On Inferring Autonomous System Relationships in the Internet

Lixin Gao*
Dept. of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01002
lgao@ecs.umass.edu

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 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 connectivity alone can not fully characterize the structural properties of the Internet. We propose an augmented AS graph representation that classifies AS relationships into customer-to-provider, peer-to-peer, and sibling-to-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 that contains close to 1 million route entries. The algorithms infer that more than 90.5% of the connected AS pairs have customer-to-provider relationships, less than 1.5% of the connected AS pairs have sibling-to-sibling relationships, and less than 8% of the connected AS pairs have peer-to-peer relationships. We verify our inference results with AT&T internal information on its relationship with neighboring ASes. 99% of our inference results are confirmed by the AT&T internal information. We also verify our inferred sibling-to-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 inter-AS relationships and this is the first attempt in understanding and inferring AS relationships in the Internet. We show evidences that some routing table entries stem from unusual AS relationships or router misconfiguration. Furthermore, we demonstrate that inferred AS relationships have many applications including enhancing the reliability of Internet routing.

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

*The author was supported in part by NSF grant ANI-9977555, 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.

is determined by the interdomain routing protocol – Border Gateway Protocol (BGP). One key distinct feature of the interdomain routing protocol is that it allows each AS to choose its own 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, and mutual-transit 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 or universities who are located close to each other and who can not afford additional Internet services for better connectivity.

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 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, the connectivity (at either the AS or the router level) alone does not fully characterize the structural properties of the Internet.

We propose an augmented AS graph representation to capture AS relationships. We classify the relationship between a pair of interconnected ASes into customer-to-provider, peer-to-peer, and sibling-to-sibling relationships according to the commercial contractual agreement between the administrative domains they belong to. A pair of interconnected ASes has a customer-to-provider (peer-to-peer) relationship if their respective administrative domains have a customer-provider (peering) relationship. Two ASes have a sibling-to-sibling relationship if they belong to the same administrative domain or their respective administrative domains have a mutual-transit agreement.

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, many ISPs are not willing to reveal their policies and even if they are, these routing policies might not specify AS relationships.

The knowledge of global inter-AS relationships have many applications. First, it can be crucial in Internet service management decisions including the optimal placement of proxy servers or web hosting servers. Second, it can help ISP or domain administrators to achieve load balancing, congestion avoidance and fault tolerance goal. Third, it can aid ISPs in planning for future contractual agreements. Fourth, it can help ISPs to reduce the effect of the misconfiguration and to debug router configuration files [13, 14, 15]. Fifth, it can potentially avoid route divergence problem. Sixth, understanding AS relationships can verify the consistency of information in IRR.

In this paper, we present heuristic algorithms that infer the augmented AS graph from BGP routing tables. Since the BGP routing tables closely capture the current state of the Internet, inferring AS relationships from them can effectively derive the most up-to-date AS relationships. 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-to-customer, peer-to-peer, or sibling-to-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. The BGP routing table is retrieved from the Route Viewer server in Oregon [16], which is publicly available and has the most complete view of the current Internet by establishing BGP peering sessions with 22 ISPs at 24 locations. The algorithms infer that more than 90.5% of the connected AS pairs have customer-to-provider relationships, less than 1.5% of the connected AS pairs have sibling-to-sibling relationships, and less than 8% of the connected AS pairs have peer-to-peer relationships. We verify our inference results with AT&T internal information on its relationship with neighboring ASes. Our result shows that 99.8% 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 our inference results, 99% of inference results are confirmed by the AT&T internal information. We also verify our inferred sibling-to-sibling relationships with the information acquired from the WHOIS lookup service [1]. More than half of the inferred sibling-to-sibling relationships are confirmed by the WHOIS lookup service. Note that not being confirmed by the WHOIS lookup service does not mean that the inference results are incorrect since the WHOIS server does not have up-to-date and complete information about ASes. We show evidences that some BGP routing table entries stem from unusual AS relationships or router misconfiguration. Furthermore, we demonstrate that inferred AS relationships have many applications including enhancing the reliability of Internet routing.

