that it provide a general-purpose AS path inference service to the research communities.

```
initActiveQueue($p, queue, G, baseASset$)
for $v \in baseASset$
do append $v$ to $queue$
    $path(v)[p] \leftarrow$ sure path of $v$
    sort $rib\_in(v)[p]$

knownPath($p, G = (V, E), baseASset$)
$queue \leftarrow \emptyset$
initActiveQueue($p, queue, G, baseASset$)
while $queue \neq \emptyset$
do $u \leftarrow$ pop($queue, 0$)
    for $v \in u.peers$
    do if $v \notin baseASset$ and $(v) + P_u \neq \phi$
        then $tmpp \leftarrow rib\_in(v)[p][0]$
            update $rib\_in(v)[p]$ with new path $(v) + P_u$
            sort $rib\_in(v)[p]$
            if $tmpp \neq rib\_in(v)[p][0]$ and $v \notin queue$
                then append $v$ to $queue$

return $\{rib\_in(u) | \forall u \in V\}$
```

Fig. 2. Pseudocode of KNOWNPATH algorithm

```
multiPaths($p, G = (V, E), baseASset, M$)
$queue \leftarrow \emptyset$
initActiveQueue($p, queue, G, baseASset$)
while $queue \neq \emptyset$
do $u \leftarrow$ pop($queue, 0$)
    for $v \in u.peers$
    do if $v \notin baseASset$
        then $tmpps \leftarrow path(v)[p][0 \cdots M]$
            update $path(v)[p]$ with $\{(v) + path(u)[p][0 \cdots M]\}$
            sort $path(v)[p]$
            if $tmpps \neq path(v)[p][0 \cdots M]$
                then append $v$ to $queue$
return $\{rib\_in(u) | \forall u \in V\}$
```

Fig. 3. Pseudocode of MULTIPATHS algorithm

Similar to [9], we applies SPF-like (Shortest Path First) algorithms to compute the AS paths in an AS graph. Two major features distinguish our algorithms from [9].

### A. Incorporating Known Paths

The AS graph is extracted from the BGP tables of $N$ given vantage point ASs. Thus for each destination prefix $p$, we have already known at most $N$ known paths to this destination before we perform the inference algorithm. Because [9] does not utilize these known paths, it does not guarantee the inference accuracy even for the vantage point ASs themselves. The intuition of our algorithm is to incorporating these known paths into the inference process.

Suppose that path $loc\_rib(v_k)[p] = P_k = (v_k v_{k-1} \dots v_1 p)$ is the known path of a vantage point AS $v_k$ to a destination prefix $p$. According to the BGP specification, the best paths of $v_k$ is actually derived from one of the best paths of $v_{k1}$ $P_{k-1}$ with path concatenation. Thus, one of the best paths of $v_{k1}$ to $p$ at this time is the sub-path $P_{k1} = (v_{k-1} \dots v_1 p)$. Similarly, we can get the best paths for $v_{k-2}, \dots, v_1$. Thus, we can derive $(k1)$ additional paths from a known path of length $k$. We call both the known path $P_k$ and the derived path $P_{k1}, \dots, P_1$ as *sure paths*. The ASs that have the sure paths are called as the *base ASs*.

The inference algorithm, which is called KNOWNPATH, starts from the initial state in which all the base ASs install their sure paths. Then the algorithm computes the shortest paths for all the other ASs by extending the sure paths. The length of the extended part is called *unsure length*. The pseudocode of the algorithm is shown in Figure 2. The algorithm maintains a global variable $queue$ which stores the ASs whose best paths have been changed. In the initial state, all the base ASs install their sure paths and thus are put into $queue$. Later, in each iteration, an AS in the at the head of the queue is pop out and to propagate its best path to its neighbors. If the best path of one of its neighbors changes, this neighbor will be put into $queue$. Note that the base ASs will never be added into queue after the initial round. At each node $u$, we apply SPF-like path comparison procedures to compare the paths in $rib\_in(u)[p]$, which are sorted in descending order of preference. Accordingly, the first path in $rib\_in(u)[p]$ is the best one, and the first $M$ paths in $rib\_in(u)[p]$ are the best $M$ paths given the path comparison procedures. We utilized several heuristics in the path comparison procedures to improve the inference accuracy. The details of the heuristics can be found in our technical report [12].

