On the Evaluation of AS Relationship Inferences

Jianhong Xia and Lixin Gao
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01002
{jxia, lgao}@ecs.umass.edu

Technical Report

***Abstract*— The commercial contractual relationship among Autonomous Systems (ASs) is important for understanding the reachability and traffic flow in the Internet. Several heuristic algorithms have been proposed for inferring AS relationships by extracting information from Border Gateway Protocol (BGP) routing tables. However, there is no evaluation of the AS relationships inferred from existing algorithms. In this paper, we present the methodology for obtaining partial AS relationships, and use them to evaluate these algorithms. Our result shows that although existing algorithms achieve good overall accuracy, the accuracy on the peer-to-peer relationships is quite poor. Furthermore, we propose an algorithm for inferring AS relationships by taking advantage of this partial information. We show that our proposed algorithm outperforms existing algorithms for inferring peer-to-peer relationships by $1.8 \sim 3.8$ times in accuracy. In addition, the overall accuracy of our algorithm is higher than that of existing algorithms.**

***Keywords*— Border Gateway Protocol, Routing Protocol, Routing Policy, Algorithm.**

I. Introduction

The Internet consists of thousands of distinct regions of administrative domains, each of which possesses one or several Autonomous Systems (ASs). These ASs exchange information about how to reach individual blocks of destination IP addresses via Border Gateway Protocol (BGP) [1], [2], [3]. BGP constructs AS paths by successively propagating updates between pairs of BGP routers that establish BGP *peering sessions* [3]. It allows each AS to choose its own policy on selecting the best routes, announcing and accepting routes.

One of the most important factors in determining routing policy is the commercial contractual relationship among administrative domains. As mentioned in [4], [5], the Internet topology alone does not imply reachability among ASs. For example, when AS *A* directly connects to AS *B*, it does not mean that *A* can directly reach the destination in *B*. The reason is that *A* might deploy a policy that prohibits the direct reachability from *A* to *B*. Hence the commercial contractual relationship between ASes is very important for us to understand the reachability and traffic flow in the current Internet. Although such information is important for Internet research, not all ISPs are willing to publish it. In fact, an ISP might keep it as a commercial secret.

Several algorithms [4], [5], [6] have been proposed for inferring AS relationships in the recent years. The accurate information on AS relationships will help us to understand the Internet hierarchy and routing policies in the AS-level topology [7], [5]. Although these algorithms have been proposed for several years, there is no evaluation of these algorithms in terms of the accuracy of AS relationships. In this paper, we present several techniques for obtaining partial AS relationships, and use this partial information to evaluate the AS relationships inferred from the existing algorithms in [4], [5]. From the AS relationship inferences on 07/10/2003, we find that although both algorithms in [4], [5] have good overall accuracy, the accuracy on the peer-to-peer relationships is quite poor. The algorithm in [4] achieves 49.08% accuracy on peer-to-peer relationships, while the algorithm in [5] achieves only 24.63% accuracy. Therefore, we propose a new algorithm for inferring AS relationships by taking advantage of the partial information. Our result shows that from only a small set of partial information, our algorithm achieves 96.37% overall accuracy, and 91.45% on peer-to-peer relationships, which is $1.8 \sim 3.8$ times as much as the accuracy of existing algorithms.

The remainder of this paper is structured as follows. Section II introduces the background materials and reviews some related work. Section III describes the methodology for obtaining partial AS relationships. Section IV evaluates and discusses the accuracy of AS relationship inferences of existing algorithms. We propose a new algorithm for inferring AS relationships in Section V. Finally, we summarize the paper and discuss some challenges in Section VII.

II. Background and Related Work

A. BGP Attributes

BGP is an interdomain routing protocol [8] that is used for exchanging routing information between ISPs. The information exchanged among BGP routers, called update message, carries one or more BGP attributes. Among these attributes, *BGP community* helps to achieve scalability and reduce management complexity in BGP configuration [9], [10]. An ISP can define the semantics on the usages of *BGP community* attribute for different routing policies. The usages of *BGP community* attribute has become more and more popular in the current Internet [11].

```
as-set:     AS2:AS-CUSTOMERS
members:    AS3 AS4
changed:    orange@ripe.net
source:     RIPE
```

Fig. 1. AS-SET Object in RPSL

B. Routing Policy and Internet Routing Registry

BGP allows for policy-based routing, which means that BGP router will propagate update messages to its BGP peering routers according to its own policy. For simplicity, we put BGP routing policy into two categories, import policy and export policy. Upon receiving an update message, a router must decide whether or not to accept this path according to its import policy. Once a router accepts paths from different neighbors, it must select one path among these candidates as the best one. Then, based on its export policy, the router decides whether or not to propagate this best path to its neighbors. In order to help maintaining the infrastructure of Internet routing and addressing, the Internet Routing Registries (IRR) [12] database is used to store the routing policies for each ISP. It uses Routing Policy Specification Language (RPSL) [13], [14] to define the common format and express a wide range of routing policies. As an example, Fig. 1 shows a usage of AS-SET object which defines AS2's customers in the IRR databases.

C. Related Work

The first study on inferring AS relationships is proposed by Gao in [4]. The paper summarizes the *valley-free* property of AS paths in BGP routing tables. Formally, an AS path is *valley-free* if and only if a provider-to-customer edge is followed by exclusively provider-to-customer or sibling-to-sibling edges, and a peer-to-peer edge is followed by exclusively provider-to-customer or sibling-to-sibling edges. The author proposes a heuristic algorithm for inferring AS relationships by extracting AS path information from public routing tables on Route Views routers [15]. Subramanian et al. [5] formulate AS relationship assignment as an optimization problem, *Type of Relationship (ToR)* problem. The authors present a heuristic algorithm for inferring the AS relationships by combining AS paths from multiple vantage points in the Internet. Battista et al. [6] analyze the complexity of *ToR* problem and present an approximately optimal solution to the *ToR* problem. However, it is not clear whether the optimal solution of ToR problem could certainly lead to an accurate inference of AS relationships in practice.