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 in Internet by using publicly available BGP routing tables. Section 6 discusses applications of understanding AS relationships such as enhancing the reliability of Internet routing. We conclude the paper in Section 7 with a summary.

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 administratrive 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 AS numbers. 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 possesses 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 FDDI ring 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.

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 (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 interconnection 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

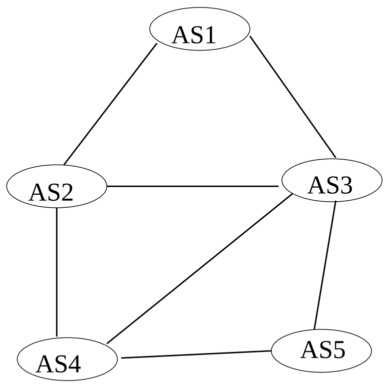

Figure 1: *An AS graph example*

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 regististry 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 admistrative 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 distance-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 the immediate neighbors via BGP sessions. The simplest distance-vector protocol would employ shortest-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*. Although routing policies are configured on each BGP speaking router, BGP speaking routers in an AS should share the same policy. For the simplicity of exposition, we assume that there is only one BGP speaking router in an AS and use the AS and its BGP speaking router interchangeably throughout this paper. The 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.

An AS uses import policies to transform incoming route updates. These import policies include denying an update, or permiting 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:

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

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, the AS picks the route with the smallest $next\_hop$.

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 contains a destination prefix, next-hop, med, local preference and AS path of the route. For the sake of simplicity, we describe the route 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 destinaton 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 cadidate route has AS path (1849 702 702 701 1), where AS 702 appears twice consecutively. This is resulted from the fact that 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 proprocessed so that no AS appears more than once thoughout this paper.

```
   Network          Next Hop            Metric LocPrf Weight Path
*> i4.2.24.0/21     134.24.127.3                            0 1740 1 i
*                   194.68.130.254                          0 5459 5413 1 i
*                   158.43.133.48                           0 1849 702 702 701 1 i
*                   193.0.0.242                             0 3333 286 1 i
*                   144.228.240.93                          0 1239 1 i
```

Routing policies are constrained by the commercial contractual agreements negotiated between administrative domain pairs. These contractual agreements include customer-provider, peering, and mutual transit. A customer pays its provider for connectivity to the rest of the Internet. 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 agreement is typically between two small administrative domains such as local ISPs or universities who are located close to each other and who can not afford additional Internet services for better connectivity. In addition, an AS can specify a diverse set of routing policies including its preference on route selection and filtering. The routing policies are 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, many ISPs are not 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 the distance map 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 the structural properties of the Internet. Further, AS relationships can help to effectively place measurement servers and better approximate the end-to-end distance.

3 AS Relationships and Routing Table Entry Patterns

Our algorithm for inferring AS relationships is based on the fact that ASes set up their export policies according to these relationships and on the resulting patterns on BGP routing table entries. In this section, we describe the annotated AS graph representation, the implied export policies, and the resulting patterns on routing table entries.

3.1 Annotated AS Graph and Selective Export Policies

The relationships between ASes arise from contracts that define the pricing model and the exchange of traffic between the administrative domains that the two ASes belong to. Two ASes that exchange traffic have a customer-to-provider, provider-to-customer, peer-to-peer, or sibling-to-sibling relationship. In a *customer-to-provider or provider-to-customer relationship*, the customer typically belongs to a smaller administrative domain that pays a larger administrative domain for access to the rest of the Internet. The provider is an AS that belongs to the larger administrative domain. In a *peer-to-peer relationship*, the two peers typically belong to administrative domains of comparable size and find it mutually advantageous to exchange traffic between their respective customers. Two ASes have a *sibling-to-sibling relationship* if they belong to the same administrative domain or to the administrative domains that have a mutual-transit agreement.

