On AS-Level Path Inference

Z. Morley Mao
Univ. of Michigan
zmao@umich.edu

Lili Qiu
Univ. of Texas, Austin
lili@cs.utexas.edu

Jia Wang
AT&T Labs – Research
jiawang@research.att.com

Yin Zhang
Univ. of Texas, Austin
yzhang@cs.utexas.edu

ABSTRACT

The ability to discover the AS-level path between two end-points is valuable for network diagnosis, performance optimization, and reliability enhancement. Virtually all existing techniques and tools for path discovery require direct access to the source. However, the uncooperative nature of the Internet makes it difficult to get direct access to any remote end-point. Path inference becomes challenging when we have no access to the source or the destination. Moveover even when we have access to the source and know the forward path, it is nontrivial to infer the reverse path, since the Internet routing is often asymmetric.

In this paper, we explore the feasibility of AS-level path inference without direct access to either end-points. We describe *RouteScope*–a tool for inferring AS-level paths by finding the shortest policy paths in an AS graph obtained from BGP tables collected from multiple vantage points. We identify two main factors that affect the path inference accuracy: the accuracy of AS relationship inference and the ability to determine the first AS hop. To address the issues, we propose two novel techniques: a new AS relationship inference algorithm, and a novel scheme to infer the first AS hop by exploiting the TTL information in IP packets. We evaluate the effectiveness of *RouteScope* using both BGP tables and the AS paths collected from public BGP gateways. Our results show that it achieves 70% - 88% accuracy in path inference.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols — Routing protocols

General Terms

Algorithms, Measurement, Experimentation

Keywords

AS-level path, network topology, Border Gateway Protocol, Internet Routing

1. INTRODUCTION

Discovering network paths is valuable for network operators and researchers to detect and diagnose problems, study routing protocol behavior, characterize end-to-end paths through the Internet, and optimize network performance. Moreover, several network applications, such as server selection and overlay routing, can benefit from the knowledge of AS path length between two end-points, since it has been shown that AS path length correlates with network performance [11, 12]. Several tools have been developed to identify the forwarding paths, such as traceroute [7], and the AS-level forwarding path discovery tool [10, 9].

However, an important question remains open: *Can we infer AS-level path without requiring direct access to the source?* Virtually all existing techniques and tools for path discovery require direct access to the source either by getting the BGP table or by launching active probes (*e.g.,traceroute*) from the source. Operational experience by large ISPs along with numerous measurement studies [13, 14] have shown that asymmetric routing is commonly used in today's Internet (*e.g.,* Paxson [14] shows that about 50% routes in his study are asymmetric). As a result, when we only have access to the source, we can not even answer question like "what's the path in the reverse direction, *i.e.,* from the destination to the source?". The problem becomes even more challenging when we have direct access to neither the source nor the destination.

In this work, we investigate the feasibility of inferring AS-level path without direct access to the source. Our approach leverages BGP table dumps from multiple vantage points, and publicly available traceroute servers. To the best of our knowledge, this is the first extensive study on this subject.

1.1 Challenges

The Internet consists of over 20,000 inter-connected Autonomous Systems (ASes) controlled by different administrative domains such as Internet Service Providers (ISPs), corporations, universities, and research institutions. Different ASes interact with each other in a very complex manner through the use of Border Gateway Protocol (BGP). BGP allows each individual administrative domain to specify its own routing policies. The enormous scale and the highly heterogeneous and uncooperative nature of the Internet have made it a major challenge for today's network operators to understand routing protocol behavior, and diagnose problems. One thing that would be invaluable for performance inference and fault diagnosis is a tool that can accurately discover the AS-level path between two end-points. It is challenging to identify network paths without direct access to end-points for the following reasons.

Asymmetric routing: The forward and reverse paths between a pair of nodes may not be the same due to policy-based inter-domain routing and traffic engineering mechanism such as hot-potato routing. Based on our extensive measurement on the AS paths between

351 public traceroute gateways and 125 BGP gateways (Table 2), we observe that over 60% AS paths are asymmetric, 2/3 of which differ in length (*i.e.,* AS hop counts). Our observations are consistent with previous work [14], which reports 50% router-level paths are asymmetric. Such high degree of asymmetry at both router-level and AS-level makes it difficult to know the path in the reverse direction based on the forward path.

Complicated routing policies: The current inter-domain routing protocol is Border Gateway Protocol (BGP), which is policy-driven. Rather than selecting the route with the shortest AS path, routers can apply complex policies to influence the selection of the *best* route for each prefix and decide whether to propagate this route to their neighbors. However, such routing policies are usually determined by the commercial relationship between ASes and the traffic engineering mechanism adopted within an AS. For example, a router may prefer to use routes learned from its customers than the one learned from its peers or providers. A router could also prepend its AS number multiple times to discourage the route being selected as the best route by making the route look longer when it propagates the route to its neighbors.

Multihoming: It has become more and more common that a network multihomes to multiple service providers for reliability, performance, and traffic engineering purposes. Without knowing the routing policies used by the multihomed network, it is very difficult to determine which upstream provider it selects to carry traffic for a given destination prefix at a given time.

1.2 Related Work

Understanding Internet topology is important for development and evaluation of networking protocols. It has received increasing interest in research community. The previous work can be broadly classified into two categories: path discovery and topology discovery.

Traceroute is the most widely used tool to discover end-to-end paths. It determines the interfaces on the forwarding path by sending a sequence of TTL-limited probes. Network operators benefit greatly from this tool to identify forwarding loops, blackholes, routing changes, and unexpected paths through the Internet [10]. More recently, [10, 9] develop techniques to discover AS-level forwarding paths. Both traceroute [7] and AS path discovery tools [10, 9], however, require direct access to the source. An open question, which we aim to address in this paper, is how to discover end-to-end paths without direct access to the source.

A number of techniques and tools have been developed to discover the Internet or ISP topologies. For example, Mercator [5] uses traceroute to infer an Internet map. It applies novel alias resolution heuristics and takes advantage of source-route capable routers to enhance the accuracy of the map. Rocketfuel [18] applies several techniques to directly measure ISP topologies, including the use of BGP routing tables, DNS, IP routing, and alias resolution. Their techniques have been shown to obtain fairly complete ISP maps [18]. The above active probing requires cooperation from routers (*e.g.,* traceroute, source-routing).

Researchers have also studied how to infer topology solely based on end-host measurements. For example, [3] proposes end-to-end packet-pair delay measurements to find correlation between nodes and hence infer topology. Mahajan [8] applies tomography-based approach to infer OSPF weights. The basic idea is to use tomography to find a feasible solution that is consistent with the observed routes. It requires enough observed paths for the given domain, so probably works better for large ISPs than for smaller ones.

1.3 Our Approach

The relationships among different ASes play an important role in determining the feasible forwarding paths. One natural approach to inferring AS-level path is to leverage recent advance on inferring AS relationship based on BGP tables taken from multiple vantage points [4, 19, 1]. While much progress have been made to understand AS relationships, several questions remain open: (i) how to leverage the inferred AS relationships to infer AS-level paths, and (ii) whether existing AS relationship inference schemes can allow us to infer AS path with high accuracy.