### B. Incorporating Multiple Paths

A disadvantage of the single path inference algorithm is that the inference inaccuracy of the predecessor ASs will be inherited by the successor ASs. Thus the inference accuracy of the best path will exponentially decrease with the growth of the unsure length. However, if we look at the first $M$ paths in $rib\_in$ of an AS instead of the first one, the probability that the actual path shows in them is significantly higher than the probability that the first one exactly matches the actual path. If the first $M$ paths are allowed to be propagated to the neighbors, the successor ASs will have higher chance to obtain the correct path. Consequently, we are more likely to find the correct path in the successors $rib_in(u)[p]$s and the inference accuracy for the potential AS path inference problem will be improved. Therefore, we further extend the KNOWNPATH algorithm to support the advertisement of multiple paths between ASs, which is called MULTIPATHS. The pseudocode is as shown in Figure 3.

## IV. EXPERIMENTS AND VALIDATION

### A. Inference Accuracy Evaluation

We use the routing tables collected from ROUTEVIEWS [4], RIPE RIS project [3] and CERNET BGP VIEW [2] at 16:00 PM GMT, Oct. 10, 2004, to examine the inference accuracy of the algorithms. The data set contains 67 complete BGP routing tables. We use 41 from ROUTEVIEWS as the vantage point $ASs$ to construct the inference database. Based on the BGP tables of these ASs, we construct the AS-prefix topology, infer the AS relationships, and then perform the algorithms. The AS relationship inference algorithm in [7] is adopted in this paper. We use the routing tables of the remaining 21 ASs for evaluation. The details about the selection of these $ASs$ are discussed in our technical report [12]. In the MULTIPATHS algorithm, the number of propagated multiple paths is set to 3. For the sake of comparison, we also check the performance of the basic inference algorithm that incorporates neither known paths nor multiple paths, called NOPATH, which is similar to the algorithm used in [9].

**Upper bounds on inference accuracy**: Before validation, we first investigate how the inaccuracy in the topology information of the resulting AS-prefix graph impact the performance of the AS path inference algorithm. We found that 10% of the paths in the evaluation tables cannot be inferred due to missing links and 0.5% due to valley paths. The existence of these uninferable paths imposes an upper bound of around $89.5\%$ to the AS path inference accuracy.

Several metrics are examined for the purpose of evaluation. For an examined AS $u$ and a particular prefix $p$, by comparing the path lengths of the inferred path and the actual path it is using, the result can be *longer*, *shorter* or *length match*; by checking whether its actual path is the inferred best path or just shows in the inferred $rib\_in$, the result can be *exact match* (the inferred best path is the actual path) or *one match* (one of the paths in the rib in matches the actual path). Besides these metrics, we also define a metric *match in* $K$, which indicates whether the actual path shows in the first $K$ paths in the $rib\_in$. The metrics *longer*, *shorter*, *length match* and *exact match* are used for the evaluation of the single AS path inference problem and the metrics *one match* and *match in K* are for the potential AS path inference problem. Note that due to the inaccuracy of the AS-prefix graphs, including the incompleteness of the topology and the inaccuracy of the AS relationships, there exists some AS paths that cannot be inferred no matter how the inference algorithms are designed. The limitation imposes an upper bound on the inference accuracy. In order to show the performance of the algorithms with regard to the limitations, we also indicate the inference upper bounds for each AS with a metric called *upper bound* in the experiments.

Besides, [9] also developed a novel technique to detect the first AS hop that a source AS will use to reach a destination. They showed that if the first AS hop information had been provided, their inference accuracy of length match could be significantly improved by up to $15\%$. In this paper, we also check how the performance of the algorithms can be improved by incorporating the first AS hop information assuming that the information had been available beforehand.

Curves in Figure 4 show the performance of KnownPath algorithm for the $21$ evaluation ASs. The $x$ axis represents the indexes of the evaluation ASs and the $y$ axis shows the value of each metric. Figure 4(a) shows the performance of the algorithm without incorporating the first AS hop information. We find the performance various for different ASs. In terms of exact match, the best case is up to 95% while the worst case is only $16\%$. The average accuracy over all the evaluation ASs is around 60%. Figure 4(b) shows the accuracy of the algorithm with the first AS hop information. It shows that by incorporating the first AS hop information into AS path inference, the performance of the KnownPath algorithm is improved significantly. The best case is up to 97% while the worse case is 54%. The average accuracy is improved to 78%.

