On Inferring Autonomous System Relationships in the Internet

Lixin Gao

Abstract

The Internet consists of rapidly increasing number of hosts interconnected by constantly evolving networks of links and routers. Interdomain routing in the Internet is coordinated by the Border Gateway Protocol (BGP). BGP allows each autonomous system (AS) to choose its own administrative policy in selecting routes and propagating reachability information to others. These routing policies are constrained by the contractual commercial agreements between administrative domains. For example, an AS sets its policy so that it does not provide transit services between its providers. Such policies imply that AS relationships are an important aspect of Internet structure. We propose an augmented AS graph representation that classifies AS relationships into customer-provider, peering, and sibling relationships. We classify the types of routes that can appear in BGP routing tables based on the relationships between the ASes in the path and present heuristic algorithms that infer AS relationships from BGP routing tables. The algorithms are tested on publicly available BGP routing tables. We verify our inference results with AT&T internal information on its relationship with neighboring ASes. As much as 99.1% of our inference results are confirmed by the AT&T internal information. We also verify our inferred sibling relationships with the information acquired from the WHOIS lookup service [1]. More than half of our inferred sibling-to-sibling relationships are confirmed by the WHOIS lookup service. To the best of our knowledge, there has been no publicly available information about AS relationships and this is the first attempt in understanding and inferring AS relationships in the Internet. We show evidence that some routing table entries stem from router misconfigurations.

1 Introduction

The Internet has experienced a tremendous growth in its size and complexity since its commercialization. The Internet connects thousands of Autonomous Systems (ASes) operated by many different administrative domains such as Internet Service Providers (ISPs), companies and universities. Since two ISPs might merge into one and each administrative domain can possess several ASes, an administrative domain can operate one or several ASes. Routing within an AS is controlled by intradomain routing protocols such as static routing, OSPF, IS-IS, and RIP. A pair of ASes interconnect via dedicated links and/or public network access points, and routing between ASes is determined by the interdomain routing protocol such as Border Gateway Protocol (BGP). One key distinct feature of the interdomain routing protocol is that it allows each AS to choose its own administrative policy in selecting the best route, and announcing and accepting routes. One of the most important factors in determining routing policies is the commercial contractual relationships between administrative domains.

The commercial agreements between pairs of administrative domains can be classified into customer-provider, peering, mutual-transit, and mutual-backup agreements [2, 3]. A customer pays its provider for connectivity to the rest of the Internet. Therefore, a provider does transit traffic for its customers. However, a customer does not transit traffic between two of its providers. A pair of peers agree to exchange traffic between their respective customers free of charge. A mutual-transit agreement allows a pair of administrative domains to provide connectivity to the rest of the Internet for each other. This mutual-transit relationship is typically between two administrative domains such as small ISPs who are located close to each other and who can not afford additional Internet services for better connectivity. A pair of administrative domains may also provide backup connectivity to the Internet for each other in the event that one administrative domain's connection to its provider fails.

These contractual commercial agreements between administrative domains play a crucial role in shaping the structure of the Internet and the end-to-end performance characteristics. Previous work on the Internet topology has been focused on the *interconnection* structure at either AS or router level [4, 5, 6, 7, 8, 9]. Since routing between ASes is controlled by BGP – a policy-based routing protocol, connectivity does not imply reachability. For example, national ISPs A and B are connected to their customer – a regional ISP C respectively. Although ISPs A and B are connected through ISP C, ISP A can not reach ISP B via ISP C since C as a customer does not provide transit services between its providers. Even if ISPs A and B

L. Gao is with the Dept. of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01002. lgao@ecs.umass.edu. The author was supported in part by NSF grant ANI-9977555, NSF grant ANI-0085848, and NSF CAREER Award grant ANI-9875513. 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.

can reach each other via other ISPs, the end-to-end performance characteristics between A and B can not be inferred from that of between A and C and between C and B. For example, the delay between A and B is independent of the total delay between A and C and between C and B. This has been observed by several measurement studies [10, 11]. Therefore, a global picture of AS relationships is an important aspect of the Internet structure.

We propose an augmented AS graph representation to capture AS relationships. We classify the relationship between a pair of interconnected ASes into customer-provider, peering, and sibling relationships. There is no publicly available information about inter-AS relationships. ISPs do not register their relationships to the Internet registries. Internet registries such as ARIN [1] do provide information such as who administrates an AS. However, the information can be out of date and does not imply anything about how ASes relate to each other. Contractual agreements between ISPs are proprietary and companies are unwilling to reveal even the names of their ISPs [12]. Internet Routing Registries (IRR) was created as a repository of routing policies. However, some ISPs are not willing to reveal their policies and even if they are, these routing policies might not specify AS relationships.

In this paper, we present heuristic algorithms that infer the augmented AS graph from BGP routing tables. We first formally present the routing policies implied by AS relationships and derive routing table entry patterns as the result of routing policies. We then infer the AS relationships based on the heuristic that the size of an AS is typically proportional to its degree in the AS graph. This heuristic has been used by Govindan and Reddy [5] in classifying ASes into four levels of hierarchy. Our heuristic algorithms classify an interconnected AS pair into having a provider-customer, peering, or sibling relationship. The running time of the algorithm is linear in the total number of consecutive AS pairs in the routing tables. To the best of our knowledge, this is the first attempt in understanding and inferring AS relationships in the Internet.

Furthermore, we perform an experimental study of AS relationships in the Internet. BGP routing tables are retrieved from the Route Views server in Oregon [16], which is publicly available and has the most complete view currently available. The Route Views server establishes BGP peering sessions with many tier-1 and tier-2 ISPs. Among the connected AS pairs, the algorithms infer that more than 90.5% of the AS pairs have customer-provider relationships, less than 1.5% of the AS pairs have sibling relationships, and less than 8% of the AS pairs have peering relationships. We verify our inference results with AT&T internal information on its relationship with neighboring ASes. Our result shows that 100% of our inferred customers are confirmed by the AT&T internal information. 100% of our inferred peers are confirmed by the AT&T internal information. 20% of our inferred siblings are confirmed by the AT&T internal information. Out of all of our inference results, 98.9% of inference results are confirmed by the AT&T internal information. We also verify our inferred sibling relationships with the information acquired from the WHOIS lookup service [1]. More than half of the inferred sibling relationships are confirmed by the WHOIS lookup service. We show evidence that some BGP routing table entries stem from router misconfigurations.

The remainder of the paper is structured as follows. Section 2 presents an overview of interdomain routing and discusses previous work on the Internet topology and routing. In Section 3, we define the types of relationships between ASes and implied export policies. We also derive routing table entry patterns resulted from the export policies. Section 4 presents heuristic algorithms for inferring AS relationships. In Section 5, we perform an empirical study of inferring AS relationships by using publicly available BGP routing tables. We conclude the paper in Section 6 with a summary and future work.

2 Background on Interdomain Routing and Related Work

In this section, we present background material on the Internet routing architecture [17] and the use of BGP for interdomain routing [18, 19]. We also summarize previous work on the Internet topology discovery.

2.1 Internet Architecture

The Internet consists of a large collection of hosts interconnected by networks of links and routers. The Internet is divided into thousands of distinct regions of administrative domain, each of which possesses one or several autonomous systems (ASes). Examples of administrative domain range from college campuses and corporate networks to large Internet Service Providers (ISPs) such as AT&T or MCI Worldcom. Each AS in the Internet is represented by a 16-bit AS number, which brings to a total of 65536 possible ASes. Not all AS numbers are assigned to administrative domains and some assigned AS numbers are not used. On January 2, 2000, there are at least 6474 ASes in use [20]. Many ISPs possess several ASes. For example, MCI Worldcom owns at least 143 ASes on Dec. 10, 1997 [20]. An AS has its own routers and routing policies, and connects to other ASes to exchange traffic with remote hosts. A router typically has very detailed knowledge of the topology within its AS, and limited reachability information about other ASes. ASes interconnect at dedicated point-to-point links or public Internet exchange points (IXPs) such as MAE-EAST or MAE-WEST. Public exchange points typically consist of a shared medium, such as a Gigabit Ethernet or an ATM switch, that interconnects routers from several different ASes. Physical connectivity at the IXP does not necessarily imply that every pair of ASes exchanges traffic with each other.