We propose to represent AS relationships by an annotated AS graph. Although there might be multiple BGP sessions between router pairs belonging to two ASes, the relationship between the two ASes should be uniquely defined since the routers within an AS share the same policies. 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. Furhtermore, 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 peer-to-peer relationship a *peer-to-peer edge* and the edge between two ASes that have a sibling-to-sibling relationship a *sibling-to-sibling edge*. Figure 2 shows an example of an annotated AS graph. An AS may have multiple customers, providers, peers and siblings. We define $customer(a)$, $peer(a)$, sibling(a), and $provider(a)$ as the set of customers, peers, siblings, and providers of a, respectively.

The AS relationships translate into the following rules that govern BGP export policies [23, 3]:

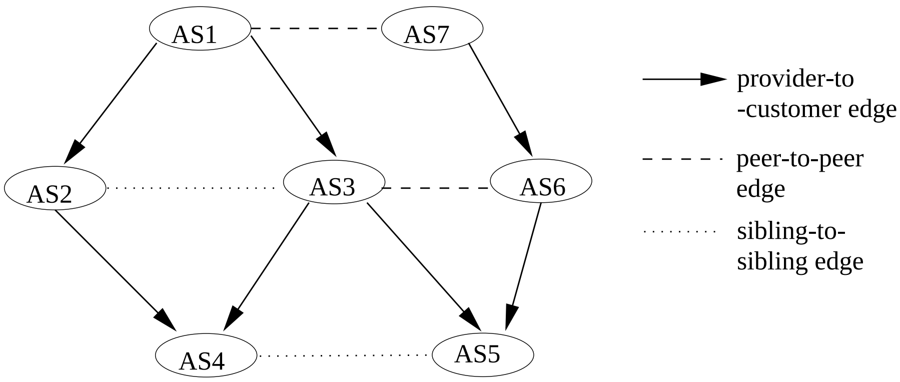

Figure 2: *An Anotated AS Graph*

- **Exporting to a provider:** In exchanging routing information with a provider, an AS can export its routes and the routes of its customers and siblings, but *can not export* routes learned from other providers or peers.
- **Exporting to a customer:** In exchanging routing information with a customer, an AS can export its routes and the route of its customers and siblings, as well as routes learned from its providers and peers.
- **Exporting to a peer:** In exchanging routing information with a peer, an AS can export its routes and the routes of its customers and siblings, but *can not export* the routes learned from other providers or peers.
- **Exporting to a sibling:** In exchanging routing information with a sibling, an AS can export its routes and routes of its customers and siblings, as well as routes learned from its providers and peers.

Note that although exporting policies are same for providers and peers (or customers and siblings), provider-customer relationships are asymetric and peer-peer (or sibling-sibling) relationships are symetric, which is the key in distinguish provider-customer relationships from peer-peer (or sibling-sibling) relationships. In summary, an AS selectively provides transit services for its neighboring ASes. Let $first(r.as\_path)$ denote the first AS in $r.as\_path$. An AS sets up its export policy according to the following rule.

Selective Export Rule : an AS does not provide transit services between any two of its providers and peers. Formally, consider AS u and AS $v \in provider(u) \cup peer(u)$. For each best route r of u,

if $first(r.as\_path) \in provider(u) \cup peer(u)$, then $export(v, u)[\{r\}] = \{\}$.

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 policies 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 $i < j < n$.

For example, in Figure 3, AS paths (1, 2, 3) and (1, 2, 6, 3) are valley-free while as_path (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 provides transit services between its two providers: AS 1 and AS 3. This violates the selective export rule. Formally, we have the following theorem.