In our study, we infer AS paths by finding the shortest policy paths (i.e., the paths that conform with AS relationships) in an AS graph obtained from BGP tables at multiple vantage points. We find the accuracy of AS path inference based on the existing AS relationship inferences is limited. There are two main factors contributing to the inaccuracy: the limited accuracy in existing AS relationship inference and the wide-spread deployment of multihoming.

To address the first issue, we compare and evaluate all three existing AS relationship algorithms based on extensive Internet measurements. In addition, we propose a novel AS relationship inference algorithm, based on which we can infer 60 - 82% AS paths accurately (i.e., 60 - 82% of the actual paths match one of the inferred paths, and a similar fraction of the actual paths match the inferred AS path length).

To address the second issue, we develop a simple technique to identify the first-hop AS on the path from a source to a destination when we have access to the destination. Evaluation based on large-scale Internet measurements suggest that the technique is effective in identifying the first-hop AS, and improve the accuracy of AS path inference to 70 - 88%.

Our major contributions can be summarized as follow.

1. We develop a tool, *RouteScope*, to infer the AS-level forwarding paths between two end-points without direct access to either end-point. Our tool infers AS paths by finding the shortest policy paths in a AS graph obtained from BGP tables at multiple vantage points.

2. We describe a novel algorithm to infer AS relationships. Different from the previous work, we take advantage of both valid and invalid forwarding paths to make inference.

3. We propose the ability to infer AS-level path as a new metric to evaluate the accuracy of AS relationship inference.

4. We use a diverse set of data to evaluate several AS relationship inference algorithms, including ours. We extensively evaluate the well-known AS relationship inference algorithms to understand their accuracy and robustness in inferring AS-level paths.

5. We present a novel technique for inferring the first AS hop by exploiting the TTL information contained in IP packet.

6. We demonstrate the effectiveness of *RouteScope* using both BGP table dumps and the AS paths collected using a large number of public BGP gateways. The results are very promising: it achieves up to 88% accuracy in AS-level path inference.

1.4 Paper Outline

The rest of the paper is organized as follows. In Section 2, we describe our data measurement methodology. In Section 3 we present the high-level approach of AS path inference, and evaluate the assumptions used in our approach. In Section 4, we show that the

Organization	ASN	Location	Dates in 2004
Univ. of Washington	73	WA, USA	Oct. 31
PSG home network	3130	WA, USA	Oct. 19
ArosNet	6521	UT, USA	Oct. 19
OPTUSCOM-AS01-AU	7474	Australia	Oct. 28
Williams Communications Group	7911	OK, USA	Oct. 28
Vineyard.NET	10781	MA, USA	Oct. 19
Peak Web Hosting	22208	CA, USA	Oct. 19
EUNET-FINLAND	6667	Finland	Oct. 28
COLT Telecom	8220	Europe	Oct. 19
MainzKom Telekom	15837	Germany	Oct. 28
Manila Internet Exchange	9670	Philippines	Oct. 28
Telkom SA Ltd.	5713	South Africa	Oct. 28
RIPE00-12	many	mostly Europe	Oct. 19
RouteViews	many	mostly USA	Oct. 19

Table 1: Location of BGP table dumps.

existing AS relationship inference schemes yield limited accuracy in inferring AS paths. In Section 5, we propose a novel algorithm to infer AS relationship, and show that our new algorithm helps improve AS path inference. In Section 6, we develop and evaluate a heuristic to infer the first AS hop to further improve the accuracy of path inference. Finally, we conclude with discussion and future work in Section 7 and Section 8.

2. DATA COLLECTION

In this section, we describe our measurement methodology. Since our goal is to understand path selection and inference in the Internet, we try to get as diverse data as possible by taking advantage of multiple data sources. The data we collected include BGP table dumps from various locations, traceroute results using publicly available traceroute servers, and BGP query results.

The BGP table dumps are obtained from a large number of locations world-wide, as summarized in Table 1. These BGP tables altogether give a fairly complete view of the Internet at AS-level. We also obtained BGP tables from the BGP route reflectors and border routers at peering links of a tier-1 ISP backbone network. These tables allow us to evaluate the accuracy of AS path inference. Combining all the data together results in an AS graph with 20,699 nodes and 53,954 edges. A recent work [21] collect the Internet AS-level topology by using BGP routing updates in addition to routing tables. The number of ASes in their data is comparable to ours. This combined with the fact that the number of ASes observed from large ISPs was around 16,000 during November 2003 [6] suggests that our AS graph is fairly complete in terms of the number of ASes. The number of edges reported in [21] is higher, slightly over 60,000, due to the use of BGP routing updates. We plan to investigate in the near future the effect of a more complete AS graph on our results.

In addition, we obtain router-level paths by querying 351 public traceroute servers [20], which are spread across over 50 countries. We have them traceroute to each other to obtain router-level paths and then convert router-level paths to AS-level paths by applying the technique proposed in [10, 9]. To diversify our data, we also obtain additional AS-level paths from a collection of 125 public gateways (mostly from the LookingGlass sites listed on [20]) that support BGP query "`show ip bgp`". We send out query to each BGP gateway to obtain the AS-level paths from that gateway to the remaining gateways. We exclude responses that state "no best route" or "network not in table" in our experiments.

To evaluate the accuracy of first AS hop inference, we make use

	# servers	# countries	# ASes
Traceroute servers	351	50+	304
BGP servers	125	30+	121
PlanetLab nodes	85	18	85

Table 2: Measurement infrastructures used in our study.

of a collection of 85 PlanetLab nodes [15], and 351 public traceroute servers spread across over 50 countries. The locations of these nodes are summarized in Table 2.

3. INFERRING AS PATHS

In this section, we describe a simple algorithm for inferring AS paths between two end-hosts without direct access to either host. The algorithm leverages the BGP tables collected from multiple vantage points.

3.1 Assumptions & Validation

Our inference algorithm is based on the following assumptions, which we will validate using the data described in Section 2. We recognize that deviations from our assumptions do exist in practice; however, these assumptions constitute the *common* cases. As part of our future work, we plan to further reduce inaccuracy caused by occasional violations of these assumptions.

1. *Explicit AS relationship:* The relationship between two ASes is clearly defined as one of the followings: peers, customer and provider.

2. *Shortest policy AS path preferred:* The actual path is the shortest one among all the paths that follow the routing policy. The routing policy is in the form of *CustomerProvider* PeerPeer? ProviderCustomer** (denoted as *AS path "valley-free" rule* [4]), where "*" represents zero or more occurrence of such type of AS edge and "?" represents at most one occurrence of such type of AS edge.

3. *Uniform routing policy within an AS:* Paths from all sources in a given AS to the same destination prefix have the same number of AS hops.

4. *AS-destination based uniform routing:* Paths from a given AS to all destinations in another AS have the same number of AS hops.

5. *Stability.* The AS-level paths are relatively stable and do not change significantly between the time the BGP tables are captured and the time BGP paths are queried.

The assumption (1) is used by many previous work, such as [4, 19, 1]. The assumption (5) has been shown to hold, especially for popular destinations [16, 10]. Below we evaluate to what extent the assumptions (2) - (4) hold based on measurement data collected in in September 2003.