We can model the connectivity between ASes in the Internet using an AS graph $G = (V, E)$, where the node set V consists of ASes and the edge set E consists of AS pairs that exchange traffic with each other. Note that the edges of AS graph represent logical relationships between ASes and do not represent the form of the physical connection. Figure 1 shows an example of an AS graph. The *degree* of an AS is the number of ASes that are its neighbors. Formally, the degree of AS u, $D(u) = |\{v|(u,v) \in E\}|$. The degree of an AS can be a good heuristic in determining the size of the AS. In [5], AS degrees have been used to classify ASes into four levels of hierarchy.

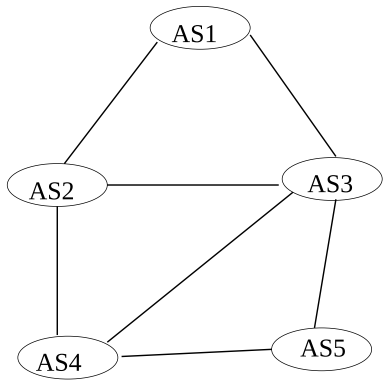

Figure 1: *An AS graph example*

Each AS has responsibility for carrying traffic to and from a set of customer IP addresses. The scalability of the Internet routing infrastructure depends on the aggregation of IP addresses in contiguous blocks, called *prefixes*, each consisting of a 32-bit IP address and a mask length (e.g., 1.2.3.0/24). An AS employs an *intradomain* routing protocol (such as OSPF or IS-IS) to determine how to reach each customer prefix, and employs an *interdomain* routing protocol (such as BGP) to advertise the reachability of these prefixes to neighboring ASes. We denote the set of prefixes that are originated from AS u by $O(u)$.

Since the commercialization of the Internet in 1995, the Internet has experienced tremendous growth in both size and complexity. The interconnections between ASes are dynamically evolving since ISPs can add or remove connections to other ASes and companies can change their Internet service providers. Furthermore, the contractual agreements between ASes can change due to ISP merging and restructuring. There are several registration services for the administration and registration of IP and AS numbers. ARIN [1] is an Internet registry that provides the WHOIS lookup service in North America, South America, the Caribbean and sub-Saharan Africa. The WHOIS service provides information about each AS such as the name and address of the administrative domain that the AS belongs to. Other registration services include RIPE NCC, which provides services for Europe, the Middle East and parts of Africa, and APNIC, which provides services for Asia Pacific. However, answering simple questions such as which ASes belong to an ISP or which prefixes are originated from AS u is not a straightforward undertaking. There is no one-to-one relationship between AS numbers and ISPs, and networks are at times connected via multiple ISPs.

2.2 Routing Policies and BGP Routing Tables

BGP is a path-vector protocol that constructs paths by successively propagating updates between pairs of BGP speaking routers that establish *BGP peering sessions* [19, 18]. Each update r concerns a particular prefix, $r.prefix$, and includes the list of the ASes along the path (the *AS path*), $r.as\_path$. Each BGP speaking router originates updates for one or more prefixes, and can send the updates to its immediate neighbors via BGP sessions. The simplest path-vector protocol would employ shortest AS path routing, where each AS selects a route with the shortest AS path. However, BGP allows a much wider range of routing policies so as to honor contractual agreements that control the exchange of traffic. Upon receiving an update, a router must decide whether or not to use this path according to *import policies* and, if the path is chosen, whether or not to propagate the update to neighboring ASes according to *export policies*. Routing policies are set by manipulating update attributes including next-hop interface address ($r.next\_hop$), local preference ($r.local\_pref$), multiple-exit discriminator ($r.med$), and community set ($r.c\_set$) as described in the following paragraphs. Routing policies are configured on each BGP speaking router. For the simplicity of exposition, we use an AS to represent BGP speaking routers in the AS and use the AS and its BGP speaking routers interchangeably throughout this paper.

An AS uses import policies to transform incoming route updates. These import policies include denying an update, or permitting an update and assigning a *local preference* to indicate how favorable the path is. We consider a BGP session $(u,v) \in E$ between two ASes, u and v. v receives a set of route updates R from u. Let import$(u,v)[R]$ represent v's update set after applying the import policy. For example, an import policy could assign $r.local\_pref = 100$ if AS 1 appears in $r.as\_path$ or deny any update that includes AS 2 in $r.as\_path$. Further, BGP discards a routing update when v already appears in the AS path of the update; this is essential to avoid introducing a cycle in the AS path. That is, BGP has the following *loop-avoidance* rule:

$$\text{if } v \in r.as\_path, \text{ then } import(u,v)[\{r\}] = \{\}$$

After applying the import policies for route updates from a BGP session, an AS saves all the imported updates in its BGP routing table. The AS then follows a route selection process that picks the best route for each prefix. Let $B(u,d)$ denote the best route selected by u for prefix d. $B(u,d)$ is selected by picking the route with the highest $local\_pref$, breaking ties by selecting the route with the shortest as_path. Note that local preference overrides the AS-path length. Among the remaining routes, the AS picks the one with the smallest *med*, breaking ties by selecting the route with the smallest intradomain routing cost. If a tie still exists, further tie-breaking rules can be found at [19].

Each AS sends only its best route for a prefix to a neighbor. Export policies allow an AS to determine whether to send its best route to a neighbor and, if it does send the route, what hint it should send to its neighbor on using the route. AS u applies export policies $export(v, u)$ to its best route set, R, for sending to a neighboring AS v. Export policies include permitting or denying a route, assigning *multiple exit discriminator* (to control how traffic enters its network), adding a community value to *community set* (to hint on what preference a neighbor should give to the route), and prepending u one or more times to AS path (to discourage traffic from entering its network by inflating the length of the AS path listing its AS number multiple times). For example, AS u could decline to advertise routes to AS v that have community 10 in the community set. Also, AS u could prepend u two times to the AS path for prefix 1.2.3.0/24 and for any route that includes AS 2 in the AS path. For any route update r, an AS always applies an implicit policy that sets $r.local\_pref$ and $r.med$ to default values, assigns $r.next\_hop$ to u's interface connecting to v, and prepends u to $r.as\_path$. Ultimately, the export policy transforms the set of updates R as $export(v, u)[R]$, which u transmits to v using a BGP session.

Each BGP speaking router keeps a *BGP routing table*, which stores a set of candidate routes for the router. We refer to a candidate route as a *routing table entry*, which includes a destination prefix, next-hop, med, local preference and AS path of the route. For the sake of simplicity, we describe the routing table entries for a fixed prefix d. The routing table entry in AS u for destination d is a route with empty AS path, denoted as e(u,d), if u originates prefix d. Otherwise, the routing table entries in u for d depend on the best route of its neighboring AS v, $B(v, d)$, as well as the import policies of u from v and the export policies of v to u. Formally, the routing table entries of u for destination prefix d

$$Routing\_Table(u, d) = \begin{cases} e(u, d) & \text{if } d \in O(u) \\ \cup_{(u,v)\in E} import(v, u)[export(u, v)[B(v, d)]] & \text{otherwise} \end{cases} \quad (1)$$

We show a router's BGP routing table entries for destination prefix 4.2.24.0/21 below. The AS has five candidate routes to 4.2.24.0/21: AS path (1740 1) via next_hop 134.24.127.3, AS path (5459 5413 1) via next_hop 194.68.130.254, etc. Note that the third candidate route has AS path (1849 702 702 701 1), where AS 702 appears twice consecutively. This is due to AS prepending; AS 702 appends its AS number twice before exporting to AS 1849. Since we are interested in inferring AS relationships in this paper, the extra appearance of an AS number does not give us additional information in this context. Therefore, for the sake of simplicity, we assume that the AS path in the BGP routing table entry is preprocessed so that no AS appears more than once throughout this paper. In addition, we list ASes in an AS path in the order that ASes are traversed when a packet is sent from the source to the destination throughout this paper.

```
Network        Next Hop        Path
*> 4.2.24.0/21 134.24.127.3    1740 1 i
*              194.68.130.254  5459 5413 1 i
*              158.43.133.48   1849 702 702 701 1 i
*              193.0.0.242     3333 286 1 i
*              144.228.240.93  1239 1 i
```

An AS can specify a diverse set of routing policies including its preference on route selection and filtering. However, routing policies are typically constrained by commercial contractual agreements negotiated between administrative domain pairs. Routing policies are often manually configured in BGP speaking routers by administrative domain operators. The potential for the various policies to conflict with and contradict one another is enormous [15]. To address these challenges, Internet Routing Registries (IRR), a distributed database of routing registries, was created. The aim of IRR is to act as a repository of routing policies and to perform consistency checking on the registered information. However, not all ISPs are willing to reveal their policies and even if they are, Routing Policy Specification Language (RPSL) [21, 22] has only been recently standardized. ISPs are still in the early stage of migrating to the new standard. As a result, information on IRR is far from complete.