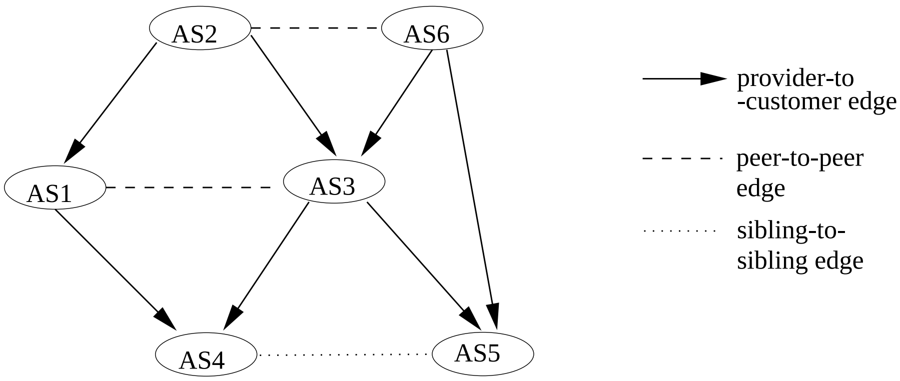

Figure 3: *AS paths (1, 2, 3) and (1, 2, 6, 3) are valley-free while AS paths (1, 4, 3) and (1, 4, 5, 3) are not valley-free.*

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.*

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_{j-1}, u_j) is either a provider-to-customer to sibling-to-sibling edge. From Lemma 3.1, we have $B(u_j, d).as\_path = (u_{j+1}, \ldots, u_n)$ and $export(u_{j-1}, u_j)[\{B(u_j, d)\}] \neq \{\}$. However, this contradicts the selective export policy rule that an AS does not provide transit services for two of its providers and peers, since both u_{j-1} and u_{j+1} are either a provider or a peer of u_j.

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-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 a BGP routing table RT, we derive an annotated AS graph G that is consistent with BGP routing table RT. In the next two sections, we present heuristic algorithms that use a BGP routing table to infer AS relationships and show experimental results derived from the Internet BGP routing tables.

4 Heuristic Algorithms for Inferring AS Relationships

In this section, we present heuristic algorithms for inferring AS relationships. 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 do not identify peer-to-peer edges from provider-to-customer and sibling-to-sibling edges. Now, we can infer that consecutive AS pairs on the left of the top provider are customer-to-provider or sibling-to-sibling edges, and consecutive pairs on the right of the top provider are provider-to-customer or sibling-to-sibling edges. 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 who can have peer-to-peer relationships in Section 4.2.

4.1 Algorithms for Inferring Provider-to-Customer and Sibling-to-Sibling Relationships

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

4.1.1 A 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 on the left of the top provider have a customer-to-provider or sibling-to-sibling relationship, and consecutive AS pairs on the right of the top provider have a provider-to-customer or sibling-to-sibling relationship. If an AS pair (u_1, u_2) is on the left of 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) is on the right of the top provider of an AS path, then u_1 provides transit services for u_2. Therefore, an AS pair (u_1, u_2) has a provider-to-customer (or customer-to-provider) relationship if u_1 (u_2) provides transit services for u_2 (u_1). An AS pair has a sibling-to-sibling relationship if the pair provides transit services for each other.

This leads to a three-phase heuristic algorithm for inferring provider-to-customer/peer-to-peer and sibling-to-sibling relationships. The first phase parses the routing table and figures out the degree of each AS. The second phase parses each entry of the routing table. It first identifies the top provider and then assigns consecutive AS pairs on the left of the top provider with a transient relationship and consecutive AS pairs on the right of the top provider with a transient relationship. The third phase finalizes the relationship between AS pairs. An AS pair has a sibling-to-sibling relationship if and only if the pair provides transit services for each other. An AS pair (u_1, u_2) has a provider-to-customer (or customer-to-provider) relationship if and only if u_1 (u_2) provides transit services for u_2 (u_1) and u_2 (u_1) does not provide transit services for u_1 (u_2). 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 table. As we will see later, we evaluate the algorithm by using the publicly available routing table, in 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 A Refined Algorithm

The Basic algorithm bases on the heuristic that the top provider has the highest degree among all ASes in the AS path. It is possible that the top provider of an AS path is not the AS that has the highest degree. Therefore, some route entries might infer the relationship of an AS pair incorrectly. To reduce the possibility of incorrect inference, we propose a refined algorithm that determines AS relationships by letting each routing table entry to vote on the relationship of an AS pair. In the refined algorithm, if a sibling-to-sibling relationship is concluded from only one entry, then we ignore that entry. One exception is that if a sibling-to-sibling relationship is inferred from only two entries, then we infer that the AS pair has a sibling-to-sibling relationship.