To test the assumption (2), we analyze BGP tables from 17 border routers at the peering links of a tier-1 ISP backbone network in North America. These routers are selected to be geographically diverse. The BGP table obtained from each router contains the best route selected by the router as well as the alternative routes to each destination network. Each BGP table contains routes to over 150K distinct destination prefixes. All the routes are specified at AS level. For each destination network, we compute two metrics to characterize the AS path length of the selected best route and all the alternative routes: *policy length* (*i.e.,* the number of ASes including prepended ASes on the path) and *actual length* (*i.e.,* the number of unique ASes on the path). We observe that the best routes tend to be the shortest among all available routes. About 16% of the destination prefixes have a single available route to the destination. They do not have alternative routes and the available routes are selected as the best routes. For the remaining destination prefixes which have alternative routes, a destination may have up to 10 available routes. Only 1.37% of the best routes have a longer policy length than the alternative routes; and only 0.65% of the best routes have a longer actual length than their corresponding alternative routes. The figures indicate that the shortest path is highly likely to be selected as the best route regardless of prepending and other routing policies from the view of a large tier-1 ISP; moreover the AS prepending does not have a significant impact on the best AS path selection. Therefore it is promising to infer AS paths by computing the shortest policy paths.

To evaluate the validity of the assumption (3), we analyze the BGP tables from 21 route reflectors of a tier-1 ISP backbone and from various vantage points listed in Table 1. We find that the paths from various sources in an AS to a destination prefix have the same policy length. Only about 1.5% of the paths differ in their actual lengths. The different sources do not significantly affect the AS path length of the best route. This suggests the assumption (3) holds for most of the paths.

Finally, we study the BGP tables from various vantage points listed in Table 1. We found that 60% of the destination ASes have more than one prefix. Over 95% of the distinct prefixes belong to such destination ASes. 10-20% paths from a single source to different prefixes in a destination AS differ in lengths. The paths from a source to a destination AS may have up to 7 distinct policy lengths and up to 5 distinct actual lengths. This indicates that the best path selection is based on destination prefixes instead of destination ASes. However, in over 84% cases, there is no difference in path length from a source to all destinations in the same AS. This suggests that the assumption (4) applies to most paths, however, there is an inherent limit on the accuracy of path inference at the AS-level.

3.2 AS Path Inference Algorithm

Based on the above observations, a natural approach to inferring AS path is to combine BGP tables from multiple vantage points to produce a fairly complete AS graph and then simulate shortest AS-hop-count routing on the graph subject to policies dictated by AS relationships.

First, to enforce the AS path rules described earlier in this section, we apply the existing algorithms proposed in [4, 19, 1] to infer AS relationships for the nodes in the AS graph. As we will show in Section 4, among the three existing AS relationship inference algorithms, [1] gives the most accurate inference. A new algorithm is also proposed in Section 5 to achieve an even higher accuracy in AS relationship inference.

Next based on the inferred AS relationships, edges in the AS graph are classified as one of the following three categories: (i) custom-provider link (UP link), (ii) provider-custom link (DOWN link), (iii) peering links (FLAT link). (We exclude edges with unknown AS relationships.) We apply the algorithm as shown in Figure 1 to derive the set of all shortest policy paths (*i.e.,* shortest paths among all paths that conform to the AS relationship) between pairs of nodes.

foreach source-destination pair find all shortest uphill paths
for each pair (src, dst)
  // cost without FLAT link
  $cost0(src, dst) = \min_m\{\ dist(src, m) + dist(dst, m)\ \}$
  where m is a node
  // cost with FLAT link
  $cost1(src, dst) = \min_{m,p}\{\ dist(src, p) + dist(dst, m) + 1\ \}$
  where m is a node, p is a peer of m
  $cost(src, dst) = \min\{\ cost0(src, dst), cost1(src, dst)\ \}$
find all shortest policy paths between src and dst by concatenating
  $uphill(src, m)$, (m, p), $reverse(uphill(dst, p))$, or
  $uphill(src, m)$, $reverse(uphill(dst, m))$

Figure 1: Compute all shortest policy paths

Inference	# invalid paths	% invalid paths
Gao	468568	25.0%
SARK	556383	29.73%
BPP	74737	3.99%

Table 3: The number of paths that violate the AS path "valley-free" rule under three AS relationship inference algorithms

4. EVALUATING AS PATH INFERENCE USING EXISTING AS RELATIONSHIP INFERENCE

In this section, we first give a brief overview of existing AS relationship inference schemes. Then we apply them to infer AS paths, and compare the inferred AS paths with the actual AS paths obtained from BGP table dumps and BGP gateway queries. Our results show that the inference accuracy varies with the AS relationship inference schemes and the location of vantage points.

4.1 Accuracy of Existing AS Relationship Inference

There are three existing algorithms for inferring relationship between a pair of ASes. They are all based on the information obtained from BGP tables at multiple vantage points.

Gao: Gao [4] proposes the first algorithm to infer AS relationships. The algorithm makes inference based on the degree of ASes along with the AS paths extracted from the BGP tables.

SARK: Subramanian *et al.* [19] simplifies the problem in [4], and formulates it as a minimization problem. They develop a heuristic by leveraging multiple vantage points.

BPP: Recently, Battista *et al.* [1] prove that the problem formulated in [19] is NP-complete. They map the problem into a 2SAT formulation, and use the insights from 2SAT to develop heuristics for inferring AS relationships that yield a small number of invalid paths.

Now we study the accuracy of the above three AS relationship inference algorithms. We construct an AS graph using the BGP tables listed in Table 1. The resulting AS graph contains 20,699 nodes and 53,954 edges as mentioned in Section 2.

Table 3 shows the number of invalid paths (*i.e.,* the paths that violates the AS path "valley-free" rule described in Section 3) according to the inferred AS relationships. We observe that BPP yields the smallest number of invalid paths. Both Gao and SARK have a significant number of invalid paths, around 25 $\sim$ 30%.

Next we evaluate the consistency among these three algorithms as follow. For every two inference algorithms, we compute the number of edges (*i.e.,* pairs of ASes) that are assigned with the

	Common peer-peer	Common provider-customer
Gao vs. SARK	229 (3.63%, 36.12%)	41730 (89.43%, 94.68%)
Gao vs. BPP	5959 (94.51%, 48.42%)	39606 (84.87%, 97.74%)
SARK vs. BPP	334 (52.68%, 2.71%)	33752 (85.66%, 93.17%)

Table 4: The number of edges that are assigned the same relationship by given inference algorithms.

same AS relationship under the two algorithms, and summarize the results in Table 4. The numbers in parentheses denote the number of common peer-peer (or provider-customer) edges divided by the total number of peer-peer (or provider-customer) edges inferred by the corresponding algorithms. We observe that the number of peer-peer and provider-customer edges that are inferred by these three algorithms varies, especially for peer-peer edges. The consistency is quite high for provider-customer edges, ranging from 85% - 95%. In comparison, peer-peer edges share significantly lower common assignments. This suggests these algorithms are better at inferring provider-customer relationships than peer-peer relationships.

