ETH zürich Research Collection

Report

Inferring AS Relationships Beyond Counting Edges

Author(s):
Huber, Beatrice; Leinen, Simon; O'Dell, Regina; Wattenhofer, Roger

Publication Date:
2004

Permanent Link:
https://doi.org/10.3929/ethz-a-006744314 →

Rights / License:
In Copyright - Non-Commercial Use Permitted →

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.

ETH Library

Inferring AS Relationships Beyond Counting Edges

Beatrice Huber, Simon Leinen, Regina O'Dell, Roger Wattenhofer

D-INFK Tech Report Nr. 446

Abstract— **Routing in the Internet is strongly shaped by the business relationships between administrative domains (Autonomous Systems, AS). In this paper we propose new techniques to classify these relationships. Inspired by Lixin Gao's seminal work on valley-free paths, we explore several ways to measure the importance of an AS which go beyond the original degree-ranking. In addition we study what information can be gathered by parsing the Routing Registry databases. Furthermore we undertake a thorough analysis and experimental comparison of all the proposed approaches known to date. Not surprisingly, the various methods all reveal different views of the business relations. In fact, by using the different measures or even a combination of them, one can tune the ranking of a given AS by a wide margin.**

I. Introduction

Routing in the Internet is determined to a large extent by the interactions of Autonomous Systems (AS). An AS can be seen as a single entity from the outside, eliminating the need to know about the intra-network details of an AS. These ASes enter into certain business relations among each other, which can be implemented via policies in the Border Gateway Protocol (BGP) [1], [2]. Generally, BGP is used to exchange routing information between Autonomous Systems, that is, how to reach a range of IP addresses, a so-called prefix.

One can construct an Internet graph based on the knowledge of (physically) connected AS pairs. Previous work has focused mainly on the connectivity (for example, Skitter [3]) or the general structure [4] of the Internet graph. However, recent efforts [5], [6] demonstrate that graph connectivity information alone is not enough to understand routing. In fact, the relations between the ASes are of central importance. For example, consider Figure 1. Since an AS pays its providers for Internet transit, a multihomed AS with several upstreams does not want its providers to exchange traffic between each other through that AS. So although there exists a connection through the customer AS, the two providers will send traffic intended for each other using a different path, such as via their own provider.

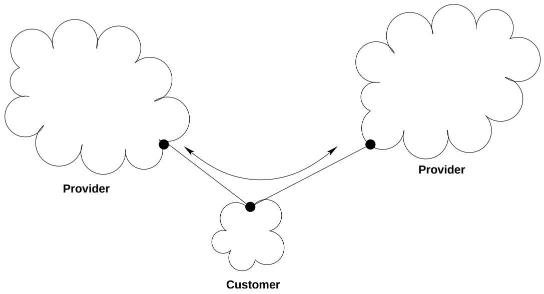

Fig. 1
Connectivity does not directly imply reachability: Providers should not route over their common customer.

Global routing cannot be understood by looking at the physical links connecting ASes without taking into account the influence of business decisions. Such an understanding would be advantageous to businesses as well as the research community. Both parties can optimize various aspects of Internet routing if sufficient information becomes available. It can assist businesses in determining what relations to enter with a potential partner, whether it be a peering or customer-provider relationship. Thorough knowledge of the Internet routing topology can also aid in choosing locations for webserver replicas. Along the same lines, an AS would be able to choose better candidates for backup providers to ensure maximum reliability. BGP misconfigurations could be identified by detecting errors in the export rules [7]. A deeper understanding would allow researchers to further investigate and characterize the robustness of the Internet with hopes of improving its overall performance. It has also been an interest to build a hierarchy of ASes which classifies them into different tiers.

Only a few others, among them [5], [6], have looked at the Internet graph with the business relations attributes attached to the edges. Their efforts are a first step and we

B. Huber and S. Leinen are with SWITCH, Zurich, Switzerland, emails: bea.huber@switch.ch, simon@limmat.switch.ch
R. O'Dell and R. Wattenhofer are with ETH Zurich, Switzerland, emails: {regina.odell,wattenhofer}@inf.ethz.ch

will discuss them in detail throughout the paper. Part of this work is to investigate the current strengths and limits of Internet routing topology inference algorithms.

