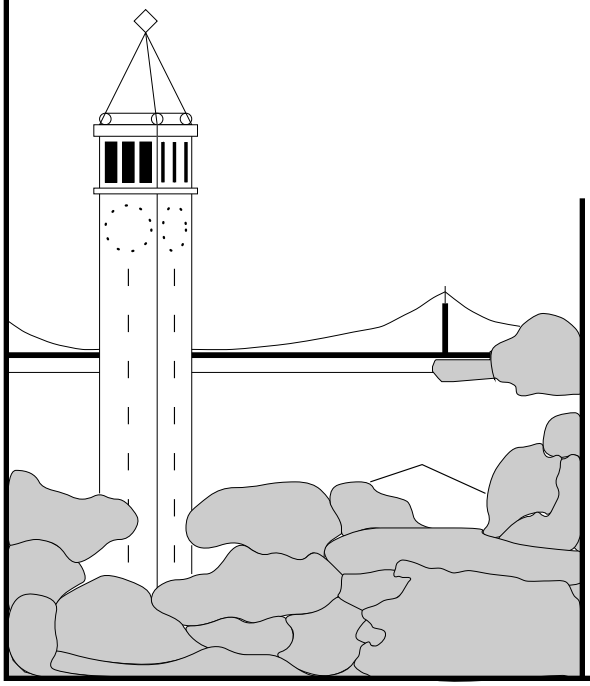
Characterizing the Internet Hierarchy from Multiple Vantage Points

Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, Randy H. Katz



Report No. UCB/CSD-1-1151

August 2001

Computer Science Division (EECS)
University of California
Berkeley, California 94720

This work was supported by DARPA Contract No. N00014-99-C-0322.

Abstract— **The delivery of IP traffic through the Internet depends on the complex interactions between thousands of Autonomous Systems (ASes) that exchange routing information using the Border Gateway Protocol (BGP). This paper investigates the topological structure of the Internet in terms of customer-provider and peer-peer relationships between ASes, as manifested in BGP routing policies. We describe a technique for inferring AS relationships by exploiting partial views of the AS graph available from different vantage points. Next we apply the technique to a collection of ten BGP routing tables to infer the relationships between neighboring ASes. Based on these results, we analyze the hierarchical structure of the Internet and propose a five-level classification of ASes. Our analysis differs from previous characterization studies by focusing on the commercial relationships between ASes rather than simply the connectivity between the nodes.**

I. Introduction

Today's Internet is divided into more than 10,000 Autonomous Systems (ASes) that interact to coordinate the delivery of IP traffic. An AS typically falls under the administrative control of a single institution, such as a university, company, or Internet Service Provider (ISP). Neighboring ASes use the Border Gateway Protocol (BGP) [1], [2] to exchange information about how to reach individual blocks of destination IP addresses (or, prefixes). An AS applies local policies to select the best route for each prefix and to decide whether to propagate this route to neighboring ASes, without divulging these policies or the AS's internal topology to others. In practice, BGP policies reflect the commercial relationships between neighboring ASes. AS pairs typically have a customer-provider or peer-peer relationship [3], [4]. A provider provides connectivity to the Internet as a service to its customers, whereas peers provide connectivity between their respective customers.

AS relationships, and the associated routing policies, have a significant impact on how traffic flows through the Internet. An understanding of the structure of the Internet in terms of these relationships facilitates a wide range of important applications. For example, consider a content distribution company that has a choice of placing replicas of a Web site in data centers hosted by different ASes. The company can identify the IP prefixes and ASes responsible for a large portion of the traffic from the site [5]. With an accurate view of the connectivity and relationship between ASes, the company can identify the best locations for its replicas. As another example, consider a new regional ISP that wants to connect to a small number of upstream providers. An accurate view of the AS topology and relationships between ASes can help the ISP determine which ASes would provide the best connectivity to and from the rest of the Internet. The Internet topology alone does not provide enough information to answer these questions. For example, suppose that AS B connects to two providers, AS A and AS C. An AS graph would show connectivity from A to B and from B to C; however, AS B's routing policies would not permit transit traffic between A and C.

In the absence of a global registry, the AS-level structure of the Internet is typically inferred from analysis of routing data. Previous work has focused on constructing a view of the AS graph from traceroute experiments or individual BGP table dumps. Traceroute provides a view of the path from a source to a destination host at the IP-level. The traceroute data must be analyzed to infer which interfaces belong to the same router and which routers belong to the same AS [6]. Running experiments between multiple source-destination pairs provides a larger collection of paths over time [6], [7], [8]. Other studies have extracted AS paths directly from BGP routing tables or BGP update messages [9], [10]. The routing table dump from the University of Oregon RouteViews server [11], [12] has been the basis of several studies of basic topological properties, such as the distribution of node degrees [13], [14]. With the exception of the work in [10], these studies have focused on the topological structure without considering the relationship between neighboring ASes. [10] presents a heuristic for inferring the relationships from a collection of AS paths and evaluates the technique on the RouteViews data.

In this paper, we propose a technique for combining data from multiple vantage points in the Internet to construct a more complete view of the topology and the AS relationships. Each vantage point offers a partial view of the Internet topology as viewed from the source node. Due to the presence of complex routing policies, these partial views are not necessarily shortest-path trees and may, in fact, include cycles. We generate a directed AS-level graph from each vantage point and assign a *rank* to each AS based on its position. Then, each AS is represented by the vector that contains its rank from each of the routing table dumps. Finally, we infer the relationship between two ASes by comparing their vectors. The work we describe in this paper is novel in two ways. First, we analyze AS paths seen from multiple locations to form a more complete view of the graph. Second, rather than simply combining the data from the various vantage points, we propose a methodology for exploiting the uniqueness of each view to infer the relationships between AS pairs.