4.2 Comparing Inferred AS Paths with BGP Tables

Next we examine whether the existing AS relationship inference algorithms enable us to accurately infer AS paths.

We evaluate the accuracy of AS path as follow. First, we selectively remove BGP tables collected from a few vantage points. Then we apply the AS relationship inference algorithms to the set of paths from the remaining BGP tables. Next, for each AS path in the removed BGP tables, we compute the inferred AS path using our algorithm described in Section 3 based on the inferred AS relationships. Finally, we compare the inferred AS paths with the actual AS paths in the removed BGP tables.

In our experiments, we use the set of BGP tables listed in Table 1. We select three groups of vantage points to evaluate the accuracy of AS path inference: one tier-1 network (AS7018), one tier-2 network (AS2152), and one tier-3 network (AS8121). Table 5 shows the inference accuracy when we remove BGP tables obtained from each of these three ASes and all its customer ASes including multihomed customers.

We classify match cases into the following three categories: (i) the actual path matches one of the inferred paths (denoted as "Match"), (ii) the length of the actual paths matches the length of the inferred paths (denoted as "Match Length"), (iii) there is a single inferred AS path between a pair of ASes, and the inferred path is identical to the actual path (denoted as "Exact Match").

In the BGP table from AS7018, there are altogether 18,085 unique paths. Among them, there are 67 $\sim$ 84% matches, around 30% exact matches, and 79-85% length-matches. A few comments follow.

First, the inferred paths that have the same length as the actual paths but do not exactly match every AS hop are not necessarily mismatches. This is because the actual AS paths in the BGP table are only from a single location within an AS, and other locations inside the AS may use different paths, but typically of the same length due to uniform policies across the entire AS. Thus the other locations may use one of the inferred paths. We have observed that such phenomenon occur frequently at several tier-1 ISPs.

Second, although in many of the "Match" or "Match Length" cases we cannot uniquely determine the actual AS path in use, several applications, such as server selection and overlay routing, can already benefit from the AS path length information (e.g., selecting the closest server in terms of AS hop count).

Third, most of the mismatches are due to inferred paths being shorter than the actual paths. They account for 15$\sim$18% of the paths in the BGP table from AS7018 (denoted as "Shorter" cases). We will elaborate the reasons for such discrepancies later.

	Total	Match	Match length	Exact match	Shorter	Longer
	AS7018					
Gao	18085	77%	80%	33%	18%	2%
SARK	18085	67%	79%	34%	15%	4%
BPP	18085	84%	85%	37%	15%	0%
	AS2152					
Gao	11990	62%	65%	10%	34%	1%
SARK	11990	48%	57%	29%	40%	3%
BPP	11990	67%	67%	12%	33%	0%
	AS8121					
Gao	15757	16%	27%	3%	69%	4%
SARK	15757	14%	23%	3%	72%	4%
BPP	15757	18%	30%	3%	66%	5%

Table 5: Evaluating AS path inference using BGP tables from selected vantage points.

Comparing the inference accuracy across different ASes, we observe that the inference accuracy is reasonably high (up to 85%) for AS7018, a tier-1 ISP. This is expected as our input BGP tables to the AS relationship inference algorithm contain many other tier-1 ISPs which all peer with AS7018. Thus, it is not difficult to infer paths originating from AS7018, which are mostly advertised to other tier-1 ISPs and in turn visible in the input BGP tables. In comparison, the inference result for AS2152, a tier-2 network, is worse, 57 $\sim$ 67% of the 11,990 unique paths fall into the "match length" cases. The inference accuracy for AS8121, a tier-3 network, is even lower – only 23% of 15,757 unique paths fall in to the "Match length" cases. A high percentage of the inferred paths are shorter than the actual BGP paths, which suggests that AS8121 may use special routing policies to prefer longer paths. One possible enhancement to our tool is to output longer paths that conform to the routing policies instead of shortest policy paths if we know the routing policies of the source AS. In addition, for all three vantage points, BPP yields more accurate AS path inference than Gao and SARK.

4.3 Comparing Inferred AS Paths with BGP Gateways

We also compare the inferred AS paths with the actual AS paths queried from BGP gateways. Table 6 shows the inference results for paths between BGP gateways across the world. 121 distinct ASes serve both as source and destination ASes in our experiments. As a result, we identify 2,457 unique AS paths between BGP gateways. For each path, we compute the inferred paths based on all the BGP tables in Table 1. We observe that the overall accuracy of inference is quite low across all three AS relationship inference algorithms – only 18 $\sim$ 38% and 29 $\sim$ 51% of the examined paths fall into "Match" and "Match length" cases, respectively. Compared with the corresponding figures in Table 5, we observe a lot more cases where the inferred paths are longer than the actual paths (denoted as "Longer" cases). This is probably due to the inaccuracy in the AS relationship inference, causing the shorter actual BGP paths to be considered invalid in our AS graph. In addition, we find that BPP yields the lowest accuracy among the three algorithms; Gao and SARK perform better, yielding similar accuracy.

We also obtain AS paths from 7 BGP gateways located in US to 3,343 unique prefixes assigned to universities, which belong to 469 distinct ASes, all of them located in US. The inference results

	Total	Match	Match length	Exact match	Shorter	Longer
Gao	2457	30%	51%	21%	15%	35%
SARK	2457	38%	61%	24%	20%	19%
BPP	2457	18%	29%	15%	5%	66%

Table 6: Evaluating AS path inference using BGP paths from BGP gateways across the world.

	Total	Match	Match length	Exact match	Shorter	Longer
Gao	1907	24%	43%	16%	18%	40%
SARK	1907	40%	57%	24%	24%	19%
BPP	1907	22%	42%	18%	10%	48%

Table 7: Evaluating AS path inference using BGP paths from BGP gateways in US.

are shown in Table 7. There are a total of 1,907 unique AS paths. The figures are comparable to those shown in Table 6: the AS path inference accuracy is low.

4.4 Possible Causes of Inference Mismatches

We now examine the mismatches in detail to identify possible causes that account for these mismatches.

4.4.1 Inaccuracy in AS Relationship Inference

One of the reasons for inaccurate AS path inference is inaccurate AS relationship inference. One way in which this inaccuracy manifests itself is through the mismatches due to inferred paths being longer than actual paths. In such cases, the inferred path is longer than the actual path, which appears to violate the inferred AS relationship. As Table 6 shows, 19% - 66% of paths fall into this category. Moreover, Table 4 shows that there is significant inconsistency among the inference results from the three AS relationship inference algorithms. This further confirms that the inferred AS relationships have limited accuracy.

4.4.2 First Hop Analysis – Multihoming

Example
Source AS = A, Destination AS = D
Inferred path = $AGHD$, Actual path = $ABCFD$

Our analysis of mismatched paths reveals that more than half of the mismatches occur right at the very first hop AS. As shown in the above example, AS A can choose between two upstream providers: B and G. Due to traffic engineering or load-balancing, AS A may choose AS B instead of AS G as the first hop AS, making the actual path longer than the shortest policy path.

As we will show in Section 6, the ratio for "Match length" cases can be improved from 73% to 88% when we infer AS paths using BGP gateways given the first hop AS in use. That is, if we can correctly infer the first hop AS, we can eliminate around 15% of the mismatches.