A. Model and Measurements

The central aim of this and related work is to model certain real-world aspects based on potentially indirect observation. The real-world aspects in our case are the business relations between the autonomous systems. The impact of these relations on reality is that they heavily influence routing in the Internet. Therefore, we want to assign *roles* to pairs of ASes which *model their business relation* which in turn determine the *route announcements*, graphically:

$$\text{role} \overset{\text{models}}{\longrightarrow} \text{relation} \overset{\text{determines}}{\longrightarrow} \text{routing}.$$

In a sense, then, we can view this as a physics-style experiment of modeling real-world phenomena.

Based on assumptions about these business relations (supported in BGP by policies), we can deduce the types of routes the different ASes announce (expressed as filters). Specifically, in BGP, one can define which routes to accept from whom and which routes to announce to whom. The task of the algorithms in this paper is to determine the inter-AS roles based on diverse observations of the Internet. These measurements are based primarily on parsing BGP routing tables or querying the Routing Registry. Furthermore, we propose to estimate other quantities such as the number of end-users (cf. Section II-A.4) to serve as indicators of the types of AS-AS relationships when other methods fail or do not provide enough information.

In accordance with previous work, we consider four types of (directed) relationships: customer to provider (C/P), provider to customer (P/C), peerings (P/P), and siblings (S). They are motivated by the fact that customers pay their providers for transit access to the Internet, peerings decrease the reliance on providers and therefore the cost of purchased Internet transit, and siblings normally have mutual transit agreements, usually for backup. We will mostly discuss the first three types here and when we compare an approach producing siblings to one without, we treat siblings as peers (P/P = S).

In practical terms, we have the following export rules (route announcements) which are applicable in most of the cases. It is common to assume these filters when talking about routing policies [5], [6], [8]. They are adapted from [5] and we reproduce them here for completeness.

Exporting to a provider An AS can export its internal routes and its customer routes to the provider, but usually does not export its other provider or peer routes.

Exporting to a customer An AS can export its internal routes and its customers routes as well as its provider and peer routes.

Exporting to a peer An AS can export its internal routes and customer routes, but usually does not export its provider or peer routes.

Exporting to a sibling An AS can export its internal routes, the routes of its customers as well its provider or other routes.

Knowing the export rule, we can deduce the complementary import rule. For instance, one imports all routes from a provider, but one imports only internal routes from a customer, and no routes of other providers or peers of customers.

Of course, this is quite a simplified view. In reality, there are hybrid forms of the four types of relationships. For example, an AS A must pay to get some of the routes announced by AS B, but for other routes they act as peers. Partial transit restricted to a geographical region (regional transit), or a combination of being a customer and partial transit provider to an upstream AS are not uncommon. Paid peerings represent such a hybrid form as well.

B. Related Work

Efforts to map the topology of the Internet are by far not new [9]. There are (at least) two levels of detail involved when we consider the Internet as a graph. One is on the AS level as we do in this work, the other is on the router level (for example, see [10]). For a discussion on the difference of the two obtained graphs, see [11]. Most of the previous work [12], [9], [13] on topology discovery on the AS level has focused on *connectivity*, yet, as has been noted above, the effective routes depend on the AS business policies as well and not on connectivity alone. The work in [14] showed the NP-completeness of the problem if formally defined as in [6].

The two most prominent papers who also aim to infer AS relationships are that of Lixin Gao [5] and Lakshminarayanan Subramanian *et al.* [6]. We will discuss them in detail in the following sections.

In a very recent work, it came to our attention that Siganos and Faloutsos [8] also describe an approach of parsing the registry as we attempt in Section II-C. Their main goal is to quantify the quality of current Internet Routing Registries. Classifying AS-AS relationships is part of their process, although their focus is on the robustness of the Internet and to serve as motivation for businesses to update their registry entries. Given the somewhat bleak outlook we give on the task of modeling the AS Internet graph to a high accuracy, we can view their work as complementary to ours.