We evaluate our technique on a collection of ten BGP routing tables and summarize the characteristics of the AS relationships. To validate the inferences, we check for paths that are not consistent with the routing policy as-

sumptions underlying customer-provider and peer-peer relationships. We show that these cases account for a small proportion of the paths and that the most common inconsistencies may stem from misconfiguration or more complex AS relationships. Then, we analyze the resulting AS graph to characterize the hierarchical structure of the Internet. We present a five-level classification of ASes with a top-most layer that consists of a rich set of peer-peer relationships between 20 so-called *tier-1* providers. This classification can aid an institution in selecting or reevaluating its connections to other ASes in the Internet.

II. Problem Formulation

In this section, we formulate the problem we are trying to solve. We first present a brief overview of AS relationships and their implications on BGP export policies. Then we formally define the Type of Relationship (ToR) problem for finding an assignment of AS relationships that maximizes the number of paths that adhere to the export restrictions.

A. AS Relationships and BGP Export Policies

The relationships between ASes arise from contracts that define the pricing model and the exchange of traffic between domains. ASes typically have a *provider-customer* or *peer-peer* relationship [3], [4]. In a provider-customer relationship, the customer is typically a smaller AS that pays a larger AS for access to the rest of the Internet. The provider may, in turn, be a customer of an even larger AS. In a peer-to-peer relationship, the two peers are typically of comparable size and find it mutually advantageous to exchange traffic between their respective customers. These relationships translate directly into policies for exporting route advertisements via BGP sessions with neighboring ASes:

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

Each BGP session defines a relationship between the two ASes it connects. Although two ASes may have multiple BGP sessions, the relationship between the two ASes should be uniquely defined.

Although ASes typically follow these guidelines, some ASes have more complicated relationships in practice. For example, two ASes operated by the same institution may have a sibling relationship where each AS provides transit service for the other [10]. Other AS pairs may have backup relationships to provide connectivity in the event of a failure [15]. Alternatively, two ASes may peer indirectly through a transit AS [16]. Also, an AS pair may have different relationships for certain blocks of IP addresses; for example, an AS in Europe may be a customer of an AS in the United States for some destinations and a peer for others. Router misconfiguration may also cause violations in the export rules. For example, a customer may mistakenly export advertisements learned from one provider to another. We initially assume that only a small fraction of the AS pairs represent exceptions to the traditional provider-customer and peer-peer relationships. Our inference technique is designed to tolerate occasional exceptions and, in fact, our algorithm can be used to identify AS pairs that have unusual relationships.

B. Type-of-Relationship (ToR) Problem

BGP export policies have a direct influence on the AS paths seen from a particular vantage point in the Internet. If every AS adheres to the customer and provider export rules, then no path would ever traverse a customer-provider edge after traversing a provider-customer or peer-peer edge, and no path would ever traverse more than one peer-peer edge [10], [15]. To formulate these properties in mathematical terms, we denote an edge from a customer to a provider with a -1, an edge from one peer to another with a 0, and edge from a provider to a customer with a $+1$. Restating a result from [10] in these terms, we have:

Theorem 1: If every AS obeys the customer, peer, and provider export policies, then every advertised path belongs to one of these two types for some $M, N \geq 0$:

1. *Type 1*: $-1 \ldots$ N times $+1 \ldots$ M times.
2. *Type 2*: $-1 \ldots$ N times 0 $+1 \ldots$ M times.

The first stage of a Type 1 path contains only customer-provider links (*uphill* portion) and the second stage contains only provider-customer links (*downhill* portion). The second type captures all paths which traverse exactly one peering link. The single peering link must appear in between the uphill and the downhill portions of any path.

The type-of-relationship (ToR) problem can be formulated as a graph theory optimization problem for labeling the edges of the graph with a -1, 0, or 1 such that the observed paths obey the export policies implied by the relationships. Given a graph G with each edge labeled as -1, 0 or $+1$, a path p is said to be *valid* if it is either of

Type 1 or Type 2 :

ToR Problem: Given an undirected graph G with vertex set V and edge set E and a set of paths P, label the edges in E as either -1, 0 or $+1$ to maximize the number of *valid* paths in P.

The graph G represents the entire Internet topology where each node is an AS and each undirected edge represents a relationship between the incident pair of ASes. The set of paths P consists of all paths seen from the various vantage points. We believe that this problem is NP-complete. However, we have not been able to prove that this problem is NP-complete nor are we aware of any theoretical work that provides a polynomial-time solution.

In previous work, Lixin Gao [10] proposes and evaluates a heuristic for inferring AS relationships from a collection of AS paths. For each AS path, the heuristic uses the degree of the nodes to identify the point that marks the boundary between the uphill and downhill portions of the path. The inferences from multiple paths are later combined to infer the relationship between the ASes. In the next section, we propose a heuristic for solving the ToR problem based on the paths seen from each vantage point.

III. Inferring AS Relationships

This section presents our algorithm for inferring AS relationships. We describe the properties of the partial AS graph observed from a single vantage point. This motivates our algorithm for assigning a rank to each AS for each of the partial views. Finally, we describe how to infer the relationships between ASes based on their ranks in the different views.

A. Partial View of the AS Graph

Routing data from a single vantage point provides a set of paths from a particular source node. These paths can be used to construct a directed graph that includes the edge (u, v) if one or more paths travel directly from AS u to AS v. If every AS employed a simple shortest-path routing policy, then this graph would be a shortest-path tree rooted at the source. However, complex routing policies result in more complicated graphs that may include non-minimal paths and even cycles. The graph from a single vantage point reveals a great deal of information about the relationship between ASes. Identifying the boundary point between the uphill and downhill portions of the paths is the key to inferring the AS relationships. The uphill portion of a route appears at the beginning of a path, near the source node, whereas the downhill portion appears in the later portion of the path. As such, a leaf node in the graph is likely to be a customer of its parent node(s). We exploit this property by successively pruning the leaf nodes and assigning ranks to ASes as we prune.