Although these algorithms have been proposed for several years, no study has been done on evaluating the performance of these algorithms. Given the facts on the incompleteness of AS-level topology from the limited sources [16], [17] and BGP misconfigurations [18] in the Internet, it becomes more and more necessary and interesting to evaluate the accuracy on the AS relationship inferences. Without any commercial agreements among ISPs, it is hard for us to evaluate the accuracy of these algorithms. Fortunately, the studies in [11], [19] suggest that some usages of *BGP community* indeed indicate the relationships among ASs. In this paper, we first introduce techniques for obtaining partial AS relationships and then use them to evaluate the inferences of existing algorithms. Furthermore, we propose a new algorithm for inferring AS relationships which outperforms existing algorithms.

TABLE I
GEOGRAPHIC LOCATION

ASN	Community Value	Meaning
6320	6320:21100	peer in LONDON
	6320:21200	peer in MANCHESTER
	6320:21220	peer in MaNAP
6705	6705:120	routes received from MCI(USA)
	6705:240	routes received from INXS(DE)
	6705:250	routes received from LINX(UK)

TABLE II
TYPES OF PEER

ASN	Community Value	Type of Peer
6320	6320:21xxx	Peer
	6320:22xxx	Transit
	6320:23xxx	Customer
	6320:24xxx	Local

III. METHODOLOGY FOR OBTAINING PARTIAL AS RELATIONSHIPS

One straight forward approach to evaluating the AS relationship inferences is to compare them with the commercial agreements among ISPs. It is infeasible to obtain all these commercial agreements in the Internet. However, we can at least obtain partial AS relationships of some ISPs from the usages of *BGP community* attribute and IRR databases. In this section, we introduce three different approaches to obtaining this information. One is from the usages of *BGP community* attribute, one from the instances of AS-SET object in the IRR databases, and another is from the explicit description of routing policies in the IRR databases.

A. From the Usages of BGP Community Attribute

In this subsection, we first review the common usages of *BGP community* attribute in practice, and then analyze the persistency on its usages in the Internet. Finally, we describe how to obtain partial AS relationships from the usages of *BGP community* attribute.

1) Usages of BGP Community Attribute: BGP community attribute consists of a set of four-octet values. Each community value is denoted as two parts *xxxxx:yyyyy*. From previous studies [11], [19], the community value often implies the type of peer, geographic location or interconnection point in practice. So the usages of *BGP community* attribute make it possible to indicate AS relationships. Table I and Table II show the examples of typical usages of *BGP community* attribute that can be obtained from RADB and RIPE databases in the IRR [12].

```
TIME: 07/02/03 11:04:07
TYPE: BGP4MP/MESSAGE/Update
FROM: 64.200.199.3 AS7911
TO: 128.223.60.102 AS6447
ORIGIN: IGP
ASPATH: 7911 3561 1239
NEXT_HOP: 64.200.199.3
COMMUNITY: 3561:21000 7911:999 7911:7302
ANNOUNCE
    141.238.0.0/16
```

Fig. 2. An Example of Update Message

TABLE III
PERSISTENCY ON THE USAGES OF BGP COMMUNITY ATTRIBUTE

Data Source	N_t	N_p
RIPE rrc00	5139	4593 (89.38%)
Route Views routers	31536	28256 (89.60%)
Overall	36675	32849 (89.57%)

Before we use *BGP community* to obtain AS relationships, we look at how persistent the usages of *BGP community* attribute are in the Internet and investigate whether this attribute is a good indication of the AS relationship.

2) Persistency on the Usages of BGP Community Attribute: To understand the persistency on the usages of *BGP community* attribute, we analyze whether the *BGP community* value is changed frequently between an AS and its neighbors.

We collect the BGP update messages from Route Views routers [15] and RIPE rrc00 [20] from 06/01/2003 to 07/10/2003. Fig. 2 illustrates the format of an update message translated by MRTd tools [21]. We parse all update messages in our data set and count the frequency of the usages of each community value. We define a tuple, {*A, B, Next_Hop*}, as a unit to maintain its community values between *AS A* and *AS B* that are observed from *Next_Hop*. For example, we can derive two tuples from the update message in Fig. 2. One is {*AS7911, AS3561, 64.200.199.3*} with community value *"7911:999 7911:7302"*, and the other is {*AS3561, AS1239, 64.200.199.3*} with community value *"3561:21000"*.

We use N_t to denote the total number of tuples that use one or more community values over the observation period, and N_p to denote the total number of tuples that use only one community value over the observation period. Table III shows the result of persistency on the usages of *BGP community* attribute. We see that almost 90% of tuples kept using one community value during the observation time.

From these observations, we argue that the usages of *BGP community* attribute are very persistent over a long period of time. It should be reliable to map AS relationships from the usages of *BGP community* attribute.