There is also recent work on applying the findings of

AS-level reachability to other network problems. Wang and Gao [15], assuming knowledge of inter-AS relationships, investigate by how much actual routing preferences correspond to what would be expected given the business relations. They use the work of [5] as the basis for the data and they argue that the potential for error introduced is negligible. However, they show this directly for only 9 ASes and their algorithm for verifying these relations uses further heuristics which presuppose already accurate AS-AS relation information. Succinctly, they use the community attribute to verify inter-AS relations, and they use assumptions about inter-AS relations to verify the community-attribute conclusion. While the data is likely to be correct for a large set of "normal" ASes, it is clear that there remains a need for interpreting the AS-graph properties to a high degree of accuracy. This is further evidenced in another line of research, namely that of network topology *generators* which try to capture the essential properties of real networks. See [16] and references therein for an overview. In this case, an accurate AS-level graph is needed as a reliable check of the generated topologies against an existing network of interest.

C. Contributions

Our contributions to the area of modeling and inferring the BGP Internet graph include the proposal of several new techniques, namely the ones introduced in Sections II-A.2-II-A.4, II-C, and a modified version of II-D.

Additionally, we present a compilation of numerous methods known to us and critically discuss the insights and shortcomings of each in Section III. So far, when a new approach was introduced, the authors only compared their results to the original work of Lixin Gao [5]. We present the first comparison encompassing all known algorithms. In Section IV, we further identify additional problems of a more general nature which we have discovered in the process.

To support our claims and to illustrate the different outcomes of the presented algorithms in individual cases, we have developed a web-interface to our database.[1] There, the user can enter any AS number and obtain the classification of all neighboring relations as determined by the different algorithms. For the valley-free-paths approaches of Section II-A, the user can assign different weights to each measure and observe the outcome on the calculated rank of the given AS in the entire (visible) Internet graph.

II. Approaches

We will begin by describing in detail the most prominent of the existing and our new approaches. We postpone the majority of the critical analysis of each to Section III, where we can base our discussion on the actual results obtained by each algorithm.

A. Valley-Free Paths

One of the first algorithms was proposed by Lixin Gao (for the journal version, see [5]) and its main premise is that BGP routing paths are inherently valley-free. This is motivated by the discussion surrounding Figure 1 and illustrated more technically in Figure 2. The measurements in this approach consist of parsing a large number of available routing tables (see [17]) so that we end up with an extensive collection of AS paths.

The idea of the algorithm is then as follows: Given that all paths are valley-free, there will be one so-called *top provider*, the peak of the path. All preceding edges on the (directed) path are in a C/P relation, all subsequent edges P/C. The problem remaining is to determine the top provider since this cannot be read off from the path itself. This is where the various approaches in this section (Sec. II-A.1-II-A.4) differ from one another. They all have a different *measure of importance*, $weight$, to decide on the peak of each path (i.e., for a path $A_0 \dots A_k$, the top provider is A_i with $\max\{weight(A_i)\}$).

Afterward, the basic algorithm counts up the number of relations determined for each seen edge and, essentially, if more than L classify $A \to B$ as a C/P edge (and at most L as P/C), then AS A is said to be a customer of AS B. In [5], $L = 1$. We can basically implement a "majority vote" at this point. If the number of C/P and P/C classifications are about the same, then Gao designates them as siblings, which we will treat as peers for the purpose of comparison. The above step does not yet account for peerings. The premise about valley-free paths also states that any P/P relationship may only occur once next to the top provider A. The motivation for the algorithm here is that peerings only occur among near equals. Therefore, unless they have already been classified as siblings or customers, we look at neighbor B of A with the next highest $weight$ value and set them as peers only if $weight(B) \geq weight(A)/R$, that is, their respective importance differs by at most some ratio R.

A.1 Degree

The importance measure of the original algorithm [5] is the degree of an AS. Since we have a plethora of paths

[1] http://dcg.ethz.ch/projects/as_relations/as.php

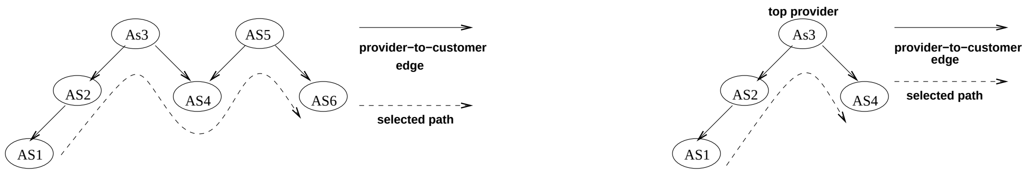

Fig. 2

AN EXAMPLE OF A PATH WITH A VALLEY AND A VALLEY-FREE PATH.

available, one can estimate[2] the degree of A by summing up the different neighbors of A in every path. This stems from the intuition that a provider typically has a larger size than its customers and that the size and importance of an AS are assumed proportional to its degree in the AS graph.