For the purpose of comparing the performance of the various algorithms, the averages of each metric in different scenarios are listed in Table I. We can find the influence of different factors on the inference accuracy. By leveraging known paths, there is a 27% improvement in terms of exact match from 33% to 60% and a 9% improvement in terms of length match from 69% to 78%. Furthermore, if the first AS hop information is given, there is another 18% gain in exact match to 78% and another 10% improvements in length match to 88%. In addition, since the MultiPaths algorithm adopts the same path comparison procedure as KnownPath, there is no distinct differences for the performance metrics pertaining to the single AS path inference problem. On the other hand, for the performance related to the potential AS path inference problem, the MultiPaths algorithms yields an average accuracy of as high as 86% for one match, which is about 95% of the inference upper bound (90%).

TABLE I

Average Inference Accuracy of the algorithms (%)

| | longer | shorter | length match | exact match | one match |
|---|---|---|---|---|---|
| NoPath | 5 | 25 | 69 | 33 | 49 |
| NoPath w/ first hop | 7 | 16 | 78 | 51 | 52 |
| KnownPath | 6 | 15 | 78 | 60 | 78 |
| KnownPath w/ first hop | 8 | 4 | 88 | 77 | 78 |
| MultiPaths | 8 | 10 | 81 | 59 | 86 |
| MultiPaths w/ first hop | 10 | 3 | 87 | 77 | 86 |
| NoPath for Vantage Points ASs | 0 | 24 | 76 | 40 | 73 |

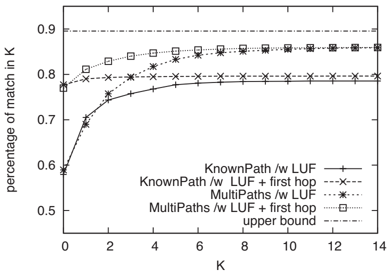
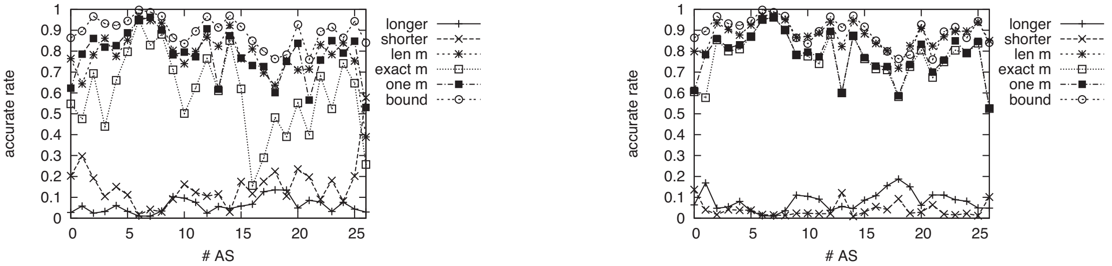


Fig. 5. Distribution of metric match in $K$ v.s. $K$

Furthermore, we also check the inference accuracy of the basic algorithm NoPath for the vantage point ASs. Note that the routing tables of these ASs are used to construct the AS graph, which is the basis of the AS path inference. The results are shown in the last row of Table I. It shows that the algorithm only yields an average accuracy of 40% in terms of exact match and 73% in terms of one match despite that fact that these AS paths have already been known. In contrast, our algorithms guarantee 100% inference accuracy for these ASs.

Figure 5 shows the distribution of average value of the performance metric match in K against the value of $K$ for different algorithms and scenarios. For the KnownPath algorithm, we observe a sharp increase in the first $5$ paths. Later, the improvement of the metric becomes marginal. Therefore, if we lower the objective of the single AS path inference problem to solve the easier potential AS path inference problem, the accuracy can be increased. This also further confirms the intuition behind the MultiPaths algorithm. Figure 5 shows that there is an 8% improvement in one match to $86\%$ if we allow multiple paths advertised to the neighbors. We can also observe the similar sharp increase of the inference accuracy in the first few paths. It shows that we can find nearly 86% paths within the first 10 paths. In addition, we found that the



(a) w/o first AS hop information (b) w/ first AS hop information

Fig. 4. Inference accuracy of the KNOWNPATH algorithm

incorporation of first hop AS information does not improve the performance of one match but it does improve the performance of exact match, i.e. it helps the algorithm to identify the best paths among the inferred paths.