3) Mapping AS Relationships from BGP Community Values: Mapping AS relationships from *BGP community* values is not trivial because there is no standard syntax on the description of *BGP community* in IRR databases. We collect the semantics of *BGP community* attribute for each AS from the IRR databases, and put them in a simple database. In addition, we use additional public information [22], [23], [24], [25] to expand the knowledge of our database. Then, we parse the BGP update messages to get the usages of *BGP community* attribute. Finally, we query our database and map the *BGP community* value of AS pair into its relationship.

```
as-set:     AS4736:AS-CUSTOMERS
descr:      Provides Transit to these ASes
members:    AS4736, AS10023, AS17473, AS9382,
            AS9900, AS4820, AS10097, AS18221,
            AS4842, AS18375, AS18359, AS9482,
            AS18210
tech-c:     RM11-AP
notify:     routemaster@nttaus.com.au
mnt-by:     MAINT-AS4736
changed:    dave.matthews@nttaus.com.au 20030707
source:     RADB
```

Fig. 3. Example of AS-SET Object

B. From AS-SET Object in the IRR Databases

As we mentioned in Section II-B, many ISPs define their customers, providers and peers as different instances of AS-SET object in the IRR databases. Fig. 3 shows an example on the usage of AS-SET object in the RADB databases on 07/14/2003 for defining customers. We parse the IRR databases, and find out the instances which contain the strings of "customer", "provider" or "peer" in their AS-SET object. Then we manually check these usages and obtain partial AS relationships.

C. From Routing Policies in the IRR Databases

The description on routing policies in the IRR databases sometimes also imply the AS relationships. For example, the common practice for defining the routing policies could be that a provider exports all routes to its customers but only import those routes from its customers that are originated by them.

However, the routing policies for the ISPs in the Internet could be more complicated than we expect and there is no standard syntax to describe them. Similar to the method in AS-SET object, we extract AS relationships only when their relationships are explicitly described in their routing policies such as "Import from downstream transit customers", etc. As an example, Fig. 4 shows the routing policies of AS20679 which explicitly describe its providers, customers and peers. Thus, we can get the information that AS20679 is a provider of AS13256.

D. Results of Partial AS Relationships

Based on the techniques we have presented in this section, we process the data on the RADB and RIPE's IRR databases published on 07/14/2003 and BGP updates on Route Views routers [15] and RIPE rrc00 [20] from 06/01/2003 to 07/10/2003. To avoid using the obsolete information, we only analyze the records that were updated in 2003. We obtain the relationships for 4914 distinct AS pairs (2419 AS pairs from the usages of BGP Community, and 2916 AS pairs from the usages of AS-SET object and routing policies). 28 of them do not have a persistent AS relationship when we combine

```
aut-num:       AS20679
as-name:       HSO
descr:         HighSpeed Office Limited
descr:         Broadband internet data and voice services provider.
remarks:       +-------------------------------------------------+
remarks:       |      Import from upstream transit providers     |
remarks:       +-------------------------------------------------+
import:        from AS6461  action pref=50;  accept ANY
import:        from AS3356  action pref=100; accept ANY
remarks:       +-------------------------------------------------+
remarks:       |     Import from downstream transit customers    |
remarks:       +-------------------------------------------------+
import:        from AS13256 action pref=100; accept AS13256
remarks:       +-------------------------------------------------+
remarks:       |              Import from LINX peers             |
remarks:       +-------------------------------------------------+
import:        from AS1901  action pref=100; accept AS-EUNETAT
import:        from AS2110  action pref=100; accept AS-IEUNET
... ... ... ...  ... ... ... ...  ... ... ... ...  ... ... ... ...
```

Fig. 4. Example of Routing Policy for AS20679

the results from these three approaches. We ignore these few special cases from our study. Among the remaining 4886 AS pairs, 3717 of them are provider-customer relationships and 1169 of them are peer-to-peer relationships. Since our partial information covers 3616 different ASs, we believe that it is a good sample to evaluate the AS relationship inferences from the existing algorithms.

IV. Evaluation on the AS Relationship Inferences

In this section, we use the partial AS relationships to evaluate the inference of the existing algorithms. We refer to the algorithm in [4] proposed by Gao as LG and the algorithm in [5] proposed by Subramanian et al. as SARK. We run the LG algorithm to obtain the inferences on the routing table from Route Views routers. The inferences of SARK algorithm are available online at [26]. We compare our partial information with the inferences of LG and SARK algorithms from the data on 07/10/2003.

A. Accuracy on the Type of Relationships

We calculate the accuracy based on the overlapping AS pairs appearing in both our partial AS relationships and the inferences of each algorithm. There are 2802 overlapping AS pairs in the inferences of the LG algorithm and 2821 overlapping AS pairs in the inferences of the SARK algorithm. We use these overlapping AS pairs as the samples to evaluate the accuracy of these two algorithms.

Table IV shows the accuracy of inferences of the LG and SARK algorithms. We see that the accuracy of the LG algorithm is higher than that of the SARK algorithm on both provider-customer and peer-to-peer relationship inferences. Although the accuracy of both algorithms on provider-customer inferences is high, the accuracy on peer-to-peer inferences is very poor. On peer-to-peer relationship inferences, the LG algorithm achieves 49.08% accuracy and the SARK algorithm achieves only 24.63% accuracy. The overall accuracy of the LG algorithm is 3% higher than that of the SARK algorithm.

B. Accuracy on the Degree of AS Pairs

In order to further investigate what AS pairs have incorrect inferences from the existing algorithms, we look at the distribution of the accuracy on the degree of AS pairs.

We define $\alpha(d)$ to be the accuracy of the inferences on the AS pairs in which both degrees are greater than or equal to d. From the definition, we know that $\alpha(1)$ denotes the overall accuracy of the inferences on all AS pairs.