A.2 Size of the Address Space

We can consider the size of the IP address range announced by an AS as an alternate estimate of its relative importance. Specifically, for every AS path P, we assign all announced prefixes in P to the last AS of P. The total address space of an AS is then calculated by summing up the number of addresses contained in each path.

There are some details which deserve mentioning. First of all, the prefixes originating in private AS numbers (64512-65535) have to be replaced by the public AS number from where it originates. We determined this by querying the routing table and by looking at the second-to-last AS in the path where the private AS number appears. Note that, technically, private AS numbers should not be announced in the global BGP routing. This mistake happens for about 100 announced prefixes, but most of the prefixes belong to the same AS. Only a few ASes continue announcing private ASes.

Secondly, some ASes announce the same network twice, for example, once as a /16 and again as a /20, which is completely contained in the /16. (For instance, AS-271 announces the prefixes 134.87.0.0/16, 134.87.110.0/23, 134.87.224.0/20 and 134.87.225.0/24.) We remove these overlaps when counting addresses. Upon taking a closer look at the prefixes, we see that there are some Multiple Origin AS (MOAS) prefixes, which means that there are two or more (up to six in one case) ASes where the prefix originates. They may be anycast addresses which belong to all. They are counted as belonging to each one of those ASes.

Thirdly, there remain some relics from the early days of the Internet, when the elementary task of maintaining a list of assigned network addresses was carried out voluntarily by Jon Postel, using (according to legend) a paper notebook. Back then, it was relatively easy to get a /8 address space. Nowadays, the minimum allocation size starts at /20, and only when about eighty percent of all the address space currently allocated is used in valid assignments or suballocations is it possible to augment the address space. Therefore, we can deduce that a /8 is an old network, but we do not know anything about the density of the addresses used in the network. Maybe it is a huge network with thousands of end users or just a vestige. To give less influence to the /8 nets, we weighted them as 4 times a /16 net. Also, some /8 have been split up and shared between multiple ASes. For instance, 12.0.0.0/8 was given to AT&T Bell Laboratories in June 1995. Today, over 600 ASes announce a prefix out of the range originally given to AT&T Bell Laboratories, whereas most of them are in a customer-provider relationship with AT&T.

Lastly, some ASes do not have their own address space, because they only provide transit for others. Although they do not have their own address space, they may have several customers and thus be of importance from the global point of view. For example AS-5673, Pacific Bell Internet Services, provides transit for some ASes which are classified as customers. In the routing table, though, AS-5673 never appears at the end of an AS path as it only provides transit for its customers.

Due to these last two reasons, it is necessary to include the size of its customers in order to calculate the real size of an AS. To do so, however, we already need to know the customer-provider relationships to calculate the address space recursively. So in a first step, we took the results produced by the original degree-based algorithm from Sec. II-A.1. Then, we were able to determine the importance of an AS based on the address range and ran the algorithm with the new importance measure. When calculating the address space recursively, some cycles became apparent in the data used as input. These had to be analyzed and broken to finish the recursive calculation of the address range. (See Section IV-B.)

There remained one final caveat. When we now run the

[2] since in practice we do not see the entire graph, we cannot know the precise value

algorithm [5] with the address space as a measure instead of the degree, the peer ratio always has to be set to infinity. This is due to the different magnitudes involved among address spaces and degrees. The degree varies only between 1 and about 2400 whereas the address space, counted recursively, varies between 254 and some millions.

A.3 Number of Important Websites

Seeing the difficulties in determining the actual host density of an address range, we considered another, totally different measure to express importance. We collected 65'000 IP addresses by querying Google.com with random words and saving the IP address of the top-ranked website. Thus, the more important websites having an IP address belonging to a prefix of AS A exist, the more important A is. Again, as with the address space, we calculated these numbers recursively for an AS and its customers in the precomputation step using the degree-based algorithm.