2.3 Related Work

The increasing importance and complexity of the Internet routing infrastructure has sparked interest in understanding Internet topology and its effect on the end-to-end performance. Previous work consists of discovering the Internet topology, constructing the Internet distance map, and identifying inherent structural properties of the Internet. Several studies [6, 9] present heuristics on discovering the router *adjacencies* by effectively using the *traceroute tool*. Motivated by important problems such as the mirror site placement, Jamin and Theilmann et al study the construction of distance maps by estimating the end-to-end distance using strategically placed measurement servers [7, 8, 10]. Faloutsos et al identify the power-law properties of the Internet connectivity at both router and AS levels [4]. In [5], inter-AS connectivity is characterized by a hierarchy of ASes, where ASes are classified into four levels by the degree of the AS.

With the exception of [5], all of the aforementioned work do not have explicit notion of AS hierarchy. To the best of our knowledge, all studies have assumed that the connectivity is equivalent to the reachability and there is no explicit notion of AS relationships in the topology characterization. Our paper is the first study that explores AS relationships, which is an inherent aspect of the policy-based Internet routing structure. The information about AS relationships is crucial in fully understanding structural properties of the Internet. Further, AS relationships can help to effectively place measurement servers and better approximate end-to-end distances.

3 AS Relationships and Routing Table Entry Patterns

Our algorithm for inferring AS relationships is based on the fact that each AS sets up its export policies according to its relationships with neighboring ASes. In this section, we describe the annotated AS graph representation, export polices, and routing table entry patterns resulted from the export polices.

3.1 Annotated AS Graph and Selective Export Rule

We propose to represent AS relationships by an annotated AS graph. An *annotated AS graph* is a partially directed graph whose nodes represent ASes and whose edges are classified into provider-to-customer, customer-to-provider, peer-to-peer and sibling-to-sibling edges. Furthermore, only edges between providers and customers are directed. When traversing an edge from a provider to a customer, we refer to the edge as a *provider-to-customer edge*. When traversing an edge from a customer to a provider, we refer to the edge as a *customer-to-provider edge*. We call the edge between two ASes that have a peering relationship a *peer-to-peer edge* and the edge between two ASes that have a sibling relationship a *sibling-to-sibling edge*. Figure 2 shows an example of an annotated AS graph.

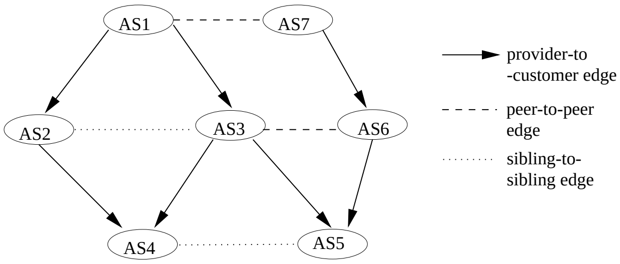

Figure 2: *An Annotated AS Graph*

Each AS sets up its export policies according to its relationships with neighboring ASes. We define $customer(a)$, $peer(a)$, $sibling(a)$, and $provider(a)$ as the set of customers, peers, siblings, and providers of a, respectively. We classify the set of routes for an AS into customer, provider, and peer routes. A route r of AS u is a *customer (provider, or peer) route* if the first non-sibling-to-sibling edge in $r.as\_path$ is a provider-to-customer (customer-to-provider, or peer-to-peer) edge. More precisely, let $r.as\_path = (u_1, \ldots, u_n)$. If (u_i, u_{i+1}) is a sibling-to-sibling edge for all $i < j$ and (u_j, u_{j+1}) is a provider-to-customer (customer-to-provider or peer-to-peer) edge, then r is a customer (provider or peer) route. If (u_i, u_{i+1}) is a sibling-to-sibling edge for all $i < n$. then r is defined to be u's own route. The AS relationships translate into the following rules that govern BGP export policies [23, 3]:

- **Exporting to a provider:** In exchanging routing information with a provider, an AS can export its routes and its customer routes, but *usually does not export* its provider or peer routes.
- **Exporting to a customer:** In exchanging routing information with a customer, an AS can export its routes and its customer routes, and as well as its provider or peer routes.
- **Exporting to a peer:** In exchanging routing information with a peer, an AS can export its routes and its customer routes, but *usually does not export* its provider or peer routes.
- **Exporting to a sibling:** In exchanging routing information with a sibling, an AS can export its routes and routes of its customers, and as well as its provider or peer routes.

In summary, an AS selectively provides transit services for its neighboring ASes. An AS sets up its export policy according to the following rule.

Selective Export Rule :
(a) Consider AS u and AS $v \in provider(u) \cup peer(u)$. For each best route r of u, if r is a provider or peer route of u, then $export(v, u)[\{r\}] = \{\}$.
(b) Consider AS u and AS $v \in customer(u) \cup sibling(u)$. There is a best route r of u such that r is a provider or peer route of u, and $export(v, u)[\{r\}] \neq \{\}$.

Note that although exporting policies are the same for providers and peers (or customers and siblings), provider-customer relationships are asymmetric and peering (or sibling) relationships are symmetric, which is the key in distinguishing provider-customer relationships from peering (or sibling) relationships. Formally, AS u *transits traffic* for AS v iff AS u transits some of its provider or peer routes to AS v, i.e., there is a best route r of u such that r is a provider or peer route of u and $export(v, u)[\{r\}] \neq \{\}$. Now we can determine AS relationships as follows.

- ASes u and v have a peering relationship iff neither u transits traffic for v nor v transits traffic for u.
- AS u is a provider of AS v iff u transits traffic for v and v does not transit traffic for u.
- ASes u and v have a sibling relationship iff both u transits traffic for v and v transits traffic for u.

Note that the relationship between two ASes might not directly correspond to the business or commercial agreement between the administrative domains that the ASes belong to. It is possible to have a commercial agreement between two ASes that includes a mixture of provider-customer, peering, and mutual-backup agreements. Two ASes might set up different export policies at different BGP sessions between the ASes. The relationship between two ASes reflects the most dominant commercial agreement between their respective administrative domains. The commercial agreements can be ordered from the most dominant to the least dominant as follows: mutual transit/backup, provider-customer, and peering agreement.

3.2 Routing Table Entry Patterns

The selective export rule indicates that a BGP routing table entry should have a certain pattern. Before we explain the pattern, we present a lemma that infers export policies from routing table entries. This lemma aids us to derive the routing table entry patterns.

Lemma 3.1 *If u_0's BGP routing table contains an entry with AS path $(u_1, u_2, \ldots, u_n)$ for destination prefix d, i.e., there is an entry e such that $e \in Routing\_Table(u_0, d)$ and $e.as\_path = (u_1, u_2, \ldots, u_n)$, then we conclude that for $1 \leq i \leq n$,*
(a) u_i selects a route with as_path $(u_{i+1}, \ldots, u_n)$ as the best route to prefix d, i.e., $B(u_i, d).as\_path = (u_{i+1}, \ldots, u_n)$.
(b) u_i exports its best route to u_{i-1}, i.e.,
$export(u_{i-1}, u_i)[\{B(u_i, d)\}] \neq \{\}$.