Fig. 5(a) shows the relation between the overall accuracy of the inferences and the degree of AS pairs for both of the LG and SARK algorithms. Although the overall accuracy of the LG and SARK algorithms is high, the accuracy for the AS pairs with medium or large degrees is not good. For example, $\alpha(100)$ is 67% for LG algorithm and only 58% for SARK algorithm. If we only look at the accuracy on peer-to-peer relationships in Fig. 5(b), we find that neither of them can achieve a good accuracy for the AS pairs with small degrees.

C. Discussion on the Evaluation

From our evaluation results, we observe two limitations of the LG and SARK algorithms on the accuracy of the AS relationship inferences.

The first limitation of the LG and SARK algorithms is that both of them have a low accuracy on peer-to-peer relationship inferences. One interesting observation is that both algorithms achieve better accuracy on inferring provider-customer relationships than on inferring peer-to-peer relationships. The reason partly comes from the power-law distribution of AS degrees in the Internet topology [27]. More than two-thirds of ASs has only one or two degrees located at the edge of the Internet, most of which are the customers of other ASs. Therefore, in the graph of AS level topology, we can easily infer the edge ASs (those ASs with one or two degrees) to be the customers of their neighbor ASs. On the other hand, it is not obvious to determine the relationships among the ASs which are other than the edge ASs.

TABLE IV

ACCURACY OF AS RELATIONSHIP INFERENCES ON 07/10/2003

Inferring Algorithm	LG algorithm	SARK algorithm	PTE algorithm	
			Evaluated by 100% partial information	Evaluated by 80% partial information
Samples of AS pairs	2802	2821	2818	2254
Accuracy of Provider-Customer Relationships	99.13%	98.35%	96.92%	96.14%
Accuracy of Peer-to-Peer Relationships	**49.08%**	**24.63%**	**91.45%**	**89.33%**
Overall accuracy	94.25%	91.24%	96.37%	95.46%

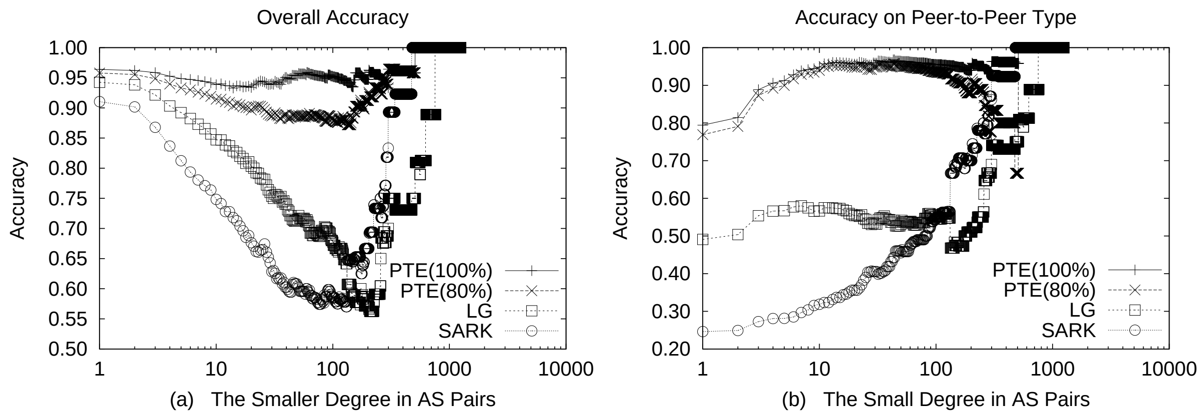


Fig. 5. Accuracy on the Degree of AS Pairs

"PTE (100%)" denotes the accuracy evaluated by 100% partial information. "PTE (80%)" denotes the accuracy evaluated by 80% partial information.

The second limitation is that neither the LG algorithm nor the SARK algorithm can achieve a consistent performance on the accuracy over the different degrees of AS pairs. Both algorithms have similar accuracy distribution on the degree of AS pairs shown in Fig. 5. For the large ISPs in the core of the Internet, both algorithms can correctly infer them as peer-to-peer relationships. So the accuracy for large degree AS pairs is high. For the small ISPs in the edge of the Internet, although both algorithms achieve good performance on overall accuracy, they are not consistent on the degree of AS pairs. Furthermore, the accuracy on peer-to-peer relationships is poor.

The LG and SARK algorithms do not perform well on inferring relationships for all AS pairs. One of possible reasons is that some AS paths in the routing table do not conform to the *valley-free* property because of BGP misconfigurations [18] or special routing policies. With the rapid growth of the Internet in the recent years, many ISPs interconnect at multiple locations for traffic engineering. From an operational viewpoint, it is not necessary for all these interconnecting points to follow the same routing policies. Thus the relationships could be more complicated than we expect. Furthermore, there are more and more ASs participating into the connection to the exchange points [28]. The policies between exchange points and their members might not conform to either provider-customer or peer-to-peer relationships. In those special cases when an AS path contains the AS of exchange point, it might not conform to *valley-free* property.

Without any prior information, existing algorithms cannot differentiate the *valley-free* and *non-valley-free* paths before they start inferring. Although the chances of *non-valley-free* AS paths are small, they do affect the accuracy of inferences in the existing algorithms. Motivated by these reasons, we propose a new algorithm in next section so that we can make use of our partial information to filter these *non-valley-free* AS paths.

V. INFERRING AS RELATIONSHIPS FROM PARTIAL INFORMATION

Based on the discussion in section IV-C, we now propose a new algorithm for inferring AS relationships. The basic idea of our algorithm is to infer the entire AS relationships from partial information. We refer to our algorithm as PTE (Partialness To Entireness).