4.4.3 Summary

The above results suggest that in order to improve the accuracy of AS path inference, we should address two challenges: (i) how to improve AS relationship inference, and (ii) how to infer the first AS hop. In Section 5 and Section 6 we investigate these two issues in turn.

5. A NEW AS RELATIONSHIP INFERENCE ALGORITHM

The previous section shows that the existing AS relationship inference algorithms yield limited accuracy in AS path inference. In this section, we propose a new algorithm to infer AS relationship, and show it can improve the accuracy in AS paths inference.

5.1 Problem Formulation

Let $G = (V, E)$ be the directed graph that consists of both directions of every edge that is contained in some BGP paths.

For any directed edge $e_i =< x, y >$, we introduce a variable, $relation(e_i)$ to indicate whether the link is FLAT, UP, or DOWN.

$$relation(e_i) = \begin{cases} 1 & if\ e_i\ is\ customer - provider \\ 2 & if\ x\ and\ y\ are\ peers \\ 3 & if\ e_i\ is\ provider - customer \end{cases}$$

We then introduce the following constraints:

1. Valid relationship variable: For every edge e_i, let e_r be the edge corresponding to its reverse direction.

$$relation(e_i) + relation(e_r) = 4 \quad (1)$$

$$relation(e_i) \in \{1, 2, 3\} \quad (2)$$

$$relation(e_r) \in \{1, 2, 3\} \quad (3)$$

2. Every path in use is valley free, or equivalently, every FLAT or DOWN link is followed by a DOWN link. For any (e_i, e_j) appearing on some valid BGP path,

$$relation(e_i) = 1 \lor relation(e_j) = 3 \quad (4)$$

3. Given any (src, dst), if there is a path P from src to dst and it is shorter (in terms of AS hop count) than the actual path we see, then P is not valley-free. In other words, there exists (e_i, e_j) on P s.t.

$$relation(e_i) \neq 1 \land relation(e_j) \neq 3 \quad (5)$$

To reduce the number of constraints we generate, we only add the non-valley-free constraints for the paths that have the shortest AS hop-count (without considering AS-relationship) and shorter than the actual routing paths.

Now our goal is to find $relation(e_i)$ to satisfy as many constraints, shown above, as possible. Note that different from the previous work, which only restrict observed paths to be valley-free, we also derive additional constraints from the unused paths that are shorter than the actual paths. These additional constraints help us to get better relationship inference as we will show later.

5.2 Our Approach

Initialization: We initialize all links to be DOWN links (i.e., provider-customer links), because most of the paths from our vantage points are towards customers.

Iteration: We use the random walk algorithm developed by Selman et al. [17]. Figures 2 and 3 show our pseudo-code. We use $walk\_prob = 0.5$, $maxFlips = 15000$, and $maxNoProg = 1000$ in our evaluation.

We make the following modifications and optimizations. First, different from [17], we can handle non-binary variables. Second, to reduce the problem size, we repeatedly apply the stub AS removal procedure as shown in Figure 4, where stub ASes are the sinks of directed graph $G = (V, E)$ with out degree of zero. This procedure reduces the number of nodes and edges by up to 2 orders of

```
iteration = 0; num_no_prog = 0;
while (iteration < maxFlips or numNoProg < maxNoProg) {
        if (rand() < walk_prob)
           // randomly select an unsatisfied edge e
           // and change Relation(e)
           progress = random_walk();
        else
           // For all unsatisfied edge and all possible relationships,
           // find the change that results in largest reduction
           // in number of unsatisfied constraints
           progress = greedy();
        iteration++;
        if (progress = 0) numNoProg = 0;
        else numNoProg++;
}
```

Figure 2: Using random walk to find the AS-relationships that satisfy the constraints in Section 5.1.

```
greedy() {
    max_prog = 0
    for each edge e {
       if numUnsatConsWithEdge(e) < max_prog
          continue
       else
          for each relation r ≠ currentRelation(e)
              rel(e) = r
              rel(reverse(e)) = 4 − r // according to eq. (1)
              prog = reducedNumOfUnsatCons();
              if (prog < max_prog)
                 max_prog = prog
                 action = "e => r"
    }
}
```

Figure 3: Greedy step in random walk

```
S = stubAS(G); // find stub ASes
while (S is not empty) {
        for (each n in S) {
            for (each < p, n > in E) {
                mark p as the provider of n;
            }
        }
        G = subgraph(G, V − S); // remove S from G
        S = stubAS(G); // find ASes whose out-degree is 0
}
```

Figure 4: Removal of stub AS repeatedly

magnitude. Third, to reduce the number of tests required to find a greedy move, we skip over the edge whose number of unsatisfied constraints is fewer than max_prog as shown in Figure 3, since changing the relationship assignment for the edge cannot reduce the number of unsatisfied constraints by more than max_prog. This leads to a speed-up by up to two orders of magnitude.

5.3 Evaluation

In this section, we evaluate the accuracy of our AS relationship inference.

First, we apply the new AS relationship inference algorithm to the BGP tables shown in Table 1, and compute the number of paths that violate the AS path "valley-free" rule. We find the number of invalid paths is 115,865, which account for 7.35% of all paths. This accuracy is comparable with BPP, the best relationship inference among the three existing ones.

	Total	Match	Match length	Exact match	Shorter	Longer
AS7018	18085	82%	83%	35%	17%	0%
AS2152	11990	64%	64%	10%	35%	0%
AS8121	15757	16%	27%	3%	69%	4%

Table 8: Evaluating the new AS path inference algorithm using BGP tables from selected vantage points.

	Total	Match	Match length	Exact match	Shorter	Longer
All BGP gateways	2457	70%	73%	30%	22%	4%
US BGP gateways	1907	60%	62%	27%	34%	4%

Table 9: Evaluating the new AS path inference algorithm using BGP paths from BGP gateways.

Next, we apply our AS relationship inference to infer AS path, and compare the inferred AS paths with the actual paths from BGP tables. Table 8 summarizes our results. Comparing it with the accuracy of the other three algorithms, shown in Table 5, we observe that its accuracy is comparable with the best of the other three in all cases.

Finally, we compare the inferred AS paths with the paths queried from BGP gateways, and summarize the results in Table 9. It can infer over 60% - 70% paths accurately, much higher than the alternatives, whose accuracy is 18-40%.

To summarize, in this section we present a novel algorithm to infer AS relationships. Our measurement results show that it is competitive: its accuracy is comparable to the best of the three alternatives when compared with the paths from the BGP tables, and significantly higher than the others when compared with the paths queried from the BGP gateways.

6. INFERRING THE FIRST AS HOP

As shown in Section 4.4.2, another important factor of AS path inference is the ability to infer the first-hop AS. Motivated by this observation, in this section we consider the problem of how to infer the first-hop AS on the path from source IP address S (in AS **S**) to destination IP D (in AS **D**), with only direct access to the destination D. Such inference not only enables us to improve AS path inference, but also allow us to understand how multihomed customers utilize different access links.