A.4 Number of Endusers

We were also interested in the ASes at which the endusers are located, and whether there is a correlation between the number of endusers and the number of hosted websites within the same AS. To count users in an AS, we collected roughly 500'000 IP addresses of Gnutella users. This is motivated by the idea that an AS becomes more important the more users connect through it.

Because Gnutella is a very open protocol, IP addresses can be faked or internally-used address may be propagated. The number of addresses in the logs which were not routed was 1%. However, we assume that the percentage of faked addresses is quite low because there is no great incentive to do so.

B. Ranks from Multiple Vantage Points

A different approach is given by Subramanian *et al.* in [6]. The key idea is that the relationships between the ASes may be interpreted differently from various vantage points, but when we "average" them, we can obtain a much more accurate picture.

Their rank-based algorithm constructs a directed graph out of the views from multiple vantage points. A single view is obtained from a routing table of a telnet or looking glass server. Then every node of every view gets a rank. This rank is assigned by the following simple algorithm. Every leaf in the graph gets rank 1, then the leaves are removed. The leaves of the remaining graph get rank 2 and are then pruned as well. This goes on as long as there are leaves. All nodes of the residual graph receive the same rank.

In the next step, every AS is assigned an N-dimensional vector (where N is the number of vantage points) $c(i) = (r_{i1}, ...r_{iN})$, where r_{ij} is the rank of AS i in view j. To decide which type of relationship is applicable for two ASes the following heuristics are used (taken from [6]):

Equivalence ASes x and y are said to be equivalent if the number of equal coordinates in $c(x)$ and $c(y)$ is greater than $N/2$. If they are joined by an edge, they are likely to be peers.

Dominance Let $l(i, j)$ refer to the number of coordinates k where $r_{ik} > r_{jk}$. Then a vector $c(i)$ is said to dominate $c(j)$ if $l(i, j) > l(j, i) = 0$. The fact that $c(i)$ dominates $c(j)$ implies that AS i is a provider of AS j.

All edges not matching these rules are determined by probabilistic rules as follows:

Probabilistic Equivalence ASes i and j are probably peers if

$$\frac{1}{\delta_E} \leq \frac{l(i,j)}{l(j,i)} \leq \delta_E, \text{ for } \delta_E = 2.$$

Probabilistic Dominance AS i probably dominates AS j if

$$\frac{l(i,j)}{l(j,i)} \geq \delta_D, \text{ for } \delta_D = 3$$

with $\delta_D > \delta_E$.

The authors of [6] choose the values of δ_E and δ_D as above. If none of these rules matches, then the relationship is not decidable (arbitrary value).

C. Registry

Another approach to determine the relationships between the ASes is to read the information out of the Internet Routing Registries (IRR) [18]. About 10'000 of the 16'000 ASes have entered at least some pieces of information about their routing policies. This gives a much more complete view than one could ever reach with the routing tables from a single router, or even several vantage points. However, a major problem is that the entries in the registries are not necessarily up to date (for some ASes, the last update was made in 1997), and there is no guarantee that these entries are complete or even correct. This happens because the registries are maintained mostly manually and no regulation exists for ASes to update or enter their routing polices. Some ASes update them diligently because they use the router configurations generated from these entries.

The entries in these IRR databases are written in RPSL, the Routing Policy Specification Language [19]. RPSL provides classes of records to save the policies or maintain information. We are interested in the data about the import and export statements. They can be found in the aut-num class.

The data out of the registry entry of AS A is interpreted in the following way:

```
import:  from AS B accept ANY
```

is interpreted as AS B is a provider of A, because as the import rule prescribes that one only imports all routes from a provider. Likewise,

```
export:  to AS B announce ANY
```

means that AS B is a customer of A, according to the corresponding export rule. Peerings are identified by the entry

```
export:  to AS B announce AS A
import:  from AS B accept AS B
```

An AS A operating partial routing has an entry for its default provider B:

```
default:  to AS B networks ANY
```

The statement `default` dominates over other relationships that would be inferred.