A. Algorithm for Inferring AS Relationships

Our algorithm consists of two major components. One is to filter *non-valley-free* paths, and the other is to infer AS relationships from partial information. We also have several techniques in our algorithm for further improvement.

1) Filtering Non-Valley-Free Paths: Using our partial AS relationships, we examine *valley-free* property for each AS path. According to the *valley-free* property, in a *valley-free* path, a peer-to-peer edge cannot follow a peer-to-peer or provider-to-customer edge, and customer-to-provider edge cannot follow a peer-to-peer or provider-to-customer edge.

Although we do not have entire AS relationships, only based on our partial AS relationships, we still can identify some AS paths which are apparently not *valley-free*. For example

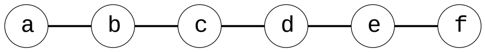


Fig. 6. An Example of AS Path

in Fig. 6, we do not know the AS relationships between (b, c), (c, d) and (e, f). However, if we know that (a, b) is a provider-to-customer edge, and (d, e) is a customer-to-provider edge, then we can claim that this path cannot be *valley-free*. Those *non-valley-free* paths are most likely caused by BGP misconfigurations or special routing policies. We remove these paths from our data before we infer the AS relationships.

2) Inferring AS Relationships: Now we present techniques on inferring the AS relationships after we have filtered *non-valley-free* paths.

We introduce a graph $G = (V, E)$ to model the connectivity between ASes, where the node set V consists of ASes and the edge set E consists of AS pairs. We denote the edge between node u and node v as (u, v). We have obtained partial AS relationships among those AS pairs in Section III-D. Our goal here is to infer the relationships for the rest of AS pairs.

Based on *valley-free* property of AS paths, we define three inference rules and one refreshing rule as follows:

- *Inference Rule 1:* In an AS path, provider-to-customer edge is followed by provider-to-customer edge. For example in Fig. 6, if the edge (c, d) is a provider-to-customer edge, we mark (d, e) and (e, f) as provider-to-customer edges.
- *Inference Rule 2:* In an AS path, customer-to-provider edge only follows customer-to-provider edges. For example in Fig. 6, if the edge (c, d) is a customer-to-provider edge, we mark (b, a) and (c, b) into provider-to-customer edges.
- *Inference Rule 3:* In an AS path, the left part of peer-to-peer edge is customer-to-provider edge, and the right part of peer-to-peer edge is provider-to-customer edge. For example in Fig. 6, if the edge (c, d) is a peer-to-peer edge, we mark (b, a), (c, b), (d, e) and (e, f) as provider-to-customer edges.
- *Refreshing Rule:* If both (u, v) and (v, u) are inferred to provider-to-customer edge, we mark (u, v) and (v, u) as sibling-to-sibling edge.

Note that if we start with our partial information and apply the above rules repeatedly, we can get more AS relationships identified. Once we cannot infer any AS relationships, we use the algorithm in [4] to infer the remaining AS pairs into provider-customer, peering-to-peering or sibling-to-sibling relationships.

3) Several Techniques for Improvement: We have several techniques in our algorithm for further improvement. For example, we collect multiple routing tables to augment the AS-level connectivity and to we can get more AS paths, then we detect BGP transient misconfigurations and filter the AS paths that caused by them. In addition, we have a heuristic algorithm to eliminate all relationship cycles [4], [5], [6] in our inferences.

- *Detecting BGP Transient Misconfiguration:*
  Misconfigurations on a BGP router could introduce a route or AS path that should not exist in BGP routing tables. We refer to those short-lived configuration errors in BGP routers as BGP transient misconfigurations. In order to reduce the impact of this kind of misconfigurations on our inference results, we need to detect them and filter those AS paths from BGP routing tables.
  To detect the AS paths caused by BGP transient misconfiguration, we examine the presence of AS path in multiple routing tables which are collected from the same BGP router during a long period of time. We assume that most of the transient misconfigurations are short-lived in the Internet. So, those AS paths caused by BGP transient misconfigurations should not appear in most of the snapshots. Previous study [18] shows that 80% of misconfigurations last for less than 2 hours. If we collect routing tables from a router each hour for one day, we could observe that most AS paths caused by BGP transient misconfigurations can only appear in two of these routing tables.
  Based on this assumption, we present a simple algorithm to filter those AS paths caused by BGP transient misconfigurations. We use the majority rule to filter AS paths which do not appear often in routing tables. Formally, we define N as the total number of routing tables collected from one BGP router over the time, $P(r)$ as the number of routing tables containing AS path r, and $freq(r)$ as the frequency of AS path r present in these routing tables. So we have
  $$freq(r) = \frac{P(r)}{N} \tag{1}$$
  We define a threshold $F_{threshold} \epsilon (0, 1)$ to determine whether AS path r is caused by BGP transient misconfigurations. The majority rule is described as follows, if $freq(r)$ is less than $F_{threshold}$, we filter r from our data.
- *Eliminating the Cycles in the Inferences:*
  We have noticed that some cycles could exist in the inference result of AS relationships [4], [5]. For example, a cycle of (A, B, C) could be that, AS *A* is a provider of AS *B*, AS *B* is a provider of AS *C*, and AS *C* is a provider of AS *A*. This kind of cycles in AS relationships hardly exist in the current Internet. In fact, it is undesirable and contradictory to the purpose of the inferring procedure.
  In our algorithm, we introduce a heuristic to break up the cycles. The intuition behind our heuristic is that the degree of a customer is generally smaller than the degree of its provider. Therefore the most unlikely customer-to-provider edge in a cycle is the one whose customer's degree is much larger than the provider's degree. Our heuristic identifies the edge which is most unlikely to have a customer-to-provider relationship, and changes its relationship to provider-to-customer.