6.1 Inference Algorithms

Figure 6 shows our inference algorithm. At a high-level, it consists of two steps: (i) gather a list of candidate first hop ASes from S, and (ii) identify the transition point T (*i.e.,* the last IP hop before entering AS **S**) that is likely to be on the path from S to D by testing whether the following condition is satisfied, where $hc(node1, node2)$ denotes the IP hop count from $node1$ to $node2$.

$$hc(S,T) + hc(T,D) = hc(S,D) \tag{6}$$

We now describe each step in details. To obtain a list of candidate first hop ASes, we launch traceroute probes from multiple public traceroute servers towards S. If the locations from which we launch traceroute are diverse enough, we can discover at least one path whose last-hop AS matches the first-hop AS for the path from S to D and whose *transition point* appears on the path from S to D. In practice, we may miss some candidate ASes because our probing points do not cover enough paths or the first-hop AS

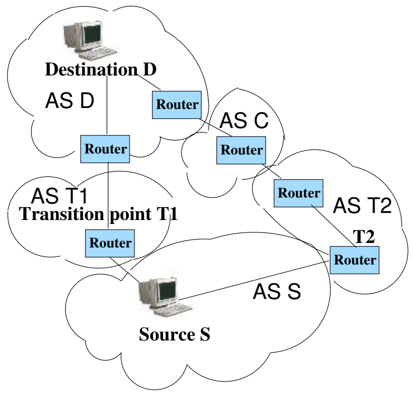

Figure 5: Our technique of inferring first-hop AS.

Goal: infer first hop AS from source IP S to destination IP D when we only have direct access to D

Steps:

1. Use public traceroute servers to launch *traceroute* probes to S, map each router-level forwarding path to AS-level path ([10, 9]), record each last-hop AS **T** and transition point T // T: border router of AS **T** that is directly connected to AS **S**
2. Find transition point T^* that is most likely to be on the path from S to D, and report its AS as the inferred first-hop AS
   a) Apply heuristic to infer
      $hc(S, D)$: IP hop count from S to D
      $hc(S, T)$: IP hop count from S to transition point T
      $hc(T, D)$: IP hop count from transition point T to D
   b) Find transition point T^* that minimizes
      $$| hc(S, T) + hc(T, D) - hc(S, D) |$$

Figure 6: Algorithm for inferring the first hop AS

only carries the out-bound traffic from S but no in-bound traffic (so we cannot discover it by launching probes towards S). However, our experimental results suggest that at least for the sites we have tested, the inference algorithm based on the above assumption has a fairly high success rate.

Next, we infer the first-hop AS for the path from S to D by identifying a transition point that is most likely to be on the path from S to D. Specifically, for each transition point T, we first apply heuristics (see below) to infer the IP hop counts $hc(S, T)$, $hc(T, D)$, and $hc(S, D)$. We then test whether T satisfies Equation 6.

Note that Equation 6 is a necessary but insufficient condition for T to be on the path from S to D. The hope is that when the number of different last-hop ASes is small, the above test is sufficient to exclude all transition points that are not on the path from S to D. Another important note is that without direct access to S, it can be very difficult to accurately estimate the hop counts. As a result, Equation 6 may not hold exactly even if T is indeed on the path from S to D. To account for such inaccuracy, we pick the transition point T^* that minimizes $|hc(S, T) + hc(T, D) - hc(S, D)|$ instead of strictly satisfying Equation 6.

Below we present our heuristics for inferring hop counts $hc(S, T)$, $hc(T, D)$, and $hc(S, D)$. We estimate $hc(S, T)$ using $hc(T, S)$, which is available through the router-level forwarding path. This assumes that the path between two routers S and T within the same AS is symmetric in terms of hop count, *i.e.,* $hc(T, S) = hc(S, T)$. This is reasonable for shortest path based IGP routing protocols like OSPF and IS-IS, because in practice the two directions of a link is assigned with the same weight, which implies that one can obtain a shortest path from S to T by reversing the shortest path from T to S. Our results suggest that this heuristic works very well in practice.

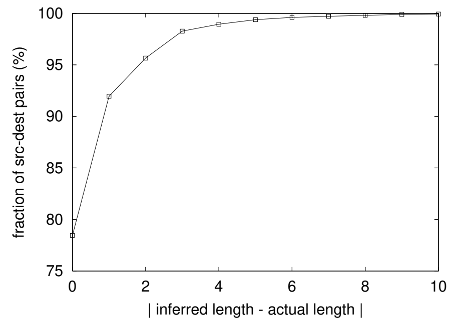

Figure 7: The distribution of difference between inferred and actual hop counts.

To infer $hc(T, D)$ and $hc(S, D)$, we take advantage of the Time-To-Live (TTL) value contained in IP packet. Specifically, when we send a *ping* packet (*i.e.,* an ICMP echo request) to a remote host H, H sends a response back with the TTL value of the response packet initialized by H and decremented by one at each router on the return path. Therefore, if we can guess the initial TTL value (TTL_0), then based on the TTL value of the received response packet (TTL_1), we can estimate the path length from H to D as $(TTL_0 - TTL_1 + 1)$. In practice, there are only a small number of common values for TTL_0. The most common values and the corresponding operating systems are 32 (Windows 95/98/Me), 64 (Linux, Compaq Tru64), 128 (Windows NT/2000/XP), and 255 (most UNIX systems). If we assume that the reverse path has fewer than 32 hops, which is often the case in today's Internet, then we can easily estimate TTL_0 from TTL_1 using the formula $TTL_0 = \min\{255,\ 32 \cdot \lceil TTL_1/32 \rceil\}$.

6.2 Evaluation

In this section, we evaluate the accuracy of our algorithm for inferring the first hop AS. We conduct a large-scale Internet experiment using a collection of 85 PlanetLab nodes and 351 public traceroute gateways shown in Table 2 in the following four steps.

1. Apply our inference algorithm to infer the first hop AS from each traceroute gateway to each PlanetLab node;

2. Launch *traceroute* probes from each traceroute gateway to each PlanetLab node to obtain the actual router-level path;

3. Compare the inferred and the actual hop count from traceroute gateways to PlanetLab nodes;

4. Use IP-to-AS mapping obtained in [9] to map the router-level forwarding paths into AS path to extract the actual first hop AS, and then compare it against the inferred first hop AS.

6.2.1 Accuracy of Hop Count Inference

Since the heuristic for inferring the reverse path hop count is critical to the overall inference algorithm, we first evaluate the accuracy of this heuristic. Figure 7 shows the distribution of the difference between inferred and actual hop counts for paths from traceroute gateways to PlanetLab nodes. The difference is at most 1 for 92% of the paths. So the inferred hop counts are fairly accurate.

6.2.2 Accuracy of First-hop AS Inference

Location	Total	Match length	Improvement
AS 7018	18085	86%	3%
AS 2152	11990	76%	12%
AS 8121	15757	48%	21%
US BGP gateways	1907	70%	8%
All BGP gateways	2457	88%	15%

Table 10: Accuracy of inferring AS paths, when the first AS hop is given.

Notation	Link type	AS relationship
$-$	FLAT	Peering
$>$	DOWN	Provider-customer
$<$	UP	Customer-provider

Table 11: AS relationship notations used in our discussion.