### *B. The Impact of the Data Sets on Inference Accuracy*

Because of the limitation of the available data sources, we can evaluate the performance of our algorithms for a limited number of vantage point ASs and evaluation ASs only. It seems unconvincing to conclude that our algorithms could yield the similar inference accuracy for the rest of the ASs. However, extensive experiments, which we take to exam the inference accuracy of the algorithms for various combinations of the vantage point ASs and the evaluation ASs, show that the performance of the algorithm is independent of the selection of the vantage point ASs and the evaluation ASs. The major findings are summarized as follows.

At first, we fixed the number of the vantage point ASs and tried different sets of the vantage point ASs to infer the AS paths for several sets of evaluation ASs. We found that the influence of the selection of the AS sets on the inference accuracy is minor with regard to the number of the vantage point ASs.

Second, we examined the impact of the number of vantage point ASs on the performance. We find that by incorporating the BGP data of more vantage points, the accuracy is improved. But the improvement brought from the additional vantage becomes marginal when more vantage points are added. The observation suggests that we are able to achieve adequate performance by exploiting the BGP routing tables of a moderate number of vantage points.

In addition, we also checked the inference accuracy of the algorithms for the data sets collected on the other three dates: 7/10/2004, 12/22/2004 and 3/20/2005, which are randomly selected. It is found that the performance seems rather stable over time.

Therefore, we believe that the algorithms could guarantee the similar performance for most of the ASs in the Internet despite of the selection of the data sets. Due to the page limitation, the details can be found in our technical report [12].

## V. SUMMARIES

In this paper, we proposed several novel techniques to solve the single AS path inference problem and the potential AS path inference problem. At first, by introducing the topological information of the destination prefixes into the AS graph, we are able to infer the AS paths specific to prefixes instead of ASs. Second, by incorporating the known paths of the vantage points into inference process, our algorithms achieve an average accuracy of 60% in terms of exact match and an average accuracy of 81% in terms of length match for the single AS path inference problem. At last, by incorporating multiple potential paths in the inference process, our algorithm achieves an average accuracy of 86% in terms of one match for the potential AS path inference problem.

## ACKNOWLEDGMENT

The authors are supported by the NSF grant CNS-0325868, ANI-0208116 and ANI-0085848, and Alfred P. Sloan Fellowship.

## REFERENCES

[1] BGP Best Path Selection Algorithm. http://www.cisco.com/warp/public/459/25.shtml.

[2] CERNET BGP View Project. http://bgpview.6test.edu.cn/bgp-view/index.shtml.

[3] RIPE Route Information Service. http://www.ripe.net/ris/index.html.

[4] Route Views Project Page. http://www.routeviews.org.

[5] traceroute.org. http://www.traceroute.org.

[6] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-law Relationships of the Internet Topology. In *Proceedings of ACM SIGCOMM*, pages 251–262, 1999.

[7] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transaction on Networking*, 9(6):733–745, December 2001.

[8] R. Gao, C. Dovrolis, and E. Zegura. Interdomain Ingress Traffic Engineering through Optimized AS-Path Prepending. In *Proceedings of IFIP Networking*, 2005.

[9] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang. On AS-Level Path Inference. In *Proceedings of ACM SIGMETRICS*, Banff, Canada, June 2005.

[10] A. Nakao, L. Peterson, and A. Bavier. A Routing Underlay for Overlay Networks. In *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[11] J. Qiu. AS Path Inference Service. http://www.bgpvista.com/asinfer.php.

[12] J. Qiu and L. Gao. AS Path Inference by Exploiting Known AS Paths. Technical Report TR-05-CSE-04, Department of ECE, University of Massachusetts, Amherst, May 2005.

[13] Y. Rekhter, T. Li, and S. H. Ed. A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF, January 2006.

[14] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang. Exploring the Performance Benefits of End-to-End Path Switching. In *Proceedings of IEEE ICNP*, Berlin, Germany, Octocer 2004.

[15] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An Analysis of BGP Multiple Origin AS (MOAS) Conflicts. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, 2001.