Still, identifying the boundary point between the uphill and downhill portions of a path is tricky. The structure of the partial view of the AS graph depends on the position of the AS in the Internet hierarchy. When viewed from a tier-1 AS that does not have any upstream providers, every path consists of zero or one peer-peer edges followed by a downhill portion. In practice, we expect the provider-customer relationship to be acyclic [16]. That is, if u is a customer of v and v is a customer of w, then w is not a customer of u. Hence, the partial view from a tier-1 AS would tend to be acyclic. In this case, successive pruning would identify provider-customer relationships. However, in other scenarios, the graph may have cycles. For example, suppose source X has two paths $p_1 = (X, A, B, C)$ and $p_2 = (X, B, A, D)$. The resulting graph has a cycle between nodes A and B. As such, it is difficult to infer the relationships between X, A, and B. We exploit this observation in our algorithm by assigning the same rank to all of the ASes in the connected component of the graph. Information from other vantage points is necessary to construct an inference for these ASes.

In practice, the Internet consists of a relatively small number of large Internet Service Providers (ISPs) and a large number of smaller ASes. A small AS must traverse one or more upstream providers to reach most of the many other small ASes. As such, a large portion of the paths in a graph should consist mostly of roughly equal downhill and uphill portions of non-zero lengths. Thus, we expect a large portion of the edges in the graph to fall in a large, acyclic portion consisting of provider-customer edges. The remaining edges should fall into a connected component near the source node. Our heuristic exploits this property by making a loose association of AS rank with the provider-customer relationship and using probabilistic comparisons to resolve incorrect inferences.

B. AS Ranking

Our algorithm assigns a rank to each AS for each vantage point. Let X denote the source AS of a particular view of the AS graph and let $P(X)$ denote the set of AS paths seen from X. Each path $p \in P(X)$ consists of a sequence of nodes, starting with X. We construct a directed graph G_X that consists of each edge that appears in one or more of the paths in $P(X)$. We let $v(G_X)$ denote the set of all vertices in G_X and let $leaves(G_X) \in v(G_X)$ denote the leaves of the graph. For a given $v' \subset v(G)$, $G_{v'}$ is the subgraph of G induced by the vertices in v'. Drawing on this notation, we assign a ranking $rank(u)$ to each vertex $u \in v(G_X)$ by applying the reverse pruning algorithm in

```
G = G_X;
r = 1;
while (leaves(G) ≠ φ) {
      for all u ∈ leaves(G)
            rank(u) = r;
      v' = v(G) − leaves(G);
      r = r + 1;
      G = G_{v'};
}
for all u ∈ v(G)
    set rank(u) = r;
```

Fig. 1. Reverse pruning algorithm on graph G_X

Figure 1. At each stage, the algorithm identifies the leaf nodes, assigns them a rank, and removes these nodes (and their incident edges) from the graph. In the end, the remaining nodes (if any) form the connected component of the original graph G_X; these nodes are all assigned the same (highest) rank.

The algorithm assigns a rank to each AS. Comparisons between AS rankings play a major role in our inference algorithm in the next subsection. However, many AS pairs do not share an edge in the partial view. In many cases, the two ASes may not have an edge in *any* of partial views because they are not connected to each other in the real graph. In this scenario, the rank does not have any particular meaning. In other cases, the two ASes may share a link in one of the other partial views. In this scenario, our algorithm imposes a relative rank for these two ASes even though they may not share an edge from source X's perspective. For example, consider a source X with paths (X, A, C, D) and (X, B, E, F) that do not use the edge (C, E). Our algorithm assigns a rank of 1 to nodes D and F, a rank of 2 to nodes C and E, and a rank of 3 to nodes A and B. Despite the fact that the edge (C, E) does not appear in G_X, we may be able to exploit the presence of both nodes in $v(G_X)$ in conjunction with the ranking from other vantage points that do include the edge to draw inferences about the relationship between C and E.

C. Inference Rules for the ToR problem

The routing data from each vantage point provides a partial view of the Internet. Given data from N vantage points, we map each AS into an N-dimensional vector $c(i) = (r_{i1}, \ldots, r_{iN})$, where r_{ij} is the rank of AS i from vantage point j. Let $l(i, j)$ refer to the number of coordinates where $r_{ik} > r_{jk}$ and $e(i, j)$ be the number of coordinates where $r_{ik} = r_{jk}$, for all $k = 1, 2, \ldots, N$.

C.1 Complete Dominance

In a view from source X, if AS i has a higher rank than AS j, then i appears to be a provider of j. In complete dominance, we assert that if the rank of i is more than that of j irrespective of the vantage point, then i is definitely a provider of j. A vector $c(i)$ is said to dominate $c(j)$, if $l(i, j) > 0$ and $l(j, i) = 0$. So, in vector terminology, if $c(i)$ dominates $c(j)$, then we can infer that i is the provider of j, assuming that the two ASes share an edge.

C.2 Equivalence

Two ASes are said to be equivalent if $e(x, y)$ >N/2. This rule states that from more than 50% of the vantage points, two ASes x and y appear in the same level of the hierarchy. Two ASes that appear in the same level in the hierarchy from different vantage points are likely to be peers. This rule is useful in finding peers among tier-1 and tier-2 providers.

C.3 Clustering

Most of the partial views generated from our routing table dumps are directed acyclic graphs. As a result, all ASes in these graphs are removed at some stage of the reverse pruning process. Hence if (i, j) is an edge in the partial view from a source, then j is removed before i. This implies that the $rank(i) > rank(j)$. Therefore we can infer that if i is a provider of j, then with high probability $|rank(i) - rank(j)| \geq 1$ for every AS. Note that if the edge is viewed from j or its customers, then $rank(j) > rank(i)$. Given this constraint, the Euclidean distance between $c(i)$ and $c(j)$ is at least $sqrt(N)$. However, it is hard to infer from this that if the distance between $c(i)$ and $c(j)$ is more than $sqrt(N)$ then they have a provider-customer relationship. This is because, i and j can be peers and still have a few coordinates where their ranks have a large difference. We observe this for some European ISPs that peer with American ISPs. However, one can infer the opposite of this rule. If the distance between i and j is strictly less than $sqrt(N)$, then they are more likely to be peers. This rule clusters these N-dimension vectors into spheres of radius $sqrt(N)$ to identify possible peers.