PROOF: We prove by induction on i. We first prove for the case that $i = 1$. From Equation (1), $B(u_1, d).as\_path = (u_2, \ldots, u_n)$ since otherwise u_0's BGP routing table does not contain an entry with AS path $(u_1, u_2, \ldots, u_n)$ for destination prefix d. If $export(u_0, u_1)[\{B(u_1, d)\}] = \{\}$, then the routing table of u_0 does not contains a route to d with AS path $(u_1, \ldots, u_n)$. Therefore, $export(u_0, u_1)[\{B(u_1, d)\}] \neq \{\}$. Suppose the lemma is true for $i < k$. That is, $B(u_{k-1}, d).as\_path = (u_k, \ldots, u_n)$. Then u_{k-1}'s BGP routing table contains an entry with AS path $(u_k, \ldots, u_n)$ for destination prefix d. Now we can use the same argument for $i = k$ as for $i = 1$. ∎

In the next theorem, we show that the selective export rule and Lemma 3.1 ensure that the AS path of an BGP routing table entry has the property

Valley-free: after traversing a provider-to-customer or peer-to-peer edge, the AS path can not traverse a customer-to-provider or peer-to-peer edge. Formally, an AS path $(u_1, u_2, \ldots, u_n)$ is valley-free iff the following conditions hold true.

- **A provider-to-customer edge can be followed by only provider-to-customer or sibling-to-sibling edges**: If (u_i, u_{i+1}) is a provider-to-customer edge, then (u_j, u_{j+1}) must be either a provider-to-customer or a sibling-to-sibling edge for any $i < j < n$.
- **A peer-to-peer edge can be followed by only provider-to-customer or sibling-to-sibling edges**: If (u_i, u_{i+1}) is a peer-to-peer edge, then (u_j, u_{j+1}) must be either a provider-to-customer or a sibling-to-sibling edge for any $i < j < n$.

or example, in Figure 3, AS paths (1, 2, 3) and (1, 2, 6, 3) are valley-free while AS paths (1, 4, 3) and (1, 4, 5, 3) are not valley-free. Note that the selective export rule ensures that BGP routing table entries contain only valley-free AS paths. For example, if AS path (1, 4, 3) appears in a BGP routing table, then AS 4 exports its provider route (3) to its provider AS 1. This violates the selective export rule. Formally, we have the following theorem.

Theorem 3.1 *If all ASes set their export policies according to the selective export rule, then the AS path in any BGP routing table entry is valley-free.*

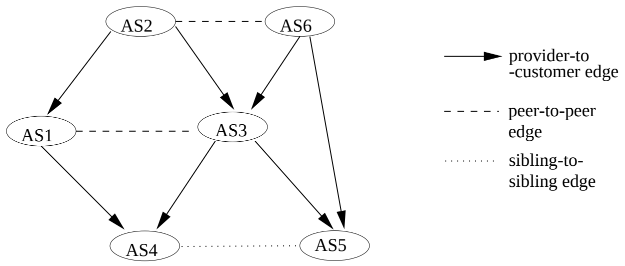

Figure 3: *AS paths (1, 2, 3) and (1, 2, 6, 3) are valley-free while AS paths (1, 4, 3) and (1, 4, 5, 3) are not valley-free.*

PROOF: We prove by contradiction. Suppose that AS path $(u_1, u_2, \ldots, u_n)$ in a BGP routing table entry is not valley-free. Let d be the destination prefix for the routing table entry. AS path $(u_1, u_2, \ldots, u_n)$ contains either (a) a provider-to-customer edge that is followed by a customer-to-provider or peer-to-peer edge, or (b) a peer-to-peer edge that is followed by a customer-to-provider or peer-to-peer edge.

In the case of (a), there is $i < n$ such that (u_i, u_{i+1}) is a provider-to-customer edge and there is $k > i$ such that (u_k, u_{k+1}) is a customer-to-provider or peer-to-peer edge. Assume that j is smallest k such that (u_k, u_{k+1}) is a customer-to-provider or peer-to-peer edge. This means that (u_{m-1}, u_m) is either a provider-to-customer or sibling-to-sibling edge for $i < m \leq j$. Let l be the largest m such that (u_{m-1}, u_m) is a provider-to-customer edge. That is, for $l < k < j$, (u_{k-1}, u_k) is a sibling-to-sibling edge. From Lemma 3.1, we have that $B(u_l, d)$ is a provider route of u_l and $export(u_{l-1}, u_l)[\{B(u_l, d)\}] \neq \{\}$. However, this contradicts the selective export policy rule since u_{l-1} is a provider or peer of u_l.

In the case of (b), a similar argument applies. ∎

The valley-free property derived from Theorem 3.1 enables us to identify patterns for BGP routing table entries. We have a corollary that indicates such patterns. But first, we define notations that simplify the description of the routing table entry patterns.

Downhill Path: a sequence of edges that are either provider-to-customer or sibling-to-sibling edges. Formally, a path $(u_1, u_2, \ldots, u_{n-1}, u_n)$ is a downhill path iff (u_i, u_{i+1}) is either a provider-to-customer or a sibling-to-sibling edge for all $i < n$.

Uphill Path: a sequence of edges that are either customer-to-provider or sibling-to-sibling edges. Formally, a path $(u_1, u_2, \ldots, u_n)$ is an uphill path iff (u_i, u_{i+1}) is either a customer-to-provider or a sibling-to-sibling edge for all $i < n$.

Corollary 3.1 *An AS path of a BGP routing table entry has one of the following patterns:*
(a) an uphill path,
(b) a downhill path,
(c) an uphill path followed by a downhill path,
(d) an uphill path followed by a peer-to-peer edge,
(e) a peer-to-peer edge followed by a downhill path,

or (f) an uphill path followed by a peer-to-peer edge, which is followed by a downhill path.

It is easy to verify that any other types of AS paths are not valley-free. Corollary 3.1 implies that an AS path can be partitioned into either:

(a) the maximal uphill path, the peer-to-peer edge and the maximal downhill path in order or
(b) the maximal uphill path and the maximal downhill path in order,

where the maximal uphill path and the maximal downhill path are defined as follows.

Maximal uphill path: the longest uphill path in the AS path. Formally, $(u_1, \ldots, u_i)$ is the maximal uphill path of AS path $(u_1, \ldots, u_n)$ iff $(u_1, \ldots, u_i)$ is an uphill path and (u_i, u_{i+1}) is a provider-to-customer or peer-to-peer edge.

Maximal downhill path: the remaining AS path after removing the maximal uphill path and the peer-to-peer edge. Formally, $(u_j, \ldots, u_n)$ is the maximal downhill path of AS path $(u_1, \ldots, u_n)$ iff $(u_j, \ldots, u_n)$ is a downhill path and (u_{j-1}, u_j) is a peer-to-peer edge or belongs to the maximal uphill path.

Note that any one or both of the maximal uphill path and the maximal downhill path of an AS path can be empty. An AS path can have an uphill top provider and a downhill top provider, where the *uphill top provider* is the last AS in its maximal uphill path and the *downhill top provider* is the first AS in its maximal downhill path. Note that an AS path's uphill top provider and downhill top provider are the same AS if there is no peer-to-peer edge in the AS path. If the uphill and downhill top providers are known, then we can infer the relationship between any consecutive pair of the AS path. Therefore, identifying the uphill and downhill top providers is the key in inferring AS relationships.

The goal of this paper is to produce an annotated AS graph by taking advantage of BGP routing table entry patterns. In other words, given BGP routing tables, we derive an annotated AS graph G that is consistent with the BGP routing tables. In the next two sections, we present heuristic algorithms that use BGP routing tables to infer AS relationships and show experimental results derived from BGP routing tables.