TABLE V

SUMMARY OF ROUTING TABLES ON 07/10/2003

Data Source	Nodes	AS Pairs	AS Paths
RV(single)	15670	34863	948108
RV(multiple)	15717	35686	1015888
RIPE(single)	15582	29088	223924
RIPE(multiple)	15608	29325	235789
Overall	15726	36246	1251677

We denote the ratio of AS degrees of a customer-to-provider edge (u, v) as $ratio(u, v)$ in Equation 2, where u is a customer of v.

$$ratio(u, v) = \frac{degree(u)}{degree(v)} \quad (2)$$

There are two steps to eliminate a cycle in our heuristic. The first step is to find AS set which contain cycles. To do this, we recursively remove all ASes which do not have providers or customers. In the set of the remaining ASes, each AS has both provider and customer. It is easy to show that if the set is not empty, there is at least one cycle in it. The second step is to find the AS pairs with the largest $ratio(u, v)$ value and break up the cycle by assigning u as a provider of v. We repeat these two steps to eliminate all cycles in the inference result of AS relationships.

B. Data Sources and Inference Result

1) Data Sources: We use routing tables from the Route Views routers [15] and RIPE rrc00 [20]. Route Views routers archive routing tables every 2 hours and RIPE rrc00 archives routing tables every 8 hours. In our experiment, we download the archived routing tables from Route Views routers and RIPE rrc00 on 07/10/2003, so we have totally 15 routing tables. In order to get a more complete view of BGP routing tables, we combine the routing tables from Route Views routers and RIPE rrc00 together.

We show the summary of routing tables in Table V. "RV(single)" denotes the summary of one routing table from Route Views routers, while "RV(multiple)" denotes the summary of combined 12 routing tables from Route Views routers. Similarly, "RIPE(single)" denotes the summary of one routing table from RIPE rrc00, while "RIPE(multiple)" denotes the summary of combined 3 routing tables from RIPE rrc00. "Overall" denotes the summary of combined all 15 routing tables from Route Views routers and RIPE rrc00.

Table V clearly suggests that we can get a more complete data set in terms of the number of AS pairs and AS paths from multiple snapshots of routing tables.

2) Filtering the Misconfigurations and Non-Valley-Free Paths: We apply the majority rule to filter out AS paths that do not appear often in routing tables. In our experiments, we choose $F_{threshold}$ as 0.4, which means that if an AS path does not appear in 40% of total number of routing tables collected from a BGP router, we filter it from our data. For a path which is not *valley-free* according to our partial information, we also filter it from our data. In practice, we could have more strict rules to filter possible misconfigurations based on well-known information. For example, we might assert that some largest ISPs, such as UUnet and AT&T, could not be the customers of any small ISPs. In our study, we do not take these well-known information. Instead, we only use the methods proposed in Section V-A to filter misconfigurations.

TABLE VI

SUMMARY ON AS PATHS FILTERING

	Nodes	AS pairs	AS paths
Before	15726	36246	1251677
After	15694	35583	1175684
Difference	0.2%	1.83%	6.07%

TABLE VII

INFERENCES ON THE AS RELATIONSHIPS

Relationship	# of AS pairs	Percentage
Provider-Customer	30355	85.31%
Peer-Peer	5116	14.38%
Sibling-Sibling	112	0.31%
Total	35583	100%

Table VI shows the summary of routing tables before and after AS paths filtering. We see that about 6.07% of AS paths have been filtered out from the original data. Only 0.03% of AS paths have been filtered because of *non-valley-free* property.

3) Inference Result: We use partial AS relationships obtained in Section III-D and our PTE algorithm to infer the AS relationships on the above data. Among 4886 partial AS relationships, only 2819 AS pairs exist in our data set. Thus, in our experiment, we use these 2819 partial AS relationships as our initial information to infer the relationships on the entire collection of 35583 AS pairs. The result is shown in Table VII. We find that 85.31% of AS pairs have provider-customer relationships, while 14.38% have peer-to-peer relationships, and only 0.31% have sibling-to-sibling relationships.

VI. EVALUATION ON OUR ALGORITHM

In this section, we first compare our PTE algorithm with the LG and SARK algorithms in terms of accuracy on the type of relationships and accuracy on the degree of AS pairs. Then we analyze the impact of partial information on our PTE algorithm. At last, we show the consistency of the inferences from our PTE algorithm.

A. Comparison with LG and SARK algorithms

In order to evaluate our PTE algorithm, we randomly split the partial AS relationships into two subsets of 20% and 80%. We take the subset of 20% partial AS relationships as initial information for the PTE algorithm and use the subset of 80% partial information or 100% partial information to evaluate the PTE algorithm.

1) Accuracy on the Type of Relationships: We run PTE algorithm for 10 times with different random seeds. We present the average accuracy of inferences in Table IV. It shows that our algorithm achieves not only high accuracy on inferring provider-to-customer relationships, but also around 90% accuracy on inferring peer-to-peer relationships. Compared to the LG and SARK algorithms, our PTE algorithm improve the accuracy of peer-to-peer relationships significantly, which is $1.8 \sim 3.8$ times as much as that of existing algorithms. On the overall accuracy, our PTE algorithm still outperforms the LG and SARK algorithms.