Next we evaluate the accuracy of first-hop AS inference as follows. We focus on paths with two candidate first hop ASes because this is a common case—among the 2,500 prefixes with multiple original ASes (MOASes) in all of our BGP tables, 2,440 (97.6%) have two original ASes. In our experiments, there are 2,415 paths with two candidate first hop ASes. Our heuristic is able to correctly identify the first hop AS for 2,028 (84%) paths. For the remaining 387 paths, 346 (14.3%) have ties, *i.e.,* a transition point T in the actual first hop AS also minimizes $|hc(S,T)+hc(T,D)-hc(S,D)|$, but is not chosen because we break the tie incorrectly (we currently break a tie by simply picking the AS with the largest number of transition points). As the number of candidate first hop ASes increases to 3, the actual first hop AS minimizes the difference between $hc(S,D)$ and $hc(S,T) + hc(T,D)$ about 75% of the time. However, the number of ties significantly increases and thereby degrading the accuracy to about 65%. A lot of the ties are likely caused by aliases, *i.e.,* different interfaces on the same router have different IP addresses. Therefore, it is likely that we can further improve the accuracy of our inference algorithm by applying alias resolution techniques such as those used in Rocketfuel [18] and Mercator [5].

6.2.3 *Accuracy of AS Path Inference Given the First AS Hop*

Finally we examine how much AS path inference can benefit from knowing the first hop. To answer this question, we assume the first AS hop is given, and compare the inferred AS paths with the AS paths extracted from BGP tables and AS paths queried from BGP gateways. Table 10 shows path inference accuracy based on our AS relationship inference presented in Section 5. As we can see, knowing the first hop helps to improve the inference accuracy by 3 - 21%, bringing the final accuracy up to 70 - 88%.

7. DISCUSSION

In this section, we discuss other challenges involved in AS path inference. We use the notation listed in Table 11.

7.1 Complicated AS Relationships

In practice, the AS relationships are more complicated than what we assumed in our inference algorithm. As we have shown in Sections 3 and 4, over 15% of the cases where the paths between a pair of source and destination ASes are not the shortest AS path conforming to the "valley-free" rules and about 2% of the BGP AS paths even violate the "valley-free" rules. Such complication in AS relationships limits the accuracy of Internet path inference.

Scenario 1: *Two consecutive FLAT links.* We have observed a few AS paths contain two consecutive FLAT links. Consider path "$A-B-C$", where ASes A and C are both large regional ISPs and AS B is an international ISP. This could happen if ASes A and C do not have direct relationship (possibly due to geographical reasons), they can reach each other through AS B. Since ASes A and C are large, AS B may have incentive to form peering relationship with both of them. This would result in two consecutive FLAT links, $A-B$ and $B-C$, in the AS paths.

Scenario 2: *DOWN link followed by FLAT link.* Some of the AS paths contain a DOWN link followed by a FLAT link. This could happen in a case similar to Scenario 1, except that ASes A and B are provider and customer, instead of peers. For example, consider path $A > B - C$. AS B has presence in Europe, but its presence is not significant enough for B to establish peering relationship with any local large ISPs in Europe. So instead B becomes a customer of local ISP A in Europe. Note that this shows that tier-1 ISP in North America may still have upstream provider in a global AS graph.

Scenario 3: *FLAT link followed by UP link.* We also observe a few cases where an AS path contains a FLAT link followed by an UP link (*i.e.,* $C - B < A$). This is just the reverse of Scenario 2. Though rarely occurs, this could result in the inferred path being longer than the actual path if we enforce that no more than one flat link in each path in our inference algorithm.

Scenario 4: *Dual transit/peering relationship.* As mentioned in Section 3, two international ISPs may have dual transit/peering relationship, *i.e.,* they are customer and provider in one continent, but peers in another continent. In such cases, assuming they have one or the other relationship can create illegal paths (as mentioned above).

7.2 Routing Policies

We assume that ASes use the shortest path routing to choose among all legal policy paths. However, in practice, this may not always be true for the following reasons.

7.2.1 *Shortest Path Versus Customer Routes*

It is well known that ISPs often first prefer customer routes over peer routes, and then prefer peer routes over provider routes due to economic incentives. To study the effect of such a policy on the accuracy of our inference, we modify our algorithm in Figure 1 to take into account of this factor. However, this can only explain 1% of the paths inferred incorrectly. Therefore, this effect is insignificant.

7.2.2 *Inconsistent Advertisement to Different Peering Locations*

Example
Source AS = A, Destination AS = D
Inferred path = $ABCD$, Actual path = $ABEFD$
Definition: LCP = Longest Common subPath
between Inferred path and all AS paths to D
Case 1: LCP = **B**CD (First-hop AS B appears in Actual path)
Case 2: LCP = CD (No common AS with Actual path)

Peers may not consistently advertise the same length paths to all peering locations (e.g., due to inconsistent export policy). In this case, our inferred paths may be indeed legal, but not seen from the measurement location. To quantify this, we calculate the Longest Common subPath (LCP) between the inferred path and all the AS paths to the *same* destination AS. This represents how many mismatched inferred paths contain legal AS paths from BGP tables. We start from the beginning of the AS paths and eliminate one AS at a time until the remaining path exists in the BGP tables. We observe that 70% of the mismatches contain legal AS paths. Among them, 13% of the mismatched cases are identical to the actual AS paths except the first-hop AS, whereas in the remaining 57% of the mismatched cases, inferred paths do not share any AS on the actual BGP path. Note that the first case indicates that our inferred path is indeed *legal*.

We now analyze these two cases in more detail. In both cases, the inferred AS paths contain legal paths observed from BGP tables; thus it is less likely to violate actual AS relationships. Consider the example: actual AS path is $ABEFD$ and inferred BGP path is $ABCD$.

Case 1: There exists a path BCD in BGP tables. In such scenarios, the fact that $ABEFD$ is selected is likely due to inconsistent advertisement by AS B to AS A. AS B itself or part of AS B is using AS path BCD, as it is available in the BGP tables. However, AS B must have advertised to AS A path $BEFD$, leading AS A to choose the longer path. Note that AS B could not have advertised two paths to the same destination at the same location.

Case 2: There exists a path CD in BGP tables, but there does not exist any partial path of $ABCD$ that shares a common AS with the actual AS path. There are several possibilities why the longer path $ABEFD$ is chosen, assuming the inferred path is a legal path. (i) The same reason as in Case 1, where AS C was not consistent in advertising the routes to destination AS D. It did not advertise route CD to AS B, leading AS B to choose a longer AS path $BEFD$. (ii) AS B deliberately chooses a longer AS path – EFD over CD for traffic engineering purposes for example. (iii) AS C prepended the AS path CD (*e.g.,* to $CCCCD$) making it appear longer than the alternate path EFD. (iv) Transient routing changes or failures made the shorter path unavailable.

7.2.3 BGP Tie-breaking Rules

BGP sometimes uses tie-breaking rules to select among multiple available paths [2]. In a stub-AS, the tie-breaking rules are sometimes deterministic (e.g., based on the router ID), and sometimes non-deterministic (e.g., based on the oldest route). In large transit ASes, the tie-breaking rules also depends on the utilization of hot or cold potato routing. This complicated tie-breaking process makes it challenging to infer AS paths with high accuracy.