The refined algorithm infers AS relationships as follows. If all routing table entries agree that an AS pair has a provider-to-customer (or customer-to-provider) relationship, then the AS pair has the relationship. If only one routing table entry infers that an AS pair has a provider-to-customer (or customer-to-provider) relationship and more than one entry infer that the AS pair has a customer-to-provider (or provider-to-customer) relationship, then the AS pair has a customer-to-provider (provider-to-customer) relationship. For all other cases, the AS pair has a sibling-to-sibling relationship. Note that unlike the basic algorithm, the refined algorithm ignores some routing

Basic Algorithm:
Input: BGP routing table RT
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 RT,
    for each $i = 1, \ldots, n-1$,
        neigbhor$[u_i]$ = neigbhor$[u_i]$ $\cup \{u_{i+1}\}$
        neighbor$[u_{i+1}]$ = neigbhor$[u_{i+1}]$ $\cup \{u_i\}$
For each AS u,
    degree$[u]$ = |neighbor$[u]$|

Phase 2: Parse AS path to initialize consecutive AS pair relationship

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

Phase 3: Assign relationships to AS pairs

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

Figure 4: *Basic Heuristic Algorithm*

Refined Algorithm:
Input: BGP routing table RT
Output: Annotated AS graph G

Phase 1: Same as Phase 1 in the basic algorithm

Phase 2: Count the number of routes that infers an AS pair as having a provider-to-customer or customer-to-provider relationship

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

Phase 3: Assign relationships to AS pairs

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

Figure 5: *Refined Heuristic Algorithm*

table entries. Figure 5 shows the refined algorithm in details.

4.2 A Heuristic Algorithm for Inferring Peer-to-Peer Relationships

Both the Basic and Refined algorithms classify AS relationships into provider-to-customer or sibling-to-sibling relationships. In this section, we present a heuristic algorithm for identifying peer-to-peer relationships. According to Corollary 3.1, a peer-to-peer edge can be only between a top provider and one of its neighboring ASes in an AS path. Therefore, if an AS pair appears in the maximal uphill (or downhill) path and neither of the AS pair is a top provider, then the AS pair can not have a peer-to-peer relationship. Therefore, the top provider can have a peer-to-peer relationship with its left or right AS in the AS path. Furthermore, an AS path has at most one consecutive AS pair that has a peer-to-peer relationship. Therefore, we assign a peer-to-peer relationship between the top provider and its neighboring AS in the AS path as follows. If the top provider has a sibling-to-sibling relatioship with its one of the neighbors, then the top provider has a peer-to-peer relationship with the other neighbor. Otherwise, since a peer-to-peer relationship is typically between ASes of comparable sizes, we use the heuristic that a peer-to-peer edge is between the top provider and its neighboring AS (in the AS path) that has a higher degree. Moreover, we use the heuristic that the degrees of two peers do not differ significantly. Specifically,

Final Algorithm:
Input: BGP routing table RT
Output: Annotated AS graph G

Phase 1: Use either basic or refined algorithm to coarsely classify AS pairs into having provider-to-customer or sibling-to-sibling relationships

Phase 2: Identify AS pairs that can not have a peer-to-peer relationship

For each AS path $(u_1, u_2, \ldots, u_n)$ in RT,
    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 relationship$[u_{j-1}, u_j]$$\neq$sibling-to-sibling and relationship$[u_j, u_{j+1}]$$\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 peer-to-peer relationships to AS pairs

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

Figure 6: *Final Algorithm*

we assume that the degrees of the two ASes that have a peer-to-peer relationship do not differ by more than R times, where R is a fixed number that depend on the current state of the Internet.

The Final algorithm infers peer-to-peer relationships as follows. Phase 1 coarsely classifies AS pairs into having provider-to-customer or sibling-to-sibling relationships. Phase 2 identifies all AS pairs that can not have peer-to-peer relationships. Finally, Phase 3 infers peer-to-peer relationships from the rest of connected AS pairs as long as the pair degrees do not differ by more than R times. Figure 6 shows the Final algorithm in details.