2) Accuracy on the Degree of AS Pairs: We also evaluate our PTE algorithm on the relation between the accuracy and the degree of AS pairs. We calculate the accuracy on the degree of AS pairs from one trial of our experiment results. The result in Fig. 5 shows that, no matter using 100% or 80% partial information to evaluate the PTE algorithm, the accuracy of PTE algorithm is consistently high on the different degree ranges of AS pairs.

From these experiments, we show that from only a small set of partial information, our PTE algorithm achieves high accuracy on both provider-customer and peer-to-peer relationship inferences. It also has a consistently high accuracy of the inferences on the degree of AS pairs.

B. Impact of the Partial Information on the PTE Algorithm

To examine the impact of the composition of partial information on the accuracy of the PTE algorithm, we use different subsets of partial AS relationships as initial information to run PTE algorithm. Especially, we look at how the different sizes and locations of partial information as well as the different percentages of peer-to-peer relationships in initial information affect the accuracy of PTE algorithm. In this subsection, we use the 100% partial information to evaluate our algorithm.

In general, an AS which is closer to the core in the Internet has a larger degree, and an AS which is closer to the edge has a smaller degree. For an AS pair, we use the sum of their degrees to denote its location in the Internet. The larger the sum of the degrees of an AS pair, the closer it is to the core in the Internet.

1) Impact of the Size and Location of the Partial Information: In order to examine the impact of the size of partial information on the accuracy, we vary the size of initial information from 10% to 40% of partial AS relationships for PTE algorithm, and use the corresponding remaining partial AS relationships to evaluate the accuracy. For each set of different sizes, we have two ways on selecting the initial information. One is to select the AS pairs close to the core, the other is to select the AS pairs close to the edge.

Fig. 7(a) shows the overall accuracy for each case. Fig. 7(a) suggests that the location of partial information affect the inference accuracy more significantly than the size of partial information. For the cases of same size, the initial information closer to the core induces better accuracy than that closer to the edge. On the other hand, for the cases of the same location, when the size of partial information is changed from 20% to 40%, the accuracy does not increase much. When the initial information is close to the edge and the size is changed from 10% to 20%, the accuracy increases from 82% to 87%. It suggests that a large size of partial information may improve the accuracy initially, but once the partial information is more than enough, it does not further improve the accuracy. Therefore the above analysis suggests that obtaining more information close to the core in the Internet will improve the accuracy of the PTE algorithm.

2) Impact of the Percentage of Peer-to-Peer Relationships in the Partial Information: In this experiment, we examine how the proportion of peer-to-peer relationships in the initial information for PTE algorithm affects the accuracy. As the same as before, we select 20% of partial AS relationships as initial information for PTE algorithm and use the corresponding remaining partial AS relationships to evaluate the accuracy. When we select initial information, we vary the percentage of peer-to-peer relationships in the initial information from 2% to 18%. For each percentage, we repeat 10 times with different random seeds and calculate the average accuracy.

The accuracy of PTE on different percentage of peer-to-peer relationships in the initial information is shown in Fig. 7(b). It shows that the PTE algorithm improves the accuracy when the percentage of peer-to-peer relationships in the initial information increases. But when the percentage of peer-to-peer relationships reaches a certain point, e.g, 10% in our result, the accuracy will saturate.

C. The Consistency of the Inference Results Over the Time

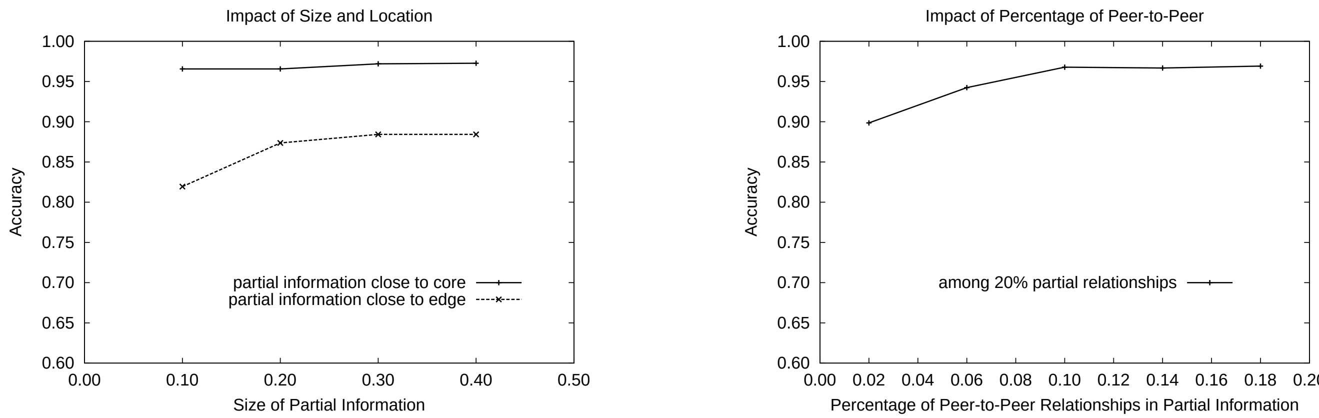
In this subsection, we look at the consistency of the inference results of existing algorithms over time. The hypothesis is that the commercial agreements among ISPs might be changed at some time, but they should not be changed frequently in one week or a shorter period of time. Thus the relationships among most ASes should remain the same in a observation window of one week long. Based on this assumption, we expect the relationships inferred from the existing algorithms to be consistent.

We download routing tables from Route Views routers and RIPE rrc00 from 07/06/2003 to 07/12/2003 and run different existing algorithms on daily data in that week. We compare the inference results of every day and evaluate the consistency of results over one week. In this study, we ignore the AS pairs that do not appear in the daily routing tables. For example, there are N AS pairs appearing every day, and only M AS pairs having the same inference results. We use M/N to denote the consistency of AS relationship inferences.