C.4 Probabilistic Rules

We introduce two probabilistic rules to tolerate uncertainty in in export policies and our ranking mechanism. *Probabilistic Dominance* states that if $l(i, j)/l(j, i) > \delta_0$ for a high value of δ_0 then i is a provider of j. Typically, in graphs from the vantage point of j or its customers, it is probable that $rank(j) > rank(i)$ even if i is a provider of

j. To avoid an incorrect inference, we introduce the rule of probabilistic dominance. *Probabilistic Equivalence* occurs when $l(i,j)/l(j,i) \leq \delta_1$ for δ_1 close to 1. We use this rule to infer peering relationships between ASes which are not in the same level in the hierarchy and those cases where the relationship between two ASes is not visible from many partial views. An AS relationship may not be visible from a partial view because ASes may assign a low preference to paths that traverse this edge. Using probabilistic equivalence, we test whether two ASes are peers. We use values of 3 and 2 for δ_0 and δ_1 respectively.

C.5 Order of Application

The relationship inferences depend on the order in which we apply these rules. We treat equivalence and dominance as the basic rules for inferring peer-peer and provider-customer relationships. We apply equivalence before checking for dominance. Since we apply dominance after the equivalence rule, if (i,j) is inferred as a provider-customer relationship using the dominance rule, then the rank of i should be more than the rank of j in at least $N/2$ of the dumps. If this is not the case, the rank of i should be equal to that of j in at least $N/2$ dumps thereby classifying the link as a peer-peer using the equivalence rule. Therefore those provider-customer relationships inferred using the dominance and equivalence rules can be treated with a high level of confidence. We apply the clustering condition before applying the probabilistic rules. The dominance, equivalence, and clustering conditions are powerful constraints for determining the type of a relationship. The probabilistic rules are applied to eliminate the AS relationships that cannot be inferred from the more basic conditions.

IV. Experimental Results

This section evaluates our inference techniques on a collection of ten publicly-available BGP routing tables. We classify the relationships between ASes and identify a small number of AS paths that are inconsistent with the relationship assignment. The most common anomalies seem to stem from recent acquisitions and mergers, suggesting that some AS pairs may have a sibling relationship.

A. BGP Routing Table Data

Our inference techniques have been applied to a collection of ten BGP routing tables available from Telnet Looking Glass servers. We automated the process of contacting each server, sending "`show ip bgp`" to the command-line interface, and storing the resulting table. For each destination prefix the table has one or more routes with a variety of BGP attributes, including the AS path. We extract the best and alternate paths for each prefix and construct a list of all AS paths that appear in the table. For each path, we add the AS number of the router to the beginning of each path and remove duplicate AS numbers that arise from AS prepending. Then we process the paths to construct a partial view of the AS graph. After constructing the partial views, we apply the ranking algorithm and inference rules from Section III to assign a relationship to each AS pair that shares an edge in one or more of the routing tables.

TABLE I
Telnet Looking Glass Servers

AS #	Name	# Edges	Change
1	Genuity	13419	+1.8%
1740	CERFnet	14287	n/a
3549	Globalcrossing	13542	+1.0%
3582	University of Oregon	23136	-0.4%
3967	Exodus Comm.	19005	+0.4%
4197	Global Online Japan	13474	+1.0%
5388	Energis Squared	13534	+2.1%
7018	AT&T	14160	+3.0%
8220	COLT Internet	11282	n/a
8709	Exodus, Europe	15519	+0.7%

Table I provides a summary of the ten tables we downloaded on April 18, 2001. The "# Edges" column shows the number of unidirectional edges in the AS paths. The "Change" column indicates the change in the number of edges from April 18 to May 1, when we downloaded a new copy of the tables. The entry for AS 3582 corresponds to the University of Oregon RouteViews server, which has 52 peering sessions with 39 different ASes [11]. The RouteViews server has an especially rich view of the AS graph, with over 23,000 edges compared to 11,000–15,000 edges for most of the other routing tables. In total, the AS paths in the ten routing tables have 24,752 unidirectional edges between 24,059 pairs of ASes. More than 25% of the edges appear in all ten routing tables, as shown in Figure 2, which plots a histogram of the percent of edges that appear in x of the 10 routing tables. More than 80% of the edges appear in at least two dumps.

We use the partial views from these ten routing tables to generate our inferences of the AS relationships. In Section IV-C, we validate our inferences using the AS paths from another collection of routing tables. We manually downloaded routing data from Ebone (AS #1755), MAE-West (AS #2548), KDDI Japan (AS #2516), and Cable and Wireless (AS #6893) on April 9, 2001. These four tables are available from Web Looking Glass servers that have a slightly different interface than the Telnet servers.

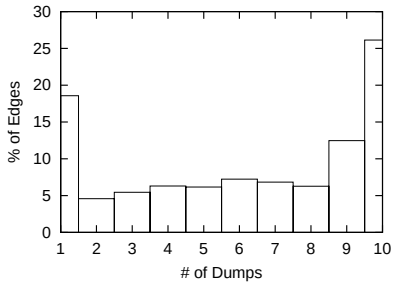


Fig. 2. Percentage of edges that appear in x of the ten tables

TABLE II
INFERRED RELATIONSHIPS FOR 24,059 AS PAIRS

Relationship	# AS pairs	Percentage
Provider-customer	22,712	94.40%
Peer-peer	1,241	5.16%
Unknown	106	0.44%

The Web interface typically does not permit users to invoke the "`show ip bgp`" command. Instead, we rely on the "`bgp paths`" command that produces a list of AS paths, without the destination prefix or an indication of the best path. As with the "`show ip bgp`" data, we extract the AS paths, add the AS number of the source AS, and remove duplicate ASes. Then, we use the results of our inference algorithm to assign a relationship to each unidirectional edge in each path and look for paths that violate the two patterns identified in Section II.