5 Inferring AS Relationships in the Internet

In this section, we present experimental results of inferring AS relationships in Internet. We use the publicly available routing table from the Route Viewer router in Oregon [16], which has the most complete view of the current Internet. The BGP routing table in the Router Viewer router closely captures the current state of the Internet by establishing BGP peering sessions with 22 ISPs at 24 locations as shown in Table 1. For the detailed description of the Route Viewer server, see [16]. The Route Viewer server collects the BGP routing table once every night [20].

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*

	Total Routing Entries	Total edges	Sibling-to-sibling edges inferred by Basic (Percentage)	Sibling-to-sibling edges inferred by Refined (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*

5.1 Experimental Results

We implement the Basic, Refined, and Final algorithms using the perl programming language. For the Final algorithm, we use the Basic algorithm for inferring sibling-to-sibling relationships and let the limit of the ratio between the degrees of two peer ASes, R, be infinite or 60. We will see in the next subsection that fine-tuning R to be 60 can significantly improve the inference result. We run the algorithms for the BGP routing tables from 1999/9/27, 2000/1/2 and 2000/3/9. The total number of edges in the AS graph, sibling-to-sibling edges inferred by both Basic and Refined, and the total number of peer-to-peer edges inferred by Final 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 twice 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. In our perl programs, we eliminate self edges.

These BGP routing tables contain almost 1 million route 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 algoirthm 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 algoirthm 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 algoirthm 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 number or the percentage of sibling-to-sibling relationship 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	99.8%
	Peer	0.2%
Peer	Peer	76.5%
	Customer	23.5%
Sibling	Sibling	20%
	Peer	60%
	Customer	20%
Nonexistent	Customer	95.6%
	Peer	4.4%

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. We compare our data for March 9, 2000 with that of AT&T Common IP Backbone in the same day.

Table 3 compares the inference results from Basic and Final($R = \infty$) algorithms with AT&T internal information. From the table, we see that 99.8% of our inferred customers are confirmed by the AT&T internal information. 76.5% 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 our inference results, 96.3% of inference results are confirmed by the AT&T internal information. Note we do not necessarily have all AT&T's neighbors from the routing table in Route Viewer since AT&T announces only its best route 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 adjacence list and most of these ASes are customers of AT&T.

Table 4 compares the inference results from Refined and Final($R = \infty$) algorithms with AT&T internal information. From the table, we see that 99.5% of our inferred customers are confirmed by the AT&T internal information. 76.5% 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, 96.3% of inference results are confirmed by the AT&T internal information. Using Refined algorithm, we improve the inference results on sibling-to-sibling relationships.

Table 5 compares the inference results from Basic and Final($R = 60$) algorithms with AT&T internal information. From the table, we see that 99.8% 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 our inference results, 99% 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 peer-to-peer relationships significantly.

Our Inference	AT&T Information	Percentage of ASes
Customer	Customer	99.5%
	Peer	0.5%
Peer	Peer	76.5%
	Customer	23.5%
Sibling	Sibling	25%
	Peer	50%
	Customer	25%
Nonexistent	Customer	95.6%
	Peer	4.4%

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

Our Inference	AT&T Information	Percentage of ASes
Customer	Customer	99.8%
	Peer	0.2%
Peer	Peer	100%
Sibling	Sibling	20%
	Peer	60%
	Customer	20%
Nonexistent	Customer	95.6%
	Peer	4.4%

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

5.3 Verifications by the WHOIS service

We verify our inferred sibling-to-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-to-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-to-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 query the WHOIS lookup service and confirm 101 of 186 inferred sibling-to-sibling relationships for the more recent data, i.e., 2000/1/2 data. This is more than 50% of the inferred sibling-to-sibling relationships. Note that other unconfirmed sibling-to-sibling relationships might still have sibling-to-sibling relationships since the WHOIS lookup service might be out of date and we do not have sufficient information about ISP mergers.