4 Heuristic Algorithms for Inferring AS Relationships

In this section, we present heuristic algorithms for inferring AS relationships given a set of routing tables. Our algorithms are based on the intuition that a provider typically has a larger size than its customer does and the size of an AS is typically proportional to its degree in the AS graph. The uphill (or downhill) top provider of an AS path should be the AS that has the highest degree among all ASes in its maximal uphill (or downhill) path. Let the *top provider* of an AS path to be the AS that has a higher degree between the uphill and downhill top provider. Therefore, the top provider of an AS path is the AS that has the highest degree among all ASes in the AS path.

For the sake of simplicity, we first classify edges into provider-to-customer and sibling-to-sibling edges. Now, we can infer that consecutive AS pairs that appear before the top provider in the AS path are customer-to-provider or sibling-to-sibling edges, and consecutive pairs that appear after the top provider in the AS path are provider-to-customer or sibling-to-sibling edges. We then identify peer-to-peer edges from the set of AS pairs that appear only as the top provider and the top provider's neighbor in an AS path. We first show algorithms that classify AS relationships into provider-to-customer and sibling-to-sibling edges in Section 4.1, and then present an algorithm that identifies AS pairs that have peering relationships in Section 4.2.

4.1 Algorithms for Inferring Provider -Customer and Sibling Relationships

In this section, we first present a basic algorithm for inferring provider-customer and sibling relationships in Section 4.1.1. We then refine this algorithm in Section 4.1.2.

4.1.1 Basic Algorithm

Our basic heuristic algorithm goes through the AS path of each routing table entry. It finds the highest degree AS and lets the AS be the top provider of the AS path. Knowing the top provider, we can infer that consecutive AS pairs before the top provider are customer-to-provider or sibling-to-sibling edges, and consecutive AS pairs after the top provider are provider-to-customer or sibling-to-sibling edges. Note that we traverse an AS path in the order that ASes are visited when a packet is sent from the source to the destination. More precisely, if an AS pair (u_1, u_2) appears before the top provider of an AS path, then u_2 provides transit services for u_1, and if an AS pair (u_1, u_2) appears after the top provider of an AS path, then u_1 provides transit services for u_2. Therefore, u_1 is a provider of u_2 iff u_1 provides transit services for u_2 and u_2 does not provide transit services for u_1. An AS pair have a sibling relationship if the pair provide transit services for each other.

This leads to a three-phase heuristic algorithm for inferring provider-customer and sibling relationships. The first phase parses routing tables and calculates the degree of each AS. The second phase parses each entry of the routing tables. It first identifies the top provider and then assigns consecutive AS pairs before the top provider with a transit relationship and consecutive AS pairs after the top provider with a transit relationship. Figure 4 shows the basic algorithm in details.

The basic algorithm has the running time complexity of $O(N)$, where N is the total number of consecutive AS pairs in the routing tables. As we will see later, we evaluate the algorithm by using a publicly available routing table, in

Basic Algorithm:
Input: BGP routing tables
Output: Annotated AS graph G

Phase 1: Compute the degree for each AS

For each as_path $(u_1, u_2, \ldots, u_n)$ in routing tables,
    for each $i = 1, \ldots, n-1$,
        neighbor$[u_i]$ = neighbor$[u_i]$ $\cup\{u_{i+1}\}$
        neighbor$[u_{i+1}]$ = neighbor$[u_{i+1}]$ $\cup\{u_i\}$
For each AS u,
    degree$[u]$ = $|$neighbor$[u]|$

Phase 2: Parse AS path to initialize consecutive AS pair's transit relationship

For each as_path $(u_1, u_2, \ldots, u_n)$ in RT,
    find the smallest j such that degree$[u_j]$= $\max_{1 \le i \le n}$ degree$[u_i]$
    for $i = 1, \ldots, j-1$,
        transit$[u_i, u_{i+1}]$=1
    for $i = j, \ldots, n-1$,
        transit$[u_{i+1}, u_i]$=1

Phase 3: Assign relationships to AS pairs

For each AS path $(u_1, u_2, \ldots, u_n)$,
    for $i = 1, \ldots, n-1$,
        if transit$[u_i, u_{i+1}]$=1 and transit$[u_{i+1}, u_i]$=1
            edge$[u_i, u_{i+1}]$ = sibling-to-sibling
        else if transit$[u_{i+1}, u_i]$=1
            edge$[u_i, u_{i+1}]$=provider-to-customer
        else if transit$[u_i, u_{i+1}]$=1
            edge$[u_i, u_{i+1}]$=customer-to-provider

Figure 4: *Basic Heuristic Algorithm*

which there are 1 million route entries and N is over 2.6 million. Therefore, it is important to construct a linear time algorithm in N.

4.1.2 Refined Algorithm

The basic algorithm assumes that all BGP speaking routers are configured correctly. However, it is possible that some BGP speaking routers are misconfigured in the sense that they do not conform to the selective export rule. This might lead to incorrect inference of AS relationships from routing tables. For example, AS u and AS v are providers of AS w. However, AS w misconfigures its BGP speaker router such that AS w transits traffic between ASes u and v. In the routing table of AS u, there is a routing table entry with AS path (u, w, v). Suppose AS v has the highest degree among the three ASes. The Basic algorithm infers that w transits traffic for u, which contradicts with the fact that u is a provider of w. To reduce the possibility of incorrect inference, we propose a refined algorithm that determines AS relationships by counting the number of routing table entries that conclude transit relationships. In the refined algorithm, we assume that misconfigured BGP speaking routers affect only a small number of routing table entries. Specifically, we use the heuristic that if no more than L routing table entries infer that AS u provides transit services for AS v and more than L routing table entries infer that AS v provides transit services for AS u, then we ignore the routing table entries that infer that AS u transits for AS v and we conclude that u is a customer of v, where L is a small constant.

The refined algorithm infers AS relationships as follows. The first phase parses routing tables and calculates the degree of each AS. The second phase parses each entry of the routing tables. It counts the number of routing table entries that infer an AS pair having a transit relationship by assigning consecutive AS pairs before the top provider with a transit relationship and consecutive AS pairs after the top provider with a transit relationship. The third phase finalizes the relationship between AS pairs. If more than L routing table entries infer that AS u transits traffic for AS v and more than L routing table entries infer that AS v transits traffic for AS u, then v is a sibling of u. If at least one and at most L routing table entries infer that AS u transits traffic for AS v and at least one and at most L routing table entries infer that AS v transits traffic for AS u, then v is a sibling of u. Otherwise, if no routing table entry infers that u transits traffic for v or at least L routing table entries infer that v transits traffic for u, then v is a provider of u. Note that unlike the basic algorithm, the refined algorithm ignores some routing table entries. Figure 5 shows the refined algorithm in details.

4.2 A Heuristic Algorithm for Inferring Peering Relationships

Both the basic and refined algorithms classify AS relationships into provider-customer or sibling relationships only. In this section, we present a heuristic algorithm for identifying peering relationships. An AS pair have a peering relationship if and only if the AS pair do not transit traffic for each other. Therefore, we first identify all AS pairs that have transit relationships. According to Corollary 3.1, if an AS pair appear consecutively in an AS path and neither of the AS pair is the top provider of the AS path, then the AS pair have a transit relationship and therefore, the AS pair do not have a peering relationship. Furthermore, according to Corollary 3.1, an AS path has at most one consecutive AS pair that have a peering relationship. That is, a top provider can have a peering relationship with at most one of its neighbors in the AS path. Since an AS pair that have a peering relationship are typically of comparable size, we identify the neighboring AS of the top provider that the top provider does not have a peering relationship with using the heuristic that the top provider is more likely to peer with its neighbor with a higher degree. That is, if the top provider does not have a sibling relationship with anyone of its neighboring ASes in the AS path, then the top provider does not have a peering relationship with the neighbor with smaller degree.