From the inferences of our PTE algorithm, the consistency over that week is 97.96%. It suggests that inference results of the PTE are relatively consistent over that week.

D. Consistency on the Percentage of Valley-Free AS Paths

Ideally, all AS paths in the routing tables should be *valley-free*. However, the complicated commercial agreements among ASes and misconfigurations in BGP routers could induce non *valley-free* AS paths in the routing tables. We use the inference results from PTE on 07/10/2003 as the reference and evaluate the percentage of *valley-free* in the routing tables from 07/06/2003 to 07/12/2003.



(a) (b)

Fig. 7. Impact of Partial Information on the Inference Result of the PTE Algorithm

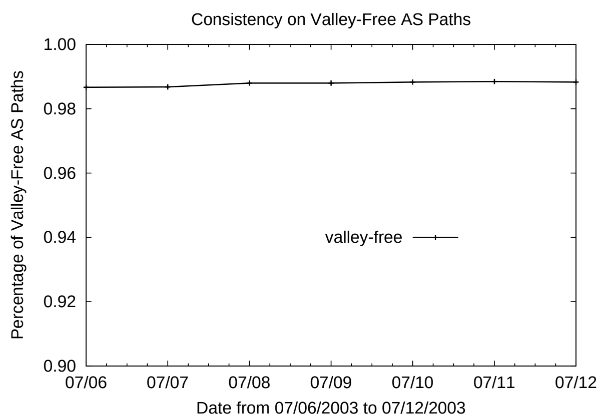


Fig. 8. Percentage of Valley-Free AS Paths

Fig. 8 shows the percentage of *valley-free* paths in that week. We see that more than 98.6% of AS paths are *valley-free* on every day. The percentages are stable and consistent over the time.

VII. Summary and Discussion

In this paper, we first introduce techniques on obtaining partial AS relationships, and then use this partial information to evaluate the accuracy of the existing algorithms. Then we propose a new algorithm on inferring AS relationships from partial information. The result shows that from only a small set of partial information, our algorithm is able to achieve better performance than existing algorithms.

With the rapid growth of the Internet in the recent years, inferring AS relationships becomes hard. First, the relationship of two ISPs might not be consistent at different interconnect points [28]. For example, they might have peer-to-peer relationship in North American, but have provider-customer relationship in Europe. Second, without a standard syntax on describing the usage of BGP community and routing policies in IRR databases, it is hard to systematically abstract partial AS relationships from them.

Acknowledgment

This work is supported in part by NSF under Grant No. ANI-0208116 and Alfred Sloan Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation and Alfred Sloan Foundation.

References

[1] Bassam Halabi. *Internet Routing Architectures*. Cisco Press, 1997.
[2] John W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, 1999.
[3] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). Request for Comments 1771, March 1995.
[4] L. Gao. On Inferring Autonomous System Relationships in the Internet. In *IEEE/ACM Trans. Networking*, December 2001.
[5] L. Subramanian, S. Agarwal, J. Rexford, , and R. H. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. In *Proc. IEEE INFOCOM*, June 2002.
[6] G. Di Battista, M. Patrignani, and M. Pizzonia. Computing the Types of the Relationships between Autonomous Systems. In *Proc. IEEE INFOCOM*, April 2003.
[7] Z. Ge, D.R. Figueiredo, S. Jaiwal, and L.Gao. On the Hierarchical Structure of the Logical Internet Graph. ITCOM' 2001.
[8] http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/bgp.htm.
[9] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. Request for Comments 1997, August 1996.
[10] E. Chen and T. Bates. An Application of the BGP Community Attribute in Multi-home Routing. Request for Comments 1998, August 1996.
[11] B. Quoitin and O. Bonaventure. A Survey of the Utilization of the BGP Community Attribute, 2002.
[12] Internet Routing Registry. http://www.irr.net/docs/list.html.
[13] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. Routing Policy Specification Language (RPSL). Request for Comments 2622, June 1999.
[14] D. Meyer, J. Schmitz, C. Orange, M. Prior, and C. Alaettinoglu. Using RPSL in Practice. Request for Comments 2650, August 1999.
[15] Route Views Project. http://www.antc.uoregon.edu/route-views/.
[16] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. The Origin of Power Laws in Internet Topologies Revisited. In *Proc. IEEE INFOCOM*, June 2002.
[17] J. Gast and P. Barford. Resource Deployment Based on Autonomous System Clustering. In *Proc. IEEE GLOBE INTERNET*, November 2002.

[18] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfiguration. In *Proc. ACM SIGCOMM*, August 2002.
[19] F. Wang and L. Gao. On Inferring and Characterizing Internet Routing Policies. In *Proc. ACM Sigcomm Internet Measurement Conference*, October 2003.
[20] RIPE Routing Information Services. http://www.ripe.net/ris/.
[21] Multi-Threaded Routing Toolkit. http://www.merit.edu/∼mrt/.
[22] Communities Used by EUnet. http://www.jippii.net/communities.shtml.
[23] http://infopage.cary.cw.net/Routing_Registry/community_announce.htm.
[24] http://cw-rr.cw.net/community_prepend.htm.
[25] https://robin.gblx.net/api/docs/customer-communities.html.
[26] http://www.cs.berkeley.edu/∼sagarwal/research/BGP-hierarchy/.
[27] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proc. ACM SIGCOMM*, August 1999.
[28] Exchange Point Information. http://www.ep.net/ep-main.html.