The Refined algorithm reduces the number of sibling-to-sibling relationships by ignoring some of the 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-to-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-to-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-to-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 MCI (AS701). According to the AS graph, MCI 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 MCI (AS701) instead of using the direct link to MCI (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 since AS5727 (AT&T) is likely to be the top provider. This scenario is resulted from the fact that AS3333 is an European ISP that has many customers.

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 we can ignore some routing table entries. However, it can not be done without the knowledge of the ISPs that each AS belongs to. Therefore, it is a challenging task to increase the accuracy of the AS relationship inference with only BGP routing tables.

6 Applications of AS Relationships

The knowledge of global inter-AS relationships have many applications. We discuss a few applications in this section. First, AS relationships can help to understand the reachability between ASes. Therefore, it can improve the construction of the distance map and the placement of the proxy or mirror site servers. Previous studies in the construction of distance map in the Internet assume that two hosts can reach each other via any intermediate routers [7, 8, 10]. This can potentially cause the miscalculation of the distance map. Proper placement of proxy or mirror site servers becomes essential to many Internet services. Given the AS relationships, we can construct an accurate distance map and thereby effectively placing proxy and mirror site servers.

Second, it can help ISPs or domain administrators to achieve load balancing, congestion avoidance and fault tolerance goal. Typically, an ISP operator might have a good understanding of the relationship between his/her own AS and its neighbors but not the relationships between other AS pairs. Given the AS relationships, an ISP operator can infer the potential choices of links used for both incoming and outgoing traffic and can effectively set up routing policies to balance load, achieve fault tolerance goal, and avoid congestion on inter-AS links.

Third, it can help ISPs or companies to plan for future contractual agreements. The contractual agreement between ISPs are constantly evolving. For example, ISP A is a customer of ISP B. As ISP A becomes larger, ISPs A and B might agree to privately peer with each other. Identifying private-peering candidates involves the knowledge of relationships between ASes. Furthermore, a customer can better select ISP(s) and an ISP can better predict the traffic pattern for a potential new customer if he/she has a picture of AS relationships.

Fourth, it can help ISPs to reduce the effect of the misconfiguration and to debug router configuration files [13, 15]. Route policies are manually configured and therefore prone to errors. Such errors can propagate further to other ASes and can potentially cause outage. Therefore, it important for ISPs to monitor the received route announcements using the 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 the routing table periodically to identify potential erroneous routes and inform the ISP of the originating AS.

Fifth, it can potentially avoid route divergence problem. Since BGP itself does not constraint routing policies to ensure route convergence, some policy configuration can lead to route oscillation and/or divergence [24, 25]. Routes can be ensured to converge given the hierarchical structures of AS relationships and a set of simple guidelines for setting up policies in each AS [26]. Even if some isolated subset of ASes AS violate the hierarchical structure (e.g., during the transition period that two ISPs are merged into one), it is important to identify the sets of ASes to coordinate their policy setting.

Sixth, understanding AS relationships can verify the consistency of information in Internet Routing Registry (IRR). Since an ISP does not necessarily keep its registry data up-to-date, it is important to have a mechanism to check the consistency if the registry is used for important issues such as route convergence or fault detection.

7 Conclusions

Interdomain routing policies are constrained by the commercial contractual relationships between administrative domains. As a result, AS relationships are an inherent aspect of the Internet routing structure. Understanding AS relationships can help to accurately estimate end-to-end distance and to properly place proxy and mirror site servers. 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 customer and two peers are typically of comparable size. We perform an experimental study of AS relationships in the Internet. Our heuristic algorithm classifies more than 90.5% of connected AS pairs into provider-to-customer relationships, less than 1.5% of connected AS pairs into sibling-to-sibling relationship, and less than 8% of connected AS pairs into peer-to-peer relationships. We verify our inferred relationships with both AT&T internal information and the WHOIS lookup service. 99.8% 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-to-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. We show that there are many applications of inferred AS relationships.

Acknowledgments. 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 information on the AT&T view.

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.