7.3 AS Prepending

AS prepending is a common practice where ISPs repeat its own AS number in its route advertisements sent to its neighbors, in the hope that these paths will be less preferred due to the increase in path length. It is questionable how effective this is, since one can still observe many paths containing prepending in any forwarding tables from large ISPs; if prepending were indeed effective, such paths would not be there. On the other hand, the presence of prepended paths in the forwarding table does not necessarily indicate the ineffectiveness of prepending. It could also be the case that the prepended path is the only available path, and it has to be chosen regardless. Since we do not have information of all the available paths, it is difficult to directly estimate how many longer paths are chosen because the alternate paths are prepended, appearing to be longer but in fact shorter.

We take the following approach to quantify the effect of AS prepending. We first identify which ASes have a tendency to prepend. We find 4,891 ASes are prepended at least once in the collection of BGP tables we examine. This constitutes more than 28% of total number of ASes, and we call such ASes *prepending ASes*. We then calculate for both mismatch and match cases the percentage of source-destination pairs whose inferred paths all contain at least one prepending AS. We place this requirement on all inferred paths because any single inferred path without prepending ASes could be chosen and eliminate the effect of prepending. We observe that 94% of the mismatched inferred paths contain prepending ASes, whereas 88% of the matched inferred paths contain such ASes. The higher likelihood for mismatched inferred paths to contain prepended AS suggests that it is likely that AS prepending accounts for some of the mismatches.

In the above analysis we choose a simple binary metric to evaluate how likely prepending affects path selection. In practice, prepending may occur on a per peer basis. We thus compute a *Score* based on the following definition to reflect how likely an AS A prepends itself based on the neighbor AS N and how frequently it is prepended across all AS paths. Let $p(A\_N)$ be the probability that AS A prepends itself on a path advertised to AS N computed using a large number of BGP table paths. The Score for a given path L is $\sum p(A_i\_N_i)/(|L|-1)$, where A_i is AS i on the path except the first AS, N_i is the previous AS of AS A_i on the path, or the AS receiving the route. $|L|$ is the length of the AS path. We find that mismatched inferred paths are more likely to contain prepending ASes than matched inferred paths (Score 0.89 versus Score 0.73). This score is an average across all paths. Based on the above results, it seems plausible that AS prepending is another contributing factor for the inaccuracy in AS path inference.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we explore the feasibility of inferring AS-level paths without direct access to either end-points. To this end, we develop RouteScope, a tool to infer AS paths by finding the shortest policy paths in an AS graph obtained from BGP tables at multiple vantage points. We also propose two enhancements: (i) a new AS relationship inference algorithm that achieves higher accuracy, and (ii) a novel technique for inferring the first AS hop by exploiting the TTL information contained in IP packet. Our results show that our enhanced tool can achieve up to 88% accuracy in AS path inference.

A number of future avenues remain. First, we would like to further improve the accuracy of path inference. Our study shows that there is a inherent limit on the accuracy of path inference at AS level. Therefore it is useful to explore the possibility of inferring paths at the prefix level. Second, our heuristic for inferring the first hop AS can be improved by leveraging the existing alias resolution techniques such as those used in Rocketfuel [18] and Mercator [5]. Finally, there are many interesting applications that can potentially benefit from the path inference, including network diagnosis, performance optimization and reliability enhancement in multihoming, content distribution, and peer-to-peer applications. We would like to explore these applications in depth.

Acknowledgments

We would like to thank Sharad Agarwal, Lixin Gao, and Maurizio Patrignani for providing AS relationship inference software, and thank Sharad Agarwal, Ratul Mahajan, Nick Feamster, the Oregon RouteViews Project, the RIPE Project, and MIT RON Project for providing the BGP data. Thanks to Jennifer Rexford, Aman Shaikh, and anonymous reviewers for valuable comments.

9. REFERENCES

[1] G. Battista, M. Patrignani, and M. Pizzonia. Computing the Types of the Relationships Between Autonomous Systems. In *Proc. IEEE INFOCOM*, March 2003.

[2] O. Bonaventure, P. Trimintzios, G. Pavlou, B. Quoitin, A. Azcorra, M. Bagnulo, P. Flegkas, A. Garcia-Martinez,

P. Georgatsos, L. Georgiadis, C. Jacquenet, L. Swinnen, S. Tandel, and S. Uhlig. Internet Traffic Engineering. In *Quality of Future Internet Services*, 2003.

[3] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang. Maximum Likelihood Network Topology Identification from Edge-based Unicast Measurements. In *Proc. ACM SIGMETRICS*, June 2002.

[4] L. Gao. On Inferring Automonous System Relationships in the Internet. In *IEEE/ACM Trans. Networking*, December 2001.

[5] R. Govindan and H. Tangmunarunkit. Heuristics for Internet Map Discovery. In *Proc. ACM SIGCOMM*, March 2000.

[6] G. Huston. BGP table statistics. `http://bgp.potaroo.net`.

[7] V. Jacobson. Traceroute. `ftp://ftp.ee.lbl.gov/traceroute.tar.gz`.

[8] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring Link Weights using End-to-End Measurements. In *Proc. Internet Measurement Workshop*, November 2002.

[9] Z. M. Mao, D. Johnson, J. Rexford, J. Wang, and R. Katz. Scalable and Accurate Identification of AS-Level Forwarding Paths. In *Proc. IEEE INFOCOM*, March 2004.

[10] Z. M. Mao, J. Rexford, J. Wang, and R. Katz. Towards an Accurate AS-Level Traceroute Tool. In *Proc. ACM SIGCOMM*, August 2003.

[11] P. R. McManus. A Passive System for Server Selection within Mirrored Resource Environments Using AS Path Length Heuristics. Apr. 1999. http://www.ducksong.com:81/patrick/proximate.pdf.

[12] T. Ogino, M. Kosaka, Y. Hariyama, K. Matsuda, and K. Sudo. Study of an Efficient Server Selection Method for Widely Distributed Web Server Networks. In *The 10th Annual Internet Society Conference (INET)*, July 2000. `http://www.isoc.org/inet2000/cdproceedings/1g/1g_1.htm`.

[13] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Trans. Networking*, pages 601–615, October 1997.

[14] V. Paxson. *Measurements and Analysis of End-to-End Internet Dynamics*. PhD thesis, U.C. Berkeley, 1997.

[15] PlanetLab. `http://www.planet-lab.org`.

[16] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP Routing Stability of Popular Destinations. In *Proc. Internet Measurement Workshop*, November 2002.

[17] B. Selman, H. Kautz, and B. Cohen. Local Search Strategies for Satisfiability Testing. In *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, 1993.

[18] N. Spring, R. Majajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proc. ACM SIGCOMM*, August 2002.

[19] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proc. IEEE INFOCOM*, June 2002.

[20] Traceroute.org. `http://www.traceroute.org/`.

[21] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level Topology. In *ACM SIGCOMM Computer Communication Review (CCR), special issue on Internet Vital Statistics*, January 2005.