Finally, since we might not have routing tables from all

Refined Algorithm:
Input: BGP routing tables
Output: Annotated AS graph G

Phase 1: Same as Phase 1 in the basic algorithm

Phase 2: Count the number of routing table entries that infer an AS pair having a transit relationship

For each AS path $(u_1, u_2, \ldots, u_n)$,
  find the smallest j such that degree[u_j]=$\max_{1 \le i \le n}$ degree[u_i]
  for $i = 1, \ldots, j-1$,
    transit[u_i, u_{i+1}]=transit[u_i, u_{i+1}]+1
  for $i = j, \ldots, n-1$,
    transit[u_{i+1}, u_i]=transit[u_{i+1}, u_i]+1

Phase 3: Assign relationships to AS pairs

For each AS path $(u_1, u_2, \ldots, u_n)$,
  for $i = 1, \ldots, n-1$,
    if (transit[u_{i+1}, u_i]$> L$ and transit[u_i, u_{i+1}]$> L$)
      or (transit[u_i, u_{i+1}]$\le L$ and transit[u_i, u_{i+1}]> 0
      and transit[u_i, u_{i+1}]$\le L$ and transit[u_i, u_{i+1}]> 0)
      edge[u_i, u_{i+1}] = sibling-to-sibling
    else if transit[u_{i+1}, u_i]$> L$ or transit[u_i, u_{i+1}]$= 0$
      edge[u_i, u_{i+1}]=provider-to-customer
    else if transit[u_i, u_{i+1}]$> L$ or transit[u_{i+1}, u_i]$= 0$
      edge[u_i, u_{i+1}]=customer-to-provider

Figure 5: *Refined Heuristic Algorithm*

BGP speaking routers, we might not be able to identify all AS pairs that do not have peering relationships using the heuristic described above. To eliminate more AS pairs from having peering relationships, we use the heuristic that the sizes of two peers do not differ significantly. Specifically, we assume that the degrees of the two ASes that have a peering relationship do not differ by more than R times, where R is a constant that has to be fine-tuned. Note that the need for the constant R is unfortunate and it is not clear how to properly set it. Accordingly, some inaccuracy might be introduced to the corresponding inference. On the other hand, the more BGP routing tables we use for the inference, the less crucial the choice of R is. This is because we can eliminate an AS pair from having a peering relationship if the AS pair appear in any BGP routing table entry as having a transit relationship or being not likely to peer as described earlier. The use of R to eliminate an AS pair from having a peering relationship plays a less significant role.

The final algorithm infers peering relationships as follows. Phase 1 coarsely classifies AS pairs into having provider-customer or sibling relationships. Phase 2 identifies all AS pairs that can not peer with each other. Finally, Phase 3 identify peering relationships from the rest of connected AS pairs by using the heuristic that two peering ASes' degrees do not differ by more than R times. Figure 6 presents the final algorithm in details.

5 Inferring AS Relationships in the Internet

In this section, we present experimental results of inferring AS relationships in the Internet. Ideally, we would perform experiments on BGP routing tables of all BGP speaking routers in the Internet. However, there are a limited number of BGP routing tables publicly available. We choose to use BGP routing tables from the Route Views router in Oregon [16], which has the most complete view currently available. The Router Views router establishes BGP peering sessions with 22 ISPs at 24 locations as shown in Table 1. The Route Views server collects the BGP routing table once every night [20]. For the detailed description of the Route Views server, see [16].

5.1 Experimental Results

We implement the Basic, Refined, and Final algorithms using the perl programming language. For the Refined algorithm, we choose $L = 1$ so as to ignore fewer number of routing table entries. For the Final algorithm, we use the Basic algorithm to infer sibling relationships and let the limit of the ratio between the degrees of two peering ASes, R, be infinite or 60. Note that we choose to use $R = 60$ due to the current connectivity of top tier providers. There are only two ASes whose degrees are greater than 420 and few ASes with degree less than 7 peer with tier-1 providers. Ad-

Final Algorithm:
Input: BGP routing tables
Output: Annotated AS graph G

Phase 1: Use either Basic or Refined algorithm to coarsely classify AS pairs into provider-customer or sibling relationships

Phase 2: Identify AS pairs that can not have a peering relationship

For each AS path $(u_1, u_2, \ldots, u_n)$,
  find the AS u_j such that degree[u_j]=$\max_{1 \leq i \leq n}$ degree[u_i]
  for $i = 1, \ldots, j-2$,
    notpeering[u_i, u_{i+1}]=1
  for $i = j+1, \ldots, n-1$,
    notpeering[u_i, u_{i+1}]=1
  if edge[u_{j-1}, u_j]$\neq$sibling-to-sibling and edge[u_j, u_{j+1}]$\neq$ sibling-to-sibling
    if degree[u_{j-1}]>degree[u_{j+1}]
      notpeering[u_j, u_{j+1}]= 1
    else
      notpeering[u_{j-1}, u_j]= 1

Phase 3: Assign peering relationships to AS pairs

For each AS path $(u_1, u_2, \ldots, u_n)$,
  for j=1, ..., n-1,
    if notpeering[u_j, u_{j+1}]$\neq$ 1 and notpeering[u_{j+1}, u_j]$\neq$ 1 and
      degree[u_j]/degree[u_{j+1}]< R and degree[u_j]/degree[u_{j+1}]> $1/R$
      edge[u_j, u_{j+1}] = peer-to-peer

Figure 6: *Final Algorithm*

ANS	(Cleveland)	206.157.77.11	through AS1673
ATT	(Chicago)	12.127.0.249	through AS7018
BBNPlanet	(Palo Alto)	4.0.0.2	through AS1
CERFnet	(San Diego)	134.24.127.3	through AS1740
DIGEX	(MAE-EAST)	192.41.177.192	through AS2548
EBONE	(EU)	192.121.154.25	through AS1755
ESnet	(GA)	134.55.24.6	through AS293
RIPE NCC	(Amsterdam)	193.0.0.56	through AS3333
IAGnet	(Chicago)	204.42.253.253	through AS267
IIJ	(Japan)	202.232.1.8	through AS2497
JINX	(Johannesburg)	196.7.106.152	through AS2905
LINX	(London)	194.68.130.254	through AS5459
C&W USA	(San Francisco)	204.70.4.89	through AS3561
PIPEX	(London)	158.43.133.48	through AS1849
Sprint	(Stockton)	144.228.240.93	through AS1239
vBNS	(Hayward)	204.147.128.137	through AS145
Verio	(MAE-WEST)	129.250.0.3	through AS2914
Verio	(MAE-EAST)	129.250.0.1	through AS2914
blackrose.org	(Ann Arbor)	204.212.44.128	through AS234
Abilene	(Indiana)	198.32.8.252	through AS11537
Concentric	(MAE-WEST)	205.158.2.126	through AS2828
GIGABELL	(Frankfurt)	195.211.222.254	through AS5409
GIGABELL	(MAE-FRANKFURT)	195.211.222.6	through AS5409
GIGABELL	(Espanix)	195.211.222.13	through AS5409

Table 1: *Current contributors of route views*

mittedly, this might exclude some peer-to-peer edges. We will see in the next subsection that fine-tuning R to be 60 can significantly improve the inference result for peering relationships. We run the algorithms for the BGP routing tables from 1999/9/27, 2000/1/2 and 2000/3/9. The number of edges in the AS graph, the number of sibling-to-sibling edges inferred by both the Basic and Refined algorithms, and the number of peer-to-peer edges inferred by the Final algorithm are shown in Table 2.

Note that the total number of edges in the AS graph is inconsistent with the publicly available data at [20]. In [20], AS summary data indicates that there are 13895 edges on 2000/1/2 and 12468 edges on 1999/9/27. Because of the AS_prepend operation in BGP, an AS can appear more than once in a routing table entry. The perl script [20] overcounts the number of edges by including self edges (A self edge is an edge between an AS and itself) when parsing the BGP routing table. We eliminate self edges in our perl programs.