B. Relationship Inferences

Table II summarizes the results from applying our inference algorithm to the ten BGP tables from Table I. Our algorithm produces an inference for over 99.5% of the edges in our AS graph (23,953 of the 24,059 AS pairs). The vast majority of AS pairs appear to have a provider-customer relationship. Approximately 5% of the AS pairs have a peer-peer relationships. Table III highlights the role of the various inference rules in drawing conclusions about AS relationships. A large percentage of the provider-customer relationships are inferred from the complete dominance rule. Complete dominance in N dimensions is a good indication of a provider-customer relationship and we can be reasonably certain of 97.93% of our provider-customer inferences. Similarly, close to 80% of the peering links are inferred from the equivalence and clustering conditions. The probabilistic rules account for 2.1% of the provider-customer inferences and 22.4% of the peer-peer inferences.

The percentages of provider-customer and peer-peer relationships in Table II are consistent with the conclusions of Lixin Gao in [10]. Our inference that 5.2% of the AS pairs (1,241 pairs) are peers is close to Gao's values between 5.3% and 7.8%. The percentage of provider-customer relationships we infer is within 1–1.5% of the figure reported in [10]. Her study drew on RouteViews data from September 1999, January 2000, and March 2000. The number of edges in the RouteViews dump has grown by over 70% over the last 13 months. With the larger RouteViews table and the nine other tables, our collection of edges is twice as large as the graph used in her earlier study. Using traceroutes from 16 sources to 400,000 destinations [8] in October 2000, CAIDA constructed an AS graph that is slightly larger than ours. Their final graph consists of 7,563 ASes and 25,005 edges. Ours contains 10,698 ASes and 24,752 edges. However, they do not explore this graph in terms of AS relationships.

TABLE III
DISTRIBUTION OF THE 23,953 INFERENCES

Rule	Number	Percentage
Complete dominance	22,241	97.93%
Probabilistic dominance	471	2.07%
Equivalence	836	67.37%
Probabilistic equivalence	278	22.40%
Clustering	127	10.23%

C. Validation of Inferences

Since the peering and customer information of an ISP are proprietary information, we cannot validate our inferences against an official list of AS relationships. Instead, we determine what percentage of the AS paths actually adhere to the export rules suggested by our inferences. There are two scenarios where we may label an AS path as an anomaly: one in which some AS in the path actually violates the export rules or the other in which our relationship inference of one of the edges in the path is wrong. The percentage error that is reported in this section is the sum total of these two scenarios. For our validation, we draw on the list of AS paths from two of the ten of the Telnet Looking Glass Servers (AS numbers 1 and 7018) used to construct our original inferences, as well as the four Web Looking Glass Servers (AS numbers 1755, 2516, 2548, and 6893).

If every AS pair has a customer-provider or peer-peer relationships, then every AS path should have have one of the two patterns identified in Theorem 1. A path is an anomaly if it has any two adjacent edges having one of the following patterns:

1. $(+1\ -1)$: An AS permits transit traffic between two of its providers.
2. $(+1\ 0)$: An AS permits transit traffic from one of its

peers to one of its providers.
3. $(0\ -1)$: An AS permits transit traffic from one of its providers to one of its peers.
4. $(0\ 0)$: An AS permits transit traffic between two of its peers.
Case 1 represents a serious violation of the export rules. This anomaly may arise from a misconfigured customer or or due to a misclassified relationship where the customer AS is actually a sibling of one or both of the providers. Backup and sibling relationships relationships can cause case 2 and case 3 anomalies. Case 4 suggests that the path traverses two consecutive peering links, which may be permissible if two the peers have a sibling relationship for some destination prefixes. Detecting these anomaly paths provides a way to identify AS pairs that may have more complicated relationships.

As shown in Table IV, the vast majority of paths are consistent with the relationship assignments and the associated export policies. The percentage of anomalies varies between 2–3.4% for five of the six routing tables. These results validate our base assumption that the export rules are observed by a large percentage of the ASes. However, KDDI (AS #2516) has a relatively high percentage of anomalous paths (8.5%). For every anomalous path, we can identify an anomaly pattern consisting of three adjacent ASes (A, B, C) where the pair of edges (A, B) and (B, C) falls into one of four cases. The results in Table IV show that case 3 anomalies are very uncommon and case 1 arises less frequently than case 2 and case 4. KDDnet exhibits an unusually high number of all four cases (especially case 4); further investigation is necessary to explain this fact. A small number of AS triples (A, B, C) are responsible for the vast majority of the anomalies. For most of the routing tables, ten different AS triples were responsible for more than 90% of the anomalous paths.

D. Common Anomaly Patterns

The last column in Table IV lists one popular triple for each routing table dump. For example, the anomaly (1 65112 6461) includes a private AS number (65112) [17] that should not appear in an AS path between two public ASes (1 and 6461). This anomaly pattern alone accounted for more than half of the anomalous paths in this dump. We analyzed the other anomaly patterns using the RADB *whois* data [18] which identifies ASes by name and sometimes includes a list of import and export policies. Consider the anomaly pattern (7018 6841 3300) with AT&T (7018), Infonet Europe (6841), and AUCS (3300). The RADB data states that AS 6841 exports and import all advertisements from AS 3300. We confirmed that Infonet and AUCS have recently merged [19]. The anomaly patterns (1239 1740 7018) and (3561 5400 5727) seem to have similar explanations; Cerfnet (1740) was acquired by AT&T (7018), and AS 5400 and AS 5727 are both part of the Concert IP backbone. For the anomaly pattern (1239 8043 6395), further investigation showed that IXC Communications has acquired SmartNAP (8043) and IXC was later renamed as Broadwing (6395) [20]. Similar anecdotes apply to many of the other popular anomaly patterns.