According to the export rule, an AS announces all its customer routes to its providers and peers. The following is an example from the entry for AS-22:

```
export:  to AS4302 announce  AS22
                       AS5855 AS5303
```

Both AS-5855 and AS-5303 are customers of AS-22.

To complicate things, the customers of an AS can be collected in an AS-Macro. The name of such a macro can be chosen freely. It may be written as in Table I.

```
export:  to AS21409  announce AS-GLOBAL
export:  to AS8729   announce AS-GLOBAL
                  AS-SWISS AS-NATIONAL
export:  to AS8387   announce AS-GLOBAL
export:  to AS8404   announce AS-SWISS
                           AS-NATIONAL
export:  to AS8406   announce AS-GLOBAL
```

TABLE I
EXAMPLE MACROS FOR AS-6730.

To resolve the symbolic names, we proceeded as follows. That AS macro announced the most to other ASes we interpreted as the macro containing the customers. Querying the database for AS-GLOBAL for AS-6730 returns the output displayed in Table II, which means that the listed members are customers of AS-6730.

```
as-set  :  AS-GLOBAL
descr   :  SUNRISE, TDC Switzerland AG
descr   :  Customers with
                   global connectivity
members :  AS3092
members :  AS3291
members :  AS5398
members :  AS5426
members :  AS6730
```

TABLE II
THE MACRO AS-GLOBAL.

In this example, we would lose data by only parsing AS-GLOBAL in Table I, because the macro AS-SWISS contains additional customers. On the other hand, parsing all macros announced to other ASes can lead to other mistakes. The macro AS-NATIONAL, for example, contains only peers, see Table III, thus it should not be interpreted as a macro containing customers.

```
descr   :  sunrise National (equal)
               peers in Switzerland
descr   :  connected to AS6730
members :  AS513
members :  AS559
members :  AS3291
members :  AS3303
```

TABLE III
THE MACRO AS-NATIONAL.

We further parse the registry entry for the occurrence of macros such as AS-PROVIDER, AS-TRANSIT, AS-PEERS, AS-CUSTOMERS, which has been quite fruitful. In order to decide whether it is a direct or indirect customer in the case of a nested macro, we double-checked the edge in an undirected graph which represents all possible connections in the routing table. Altogether, we observe that several simple rules can already parse a large part of the Registries, yet there remain a number of exceptions.

D. Simplified Robban Tool

Finally, a simple idea to infer the type of relationships between ASes is to only look at the last two ASes in a path with at least three ASes. The last AS in the path is considered to be a customer of the next-to-last AS. The occurrences of every pair are counted from the routing tables seen at the Oregon Exchange BGP Route Viewer [17]. We consider only pairs that are seen at least three times, otherwise the pair is deemed irrelevant. We can deduce a peering relationship if there are pairs where AS A provides transit for AS B and AS B provides transit for AS A

Algorithm	(Sec.)	**Peers**	**Total**
Degree $R = 60$	(II-A.1)	7.2%	33636.66
Degree $R = \infty$	(II-A.1)	7.98%	33636.66
Address Space	(II-A.2)	7.38%	33636.66
Websites	(II-A.3)	7.34%	33636.66
Endusers	(II-A.4)	7.36%	33636.66
Rank from [20]	(II-B)	4.03%	34299.2
Rank $\delta_E = 2$, $\delta_D = 3$	(II-B)	1.41%	34814.5
Rank $\delta_E = 200$, $\delta_D = 300$	(II-B)	3.86%	34646.7
Registry	(II-C)	47.89%	77261
Simplified Robban Tool	(II-D)	0.51%	30428

TABLE IV

PERCENTAGE OF PEERINGS. THE TOTAL EXPRESSES THE NUMBER OF ALL CONNECTIONS SEEN BY AN ALGORITHM, FOR THE VALLEY-FREE PATH APPROACHES AVERAGED OVER 3 RUNS, FOR THE RANK-BASED APPROACHES OVER 10 RUNS.

as well.

The original idea for this approach comes from Netlantis [21], where the implementation is more complex than the version described above. There, they fix the highest occurrence of a link, where the considered AS is the last AS in the path, as a C/P relationship. Then they compare the ratio between the highest and the second highest occurrence of a link and deduce from this ratio whether it is also a P/C relationship or a peering. Afterward, the second highest occurrence is compared to the next smaller one and so on. Once classified as a peering, all smaller occurrences of links will be classified as peerings as well.