These BGP routing tables contain almost 1 million routing table entries. From the BGP routing table on 1999/9/27, the Basic and Final algorithms infer that among 11288 AS graph edges, there are 10745 provider-to-customer edges, 149 sibling-to-sibling edges, and 884 peer-to-peer edges. By using the Refined algorithm, the number of sibling-to-sibling edges is reduced to 124 and by using the Final algorithm with $R = 60$, the number of peer-to-peer edges is reduced to 733.

From the BGP routing table on 2000/1/2, the Basic and Final algorithms infer that among 12571 AS graph edges, there are 12013 provider-to-customer edges, 186 sibling-to-sibling edges, and 372 peer-to-peer edges. By using the Refined algorithm, the number of sibling-to-sibling edges is reduced to 135 and by using the Final algorithm with $R = 60$, the number of peer-to-peer edges is reduced to 668. From the BGP routing table on 2000/3/9, the Basic and Final algorithms infer that among 13800 AS graph edges, there are 13661 provider-to-customer edges, 203 sibling-to-sibling edges, and 836 peer-to-peer edges. By using the Refined algorithm, the number of sibling-to-sibling edges is reduced to 157 and by using the Final algorithm with $R = 60$, the number of peer-to-peer edges is reduced to 713. Therefore, for all three routing tables, we can find a consistent view of AS relationships which has more than 90.5% provider-to-customer edges, less than 1.5% of sibling-to-sibling edges and less than 8% of peer-to-peer edges. Note that the small percent of peer-to-peer edges might be caused by the incomplete view of the Route Views router. Since the Route Views router peers with mostly tier-1 providers, peering between tier-2 or tier-3 ISPs might not be manifested in the Route Views routing table due to the selective export rule and the fact that only the best routes are exported. We also observe that the number or the percentage of sibling-to-sibling edges is increasing. It might be caused by the increasing number of complex AS relationships and ISP mergers.

Our Inference	AT&T Information	Percentage of ASes
Customer	Customer	100%
Provider(1)	Peer	100%
Peer	Peer	77.4%
	Customer	22.6%
Sibling	Sibling	20%
	Peer	60%
	Customer	20%
Nonexistent	Customer	95.6%
	Peer	4.4%
Overall	Confirmed	96.3%
	Unconfirmed	3.7%

Table 3: *Comparing inference results from Basic and Final($R = \infty$) with AT&T internal information*

5.2 Verification of Inferred Relationships by AT&T

Although there is no publicly available information about AS relationships, we verify our inferred relationships by comparing with AT&T internal information on AT&T Common IP Backbone. We compare our data for March 9, 2000 with that of AT&T Common IP Backbone on the same day.

Table 3 compares the inference results from Basic and Final($R = \infty$) algorithms with AT&T internal information. Since we can not reveal AT&T internal information on each type of relationship, we present the comparison results in terms of percentage except for some special cases. From the table, we see that 100% of our inferred customers are confirmed by the AT&T internal information. 0% of our inferred provider is confirmed by the AT&T internal information. Note that we infer that AT&T has only one provider while AT&T has no provider. 77.4% of our inferred peers are confirmed by the AT&T internal information. 20% of our inferred siblings are confirmed by the AT&T internal information. Note we do not necessarily have all AT&T's neighbors from the routing table of Route Views since AT&T announces only its best routes to outside and some of its announced routes are aggregate routes. Out of neighbors from the AT&T list, 20% ASes do not exist in our adjacency list and most of these ASes are customers of AT&T. Out of all of our inference results, 96.3% of inference results are confirmed by the AT&T internal information. Note that 96.3% accuracy is relative to only AT&T's relationships with its neighbors. Using AT&T's internal information, we can verify 3.32% of edges that appear in the Router Views' BGP routing table. For AT&T's relationships with its neighbors, most of the inaccurate inference is caused by the misclassification of peering relationships into provider-customer relationship. This confirms the need of fine-tuning R due to the lack of sufficient BGP routing tables. We will see later that by fine-tuning R, it is possible to achieve better accuracy for AT&T's relationships with others.

Table 4 compares the inference results from Refined($L =$

	Total Routing Entries	Total edges	Sibling-to-sibling edges inferred by Basic (Percentage)	Sibling-to-sibling edges inferred by Refined[$L = 1$] (Ignored Entries)	Peer-to-peer edges inferred by Final[$R = \infty$] (Percentage)	Peer-to-peer edges inferred by Final[$R = 60$] (Percentage)
1999/9/27	968674	11288	149 (1.3%)	124 (25)	884 (7.8%)	733 (6.5%)
2000/1/2	936058	12571	186 (1.47%)	135 (51)	838 (6.7%)	668 (5.3%)
2000/3/9	1227596	13800	203 (1.47%)	157 (46)	857 (6.2%)	713 (5.7%)

Table 2: *Inference Results*

Our Inference	AT&T Information	Percentage of ASes
Customer	Customer	100%
Provider(1)	Peer	100%
Peer	Peer	77.4%
	Customer	22.6%
Sibling	Sibling	25%
	Peer	50%
	Customer	25%
Nonexistent	Customer	95.6%
	Peer	4.4%
Overall	Confirmed	96.5%
	Unconfirmed	3.5%

Table 4: *Comparing inference results from Refined($L = 1$) and Final($R = \infty$) with AT&T internal information*

Our Inference	AT&T Information	Percentage of ASes
Customer	Customer	100%
Provider(1)	Peer	100%
Peer	Peer	100%
Sibling	Sibling	25%
	Peer	50%
	Customer	25%
Nonexistent	Customer	95.6%
	Peer	4.4%
Overall	Confirmed	99.1%
	Unconfirmed	0.9%

Table 5: *Comparing inference results from Refined($L = 1$) and Final($R = 60$) with AT&T internal information*

1) and Final($R = \infty$) with AT&T internal information. From the table, we see that 100% of our inferred customers are confirmed by the AT&T internal information. 77.4% of our inferred peers are confirmed by the AT&T internal information. 25% of our inferred siblings are confirmed by the AT&T internal information. Out of all of our inference results, 96.5% of inference results are confirmed by the AT&T internal information. Using Refined algorithm, we improve the inference results on sibling relationships.

Table 5 compares the inference results from Refined($L = 1$) and Final($R = 60$) with AT&T internal information. From the table, we see that 100% of our inferred customers are confirmed by the AT&T internal information. 100% of our inferred peers are confirmed by the AT&T internal information. 25% of our inferred siblings are confirmed by the AT&T internal information. Out of our inference results, 99.1% of inference results are confirmed by the AT&T internal information. Using the heuristic that peers are typically of comparable sizes by setting a reasonable value for R, we improve the inference results on peering relationships significantly. Note that although it is problematic to select a proper value for R, it is encouraging to see that it is possible to achieve 100% confirmation for peering relationship inference for an ISP. At the same time, this suggests that we should combine other information with our inference techniques to achieve better accuracy for important business applications.

5.3 Verifications by the WHOIS server

We verify our inferred sibling relationships by checking with the WHOIS lookup service. Since the WHOIS lookup service supplies the name and address of the company that owns an AS, we can confirm that an AS pair has a sibling relationship if the two ASes belong to the same company or two merging companies (such as AT&T and Cerfnet). Further, we also confirm that an AS pair has a sibling relationship if the ASes belong to two small companies that are located in the same city (which increases the likelihood that they have a mutual-transit agreement). We manually queried the WHOIS lookup service and confirmed 101 of 186 inferred sibling relationships for the 2000/1/2 data. This is 54.3% of the inferred sibling relationships. Note that, however, the WHOIS server might not be completely accurate since its database might contain stale records. Therefore, it is possible that the WHOIS server might falsely confirm a relationship. With that caveat in mind, we use it for the lack of a better avenue by which to verify our inference results. By the same token, other unconfirmed sibling relationships might still have sibling relationships since the WHOIS lookup service might be out of date and we do not have sufficient information about ISP mergers.