Identifying anomaly patterns may be a useful way to detect sibling relationships. In the absence of misconfigurations, we can label all case 1 anomalies as caused by sibling relationships. That is, if the AS path (A, B, C) is a case 1 anomaly, either A and B or C and B are siblings. We do not extend this to case 2 or case 3 anomaly patterns since these anomaly patterns may represent backup relationships or other complex transit agreements. Ignoring the KDDNet dump, we observed 109 unique case 1 anomaly patterns. In these 109 patterns, we found 190 unique AS pairs with possible sibling relationships; 22 of these possible sibling relationships appeared in multiple paths. As an example, AS 2685 (AT&T Global Network Services) appears in the middle as a customer in many case 1 anomalies. This AS may have a sibling relationship with AS 7018 (AT&T). Our sibling inferences account for roughly 0.8% of the edges in the AS graph. The work by Gao [10] identifies 1.5% of the edges to be siblings; her validation of a subset of these inferences on a private data set found that 20% of these inferences were valid. We plan to explore our approach for detecting sibling relationships in more detail as part of future work.

V. Internet Hierarchy

The term *tiers* has been used informally in discussions about the hierarchy of ASes in the Internet topology. However, precise rules for classifying ASes into tiers have not been resolved. The work in [9] uses node degree to group ASes into different classes. ASes with a large number of neighbors are placed above ASes with a small node degree. However, a simple degree-based classification may not capture the essence of *tiers* in the hierarchy. In this section, we infer a hierarchy that symbolizes the business relationships between ASes. Typically, a customer should be at a lower level in the hierarchy than its providers. An essential component to such a characterization is a knowledge of the relationships between ASes. We now briefly describe the rationale behind our approach for inferring the different levels in the AS hierarchy.

In order to capture these hierarchical properties, we represent the AS topology as a directed graph where the direction of an edge indicates the type of relationship between the two ASes. To the best of our knowledge, previ-

TABLE IV
QUANTIFICATION & DISTRIBUTION OF PATH ANOMALIES

AS #	AS Name	# of Paths	Anomaly Paths	Anomaly %	Unique Anomalies	Case 1	Case 2	Case 3	Case 4	Popular Anomaly
1	Genuity	65,383	1,679	2.57%	115	23	89	3	159	(1 65112 6461)
7018	AT&T	141,283	3,357	2.37%	101	16	85	0	181	(7018 6841 3300)
6893	CW	70,253	2,384	3.39%	148	24	115	9	210	(3561 5400 5727)
2548	MaeWest	115,199	2,298	2.00%	233	57	171	5	256	(1239 8043 6395)
1755	Ebone	23,469	703	3.00%	131	21	103	7	212	(3300 8933 2200)
2516	KDDI	126,414	10,709	8.47%	594	248	306	40	1101	(209 1800 1239)

ous works have considered the topology as an undirected graph that simply captures the connectivity between the ASes. In our graph, a provider-customer relationship between A and B is represented by a directed edge from A to B and a peering relationship between A and B is represented by two directed edges, one from A to B and the other from B to A. We analyze the graph constructed by applying our inference techniques to the ten Telnet Looking Glass servers discussed in Section IV.

A. Customers and Small Regional ISPs

Customers are the easiest class of ASes that can be classified from this directed graph structure of the AS topology. Customers are those stub networks which are origins and sinks of traffic and which do not carry any transit traffic. From the very definition of the direction of edges in our graph, we can infer the customer ASes to be the leaves of this directed graph. In a directed graph, a leaf is a node with out-degree 0. Since an undirected graph makes no distinction between out-degree and in-degree, customers with multiple providers would have a degree more than 1 and hence would not appear as leaves of the graph. Modeling the topology as a directed graph provides a more precise characterization of the bottom-most layer in the AS hierarchy. In the directed graph constructed from the ten BGP dumps, $8,852$ of the $10,698$ ASes are leaf nodes. The rest of the graph contains just 17.5% of the ASes.

Once we identify the customers and remove these nodes, the resulting graph has a new set of leaves. These leaves represent small regional ISPs that have one or more customers. We can continue the process of pruning the leaves of the graph until we reach a point where the graph has no leaves. This involves applying a reverse pruning algorithm similar to Figure 1 in Section III-B. We define the set of nodes removed by this process as *small regional ISPs*. Since every peering relationship is represented as a loop of two edges in the graph, no ASes with peering relationships are included in this layer. Applying the reverse pruning algorithm to our graph reveals 950 small regional ISPs. We define the remainder of the graph as the *core*, consisting of a connected component with just 857 ASes and $6,578$ unidirectional edges. This represents more than 25% of the total number of edges in the graph. The nodes in the core have an average degree of 6.

B. Dense Core

The set of ASes that remain after the pruning process represent the *core* of the Internet. Given the nature of the reverse pruning process, we can infer that for every AS present in the core, all of its peers and its provider should also be present in the core. The core of the graph should include the small number of so-called *tier-1* providers. In practice, the term "tier-1 provider" is loosely defined as a "large" AS or as an AS that does not have any upstream provider. We could identify these ASes by looking for all provider-free nodes. However, this approach would be sensitive to a small number of missing edges or misclassified relationships in our AS graph. Instead, we could exploit the observation that every provider-free AS would peer with every other provider-free AS to ensure reachability to all destinations. That is, the set of tier-1 ASes should form a clique where every AS has an edge to and from each of the other ASes. Other provider-free ASes, if they exist in our graph, would be excluded from the set of tier-1 providers.

In practice, some ASes may have complex transit or backup relationships to provide connectivity. We define a weaker notion of the *dense core* as the largest subset of ASes whose induced subgraph is "almost a clique." We define a directed graph of N nodes to be *dense* if every node in the graph has an in-degree and out-degree of at least $N/2$. We have set $N/2$ as an artificial cut-off for determining the dense core in the AS topology. The problem of determining the largest clique in a graph is NP-hard. Given that a clique is just one example of a dense graph, the problem of finding the largest dense subgraph of a graph

compute $z \in v(G)$ with maximum out-degree;
$X = \{z\}$;
$pos(z) = 1; r = 1$;
while ($X \neq v(G)$) {
    compute $y \in v(G) - X$ with max $d(y, X)$
        (selecting the y with the max out-degree)
    $X = X \cup \{y\}$;
    $maxindegree(r) = d(y, X)$;
    $r = r + 1$;
    $pos(y) = r$;
}

Fig. 3. Greedy algorithm to order the nodes

becomes much harder. We have developed a greedy algorithm for identifying the ASes in the dense core.

B.1 Identifying the Dense Core

First, we order the vertices based on a "greedy" notion of connectivity, following the algorithm in Figure 3. Let G represent the directed graph representation of the core. Let $v(G)$ and $E(G)$ represent the vertices and edges of the graph G. Let $d(x, Y)$ for $x \in v(G)$ and $Y \subset v(G)$ denote the number of edges of the form (x, z) where $z \in Y$. Connectivity from a node to a given set of nodes refer to the number of directed edges from that node to any of the nodes in the set. Assume that k of the N nodes are already ordered. For each of the remaining $N - k$ nodes, we determine the connectivity to the k nodes and pick the node with the maximum connectivity as the $(k+1)^{th}$. When multiple nodes have the same connectivity, we choose the node with a higher out-degree. In Figure 3, $pos(x)$ denotes the position of a node x in the final ordering.

Let x_i denote the i^{th} AS in the ordering and X_i be the set of the top i ASes. Let $conn(i)$ represent the connectivity of x_i which is equal to $d(x_i, X_{i-1})$. We define the dense core as the set X_k for the smallest value of k such that $conn(k+1) < (k+1)/2$ and X_k is dense. Once the value of $conn(k+1)$ falls below the value $(k+1)/2$, the $(k+1)^{th}$ node will violate the *dense* property. Therefore if $conn(k+1) < (k+1)/2$, the induced subgraph of X_{k+1} will not be dense since the out-degree of x_{k+1} will be less than $(k+1)/2$. However this does not mean that if $conn(k+1) > (k+1)/2$, then X_{k+1} is dense. Consider the scenario where a node x_j for some $j < k$ is linked to more than $j/2$ elements in X_j and not linked to any node in $X_k - X_j$. This is an example where $conn(k) > k/2$ but X_k is not dense. In this regard, our algorithm is greedy. For the AS topology that we obtained, the point where $conn(k)$ dropped below $k/2$ was the first value of k for which X_k was not dense. This indicates that the ordering output by the algorithm was indeed a good ordering for choosing the vertices of the dense core. In other words, it validates the rationale behind our greedy approach that if y appeared before z in the ordering then y had a better chance of being present in the dense core than z.

B.2 Properties of the Dense Core

Applying this algorithm to the core of our graph, we identify a dense core consisting of 20 ASes. These ASes include the large ISPs such as Genuity, Sprint, AT&T, Exodus.net, and Alternet. The top 20 ASes have a very dense connectivity of 329 peering links. The top 15 of the 20 ASes almost form a clique with only three edges missing from the clique. The largest clique we observed in this innermost core consisted of 13 ASes. The 20 ASes have 6,852 provider-customer edges to customer ASes and 964 provider-customer edges to the small regional providers. After removing the dense core, the remainder of the core consists of 837 ASes.

C. Transit Core

After removing the dense core, we noticed the presence of other large national providers and hosting companies that have peering relationships with many of the ASes in the dense core. To identify these ASes, we define the notion of a *transit core*. Nodes in the transit core peer with each other and with ASes in the dense core, but they do not tend to peer with many other ASes. In our directed graph representation, these peering links are essentially the incoming directed edges from vertices outside this set to vertices within the set. We define such a set of edges to be the *in-way cut* of the graph induced by the given set. Using this property, we define the transit core as the smallest set of ASes containing the dense core which induces a weak in-way cut. We can presently visualize a weak in-way cut to have a small number of edges compared to the total number of ASes in the transit core.

C.1 Identifying the Transit Core

Given $X \subset v(G)$, let $cut_{in}(X)$ denote the set of all edges of the form (y, z) where $y \in v(G) - X$ and $z \in X$. We define a cut X of the vertex set $v(G)$ to be a weak cut if $|cut_{in}(X)| < |X|/2$. The problem of finding weak cuts in a graph is NP-complete and there are no good approximation algorithms for that problem. Given that the transit core is a super-set of the dense core and that the dense core is derived by the greedy ordering, we apply the same ordering to find the transit core as was used to find the dense core. A natural way of using this ordering to find the transit core is to find the smallest value of k such that

TABLE V
DISTRIBUTION OF ASES IN THE HIERARCHY

Level	# of ASes
Dense core (0)	20
Transit core (1)	162
Outer core (2)	675
Small regional ISPs (3)	950
Customers (4)	8852

TABLE VI
INTER-CONNECTIVITY ACROSS LAYERS

Layer	0	1	2	3	4
0	329	776	931	964	6852
1	213	1052	1344	728	3660
2	8	74	1070	390	3196
3	0	0	0	202	2376

$|cut_{in}(X_k)| < k/2$. Surprisingly we found that the value of k at which $|cut_{in}(X_k)| < k/2$ also satisfied the property that $conn(k+1) = 1$. This means that no two edges in $cut_{in}(X_k)$ have the same source. A weak cut also means that more than 50% of the ASes in X_k do not have any peering relationship with any of the ASes in $v(G) - X_k$. Hence by this definition, X_k should indeed contain all the transit providers.

C.2 Properties of the Transit Core

Applying the in-way cut algorithm to our graph, we discover a transit core consisting of the 162 ASes, not including the 20 ASes in the dense core of the graph. These 162 ASes have 213 peering links with ASes in the dense core. Concert, Singapore Telecommunications, UUNet European division, Teleglobe European division and KDDi Corporation, Japan are some example ISPs in our transit core. We found many of the top providers in Europe and Asia to be present in our transit core.

D. Outer Core

We classify all of the remaining ASes in the core as the *outer core*. The members of the outer core typically represent regional ISPs which have a few customer ASes and a few peering relationships with other such regional ISPs. The outer core consists of 675 ASes that have 8 peering sessions with ASes in the dense core and 74 peering sessions with ASes in the transit core. We observed that many members of our outer core are regional ISPs. Some examples include Turkish Telecomm, Williams Communications Group, CAIS Internet, Southwestern Bell Internet Services and Minnesota Regional Network. It is interesting to note that while Exodus Communications (AS 4197) is present in our outer core, Exodus.net (AS 3967) is present in the dense core.