Refined algorithm reduces the number of sibling relationships by ignoring some of routing table entries as shown in Table 2. For 1999/9/27 data, the Refined algorithm infers 124 sibling-to-sibling edges by ignoring only 25 route entries. For 2000/1/2 data, the Refined algorithm infers 135 sibling-to-sibling edges by ignoring only 51 route entries. For 2000/3/9 data, the Refined algorithm infers 157 sibling-to-sibling edges by ignoring only 46 route entries. This encourages us to look into routing table entries that infer unconfirmed sibling relationships and analyze routes that might mislead us in inferring AS relationships.

We analyze the routes that contribute to the inference of unconfirmed sibling relationships. Our goal is to find the possible patterns for these routes and perhaps to use the patterns to increase the accuracy of then inference. We report here several possible reasons behind the inference of unconfirmed sibling relationships.

1. **Router Configuration Typo:** some router prepends its AS number by explicitly specifying the AS numbers to prepend. A typo in the configuration can result routing table entries that violate the loop-avoidance rule defined in BGP. For example, in AS path (7018 3561 7057 7075 7057), AS7057 appears twice and does not appear consecutively. This might be the result of the router configuration typo in AS7057.
2. **Misconfiguration of small ISPs:** some small ISPs do not follow the selective export rule in their router configuration. For example, AS path (1239 11116 701 7018) has Sprint (AS1239) uses a small ISP in California (AS11116) to get to AT&T (AS7018) via UUnet (AS701). According to the AS graph, UUnet and Sprint are directly connected although the route does not use the direct connection. It is likely that AS11116 is a customer of both AS701 and AS1239. Therefore, this route might be caused by the misconfiguration of AS11116 that announces its provider route to another provider.
3. **Unusual AS relationships:** some AS pairs have their relationships defined at the prefix level. For example, in AS path (1239 3561 2856 701 702 1849 9090), Sprint (AS1239) uses AS3561 and AS2856 to get the route of UUnet (AS701) instead of using the direct link to UUnet (AS701). Note that AS2856, AS1849 and AS9090 are European ASes. This might be the result of specifically defined relationship for prefixes in Europe.
4. **Inaccuracy of the heuristic:** the top provider does not have the highest degree. For example, in AS path (3333 7905 5727 1327), although AS3333 has the highest degree, it might not be the top provider of the AS path since AS5727 (AT&T) is likely to be the top provider. Note that AS3333 is an European ISP.

Reasons 1, 2 and 3 suggest that we have to ignore some routing table entries in inferring AS relationships. It is not clear, however, how to identify these entries. Reason 4 hints that it might be wise to modify our heuristic for special ASes. However, this can not be done without additional knowledge such as which ISP an AS belongs to. Therefore, it is a challenging task to increase the accuracy of the AS relationship inference with only BGP routing tables.

6 Conclusions and Future Work

Interdomain routing policies are constrained by commercial contractual relationships between administrative domains. As a result, AS relationships are an inherent aspect of the Internet routing structure. We present heuristic algorithms that infer AS relationships from BGP routing tables. The algorithm is based on the fact that a provider is typically larger than its customers and two peers are typically of comparable size. We perform an experimental study of AS relationships in the Internet. Out of the connected AS pairs seen from Route Views, our heuristic algorithm classifies more than 90.5% of AS pairs into provider-customer relationships, less than 1.5% of AS pairs into sibling relationships, and less than 8% of AS pairs into peering relationships. We verify our inferred relationships with both AT&T internal information and the WHOIS lookup service. 99.1% of our inferred relationships between AT&T and its neighboring ASes are confirmed by the AT&T internal information. More than 50% of inferred sibling pairs can be confirmed by the data from the WHOIS lookup service. Furthermore, we identify routing table entries that stem from unusual AS relationships or router misconfiguration/bugs.

As part of ongoing work, we are exploring heuristics that can improve AS relationship inference. Many of scenarios discussed in Section 5.3 can be combined with additional knowledge about ASes to improve our heuristic algorithms. In addition, we plan to study several applications of AS relationships. First, ISPs can reduce misconfiguration and debug router configuration files [13, 15]. Route policies are often manually configured and therefore prone to errors. Such errors can propagate further to other ASes and can potentially cause outage. Therefore, it is important for ISPs to monitor the received route announcements using AS relationship knowledge and perhaps filter erroneous routes such as a route using a customer for transit traffic between its two providers. Further, an ISP operator can scan its BGP routing tables periodically to identify potential erroneous routes and inform the originating AS. We would like to build tools to improve the reliability of Internet routing. These tools include features such as debugging router configuration files so as to conform to the selective export rule.

Second, ISPs or companies can use AS relationship information to plan for future contractual agreements. The contractual agreement between ISPs is constantly evolving. For example, a company might decide to switch to or add a tier-1 ISP as its provider. Verifying whether an ISP is a tier-1 ISPs involves understanding whether the ISP has a provider. As another example, an ISP might decide to establish private peering relationships with other ISPs as it becomes larger. The ISP might want to first understand which tier that a potential ISP belongs to before collecting information on traffic volume between the two ISPs. We plan to systematically study the AS hierarchical structure using the AS relationship information.

Acknowledgment

The author would like to thank Jennifer Rexford at AT&T Research Labs for many helpful discussion and comments on the paper, Joel M. Gottlieb at AT&T Research Labs for providing the AT&T internal information, David Meyer at CISCO for allowing me to use Route Views data, and anonymous reviewers for many valuable comments.

References

[1] `http://www.arin.net/whois/arinwhois.html`.

[2] G. Huston, "Interconnection, peering and settlements-Part I," in *Internet Protocol Journal*, March 1999.

[3] G. Huston, "Interconnection, peering and settlements-Part II," in *Internet Protocol Journal*, June 1999.

[4] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. ACM SIGCOMM*, August 1999.

[5] R. Govindan and A. Reddy, "An analysis of Internet inter-domain topology and route stability," in *Proc. IEEE INFOCOM*, April 1997.

[6] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," in *Proc. IEEE INFOCOM*, March 2000.

[7] W. Theilmann and K. Rothermel, "Dynamic distance maps of the Internet," in *Proc. IEEE INFOCOM*, March 2000.

[8] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "On the placement of Internet instrumentation," in *Proc. IEEE INFOCOM*, March 2000.

[9] R. Siamwalla, R. Sharma, and S. Keshav, "Discovering Internet topology," tech. rep., Cornell, May 1999.

[10] S. Jamin, P. Francis, V. Paxson, L. Zhang, D. Gryniewicz, and Y. Jin, "An architecture for a global Internet host distance estimation service," in *Proc. IEEE INFOCOM*, March 1999.

[11] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of Internet path selection," in *Proc. ACM SIGCOMM*, August 1999.

[12] J. Rickard, "Mapping the Internet with traceroute," in *Broadwatch Magazine*, December 1996.

[13] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental study of Internet stability and wide-area network failures," in *Proc. International Symposium on Fault-Tolerant Computing*, June 1999.

[14] V. Paxson, "End-to-end routing behavior in the Internet," in *IEEE/ACM Trans. Networking*, October 1997.

[15] A. Feldmann, "Netdb: IP network configuration debugger/database," in *AT&T Software Symposium*, September 1999.

[16] `http://www.antc.uoregon.edu/route-views/`.

[17] B. Halabi, *Internet Routing Architectures.* Cisco Press, 1997.

[18] J. W. Stewart, *BGP4: Inter-Domain Routing in the Internet.* Addison-Wesley, 1999.

[19] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)." Request for Comments 1771, March 1995.

[20] National Laboratory for Applied Network Research, `http://moat.nlanr.net/AS/`.

[21] D. Meyer, J. Schmitz, C. Orange, M. Prior, and C. Alaettinoglu, "Using RPSL in practice." Request for Comments 2650, August 1999.

[22] 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.

[23] C. Alaettinoglu, "Scalable router configuration for the Internet," in *Proc. IEEE IC3N*, October 1996.

[24] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," Tech. Rep. 96-631, USC/ISI, February 1996.

[25] T. G. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *Proc. ACM SIGCOMM*, September 1999.

[26] L. Gao and J. Rexford, "A stable Internet routing without global coordination," in *Proc. ACM SIGMETRICS*, June 2000.