E. Summary

Table V summarizes the number of ASes at each level in the hierarchy—dense core (layer 0), transit core (layer 1), outer core (layer 2), small regional ISPs (layer 3), and customers (layer 4). Table VI summarizes the connectivity between various layers in the AS hierarchy. Each number refers in the table refers to the total number of edges from the layer represented in the same row to the layer represented in the same column. For example, 776 is the total number of edges from layer 0 to layer 1. The tables shows several key properties of the Internet topology:

- The ASes in dense core are very well connected.
- As we move from the dense core toward customers, the inter-layer and intra-layer connectivity drops significantly.
- The large number of customer ASes have their providers distributed across all the layers. The ASes in layer 0 support a large number of customer ASes. This indicates that the connectivity across layers is not strictly hierarchical, as also observed in [9].
- The number of edges within the outer core is less than the total number of vertices in the outer core. This indicates the presence of multiple disconnected groups of ASes in the outer core; ASes in different groups communicate via ASes in the dense core and the transit core.

The graphs in Figure 4 explores the relationship between node degree and the layers in the hierarchy. We define node degree as the number of neighboring ASes without regard to the relationship. The top graph plots the cumulative distribution of node degree on a logarithmic scale and the bottom graph focuses on the large number of ASes with no more than 15 neighbors. In general, layer 0 and 1 ASes have high degree, and layer 3 and 4 ASes tend to have low degree. However, this is not universally true. Some customers at layer 4 have a large number of upstream providers, and some ASes in the dense core at layer 0 have a relatively small number of neighbors. For example, our results suggest that AS 1833 (TeliaNet USA) has a degree of only 40. Yet, we classify TeliaNet as part of the dense core due to its rich collection of peering relationships. A hierarchy based solely on degree distribution would not be able to make this distinction.

VI. CONCLUSIONS

The relationships between ASes has a significant impact on the flow of traffic through the Internet. Our work makes two important contributions toward understanding the structure of the Internet in terms of these relationships:

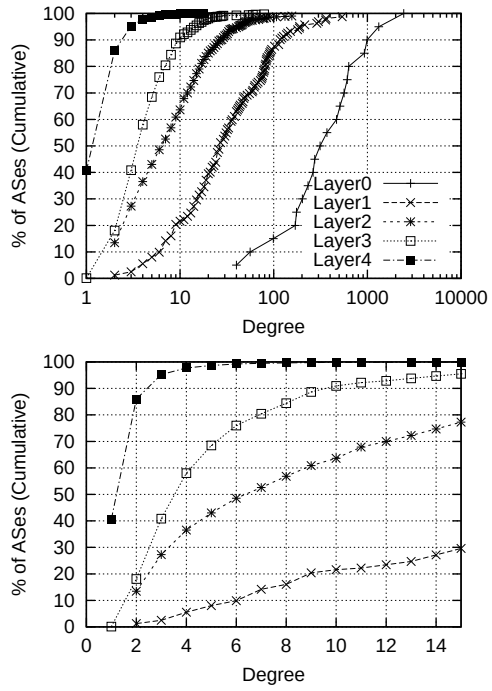


Fig. 4. Cumulative distribution of AS degree by layer

- An algorithm for inferring AS relationships from partial views of the AS graph from different vantage points
- A mechanism for dividing the Internet hierarchy into layers based on AS relationships and node connectivity

The complete structure of the Internet is unknown and difficult, if not impossible, to obtain. Our approach is comprised of many heuristics, with certain limitations:

- We draw our inferences based on only ten vantage points available from Telnet Looking Glass servers. Ideally, we would have a larger collection of routing tables from more diverse vantage points, including smaller customer ASes.
- We treat the RouteViews routing table as a view from a single AS. In future work, we plan to extract a separate view for each AS participating in the RouteViews project.
- Multiple ASes may fall under the administrative control of a single institution, due to historical artifacts and market forces. We plan to extend our methodology to incorporate more complex routing policies that are not captured by the traditional customer-provider and peer-peer relationship.

Despite these limitations, we have shown that our approach provides a detailed view of the Internet topology in terms of the relationships between ASes.

REFERENCES

[1] J. W. Stewart, *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, 1998.

[2] S. Halabi and D. McPherson, *Internet Routing Architectures*. Cisco Press, second ed., 2001.

[3] G. Huston, "Interconnection, peering, and settlements," in *Proc. INET*, June 1999.

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

[5] B. Krishnamurthy and J. Wang, "On network-aware clustering of web clients," in *Proc. ACM SIGCOMM*, August/September 2000.

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

[7] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network topologies, power laws, and hierarchy," Tech. Rep. 01-746, Computer Science Department, University of Southern California, 2001.

[8] "CAIDA Web Site." `http://www.caida.org`.

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

[10] L. Gao, "On inferring automonous system relationships in the Internet," in *Proc. IEEE INFOCOM*, November 2000.

[11] "University of Oregon RouteViews project." `http://www.routeviews.org/`.

[12] "BGP tables from the University of Oregon RouteViews Project." `http://moat.nlanr.net/AS/Data/`.

[13] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. ACM SIGCOMM*, pp. 251–262, August/September 1999.

[14] C. Jin, Q. Chen, and S. Jamin, "Inet: Internet topology generator," Tech. Rep. CSE-TR-433-00, U. Michigan, September 2000.

[15] L. Gao, T. G. Griffin, and J. Rexford, "Inherently safe backup routing with BGP," in *Proc. IEEE INFOCOM*, April 2001.

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

[17] J.Hawkinson, "Guidelines for creation, selection and registration of an Autonomous System," RFC 1930, IETF, March 1996.

[18] "RADB Whois Server." `whois.radb.net`.

[19] "Infonet Europe Web Site." `http://www.infonet-europe.com/`.

[20] "Preston Gates Ellis Acquisitions." `http://www.prestongates.com/meetpge/pro.asp?proID=611`