III. RESULTS AND ANALYSIS

In this section, we provide an overview of the results in terms of the number of types of connections seen by all the aforementioned algorithms. Based on that, we discuss the advantages and disadvantages of each approach. The section numbering corresponds to that of Section II.

In Table IV we see the percentages of connections classified as peerings by the different algorithms along with the total number of relationships seen for each. Siblings are counted to the peerings for the purpose of comparison. The rest of the relationships are therefore of type customer-provider. For the valley-free paths and rank-based algorithms, the results are averaged over 3 and 10 runs, respectively. For algorithm [6], we took the results published on the authors' website [20] and also ran the algorithm with the code they provided us, once with the recommended $\delta_E = 2$, $\delta_D = 3$ and again multiplied by a factor of 100. The factor 100 was chosen to achieve a clear effect and an upper limit. There is an upper limit where the parameters can still rise, but do not produce any more effect. When talking about the percentage of peerings, it also greatly depends on the subgraph of the Internet under consideration. For instance, the registry approach can identify local peerings between small ASes whereas all algorithms using routing tables do not even see a connection.

For the details of the actual sources for each data set, see the individual forthcoming sections as well as the description of the approaches in the previous section.

A. Valley-Free Paths

We will take a look at the results of each variant of the algorithm which assumes that all paths are valley-free. We find that errors arise for every importance measure. An AS providing web-content, for example, probably does not care about a huge address space as a peering criterion, instead, it is interested in peerings with ASes where the readers of its websites are located. So the reasons for choosing providers and peerings not only depend on the address space or the degree of an AS, but likely an intricate mixture of all factors. Thus, the complexity of peering decisions cannot be mirrored by one general definition of importance. Every weighting mechanism leads to a different ranking of the ASes. Tables VII-X, showing the top five providers for each approximation of importance, illustrate this effect.

The main source for all valley-free-paths algorithms is the data archive of the Route Views Project [17]. The routing table data is from July until December 2003.

A.1 Degree

The assumption that the degree correlates with the importance of an Autonomous System in the global hierarchy of ASes may be problematic for the following reasons. A transit AS may have few, but important providers and peers. The customers of this AS may use private AS numbers or an internal routing protocol, so they will not be seen at all in the BGP routing tables.

The input data for this algorithm is gathered by a route collector which receives all routes of peering ASes. These neighbors of the route collector announce all their peer routes to the route collector as well. According to the export rules, peer routes are not announced to other peers or providers, so most peerings are only seen locally. An AS with lots of peers cannot be distinguished from an AS with few peers, because the peerings do not influence the degree of the AS seen by the algorithm, except for the ASes directly connected to the route collector.

There is an added difficulty in fine tuning the ratio R, as already noted in the original paper. In [5], the value R has been set to 60. The more BGP routing tables, the less important is the choice of R (can be set to infinity then). This is because we can eliminate an AS pair from having a peering relationship if the AS pair appears in any BGP routing table as having a transit relationship.

The advantage of this algorithm is that it infers siblings, so we get a more specific analysis. Reducing a sibling to a peer in other cases can hurt the export rule.

Table V displays the results of this algorithm (using $R = 60$) over the course of half a year. We can see the number of appearing ASes and the number of connections rise steadily over time.

Total	**Peers**	**%**	**ASes**	**Date**
34154	2617	7.66235	16540	2003/12/29
33512	2365	7.05717	16375	2003/11/28
33244	2288	6.88244	16320	2003/11/13
32623	2163	6.63029	16221	2003/10/29
32850	2280	6.94064	16132	2003/10/15
32182	2019	6.27369	16019	2003/09/30
31328	1657	5.2892	15888	2003/09/02
31111	1609	5.1718	15829	2003/08/18
30584	1422	4.64949	15742	2003/08/04
30345	1384	4.56088	15658	2003/07/21

TABLE V

RESULTS OF ALGORITHM [5] OVER TIME, FROM LATEST TO EARLIEST.

A.2 Address Space

The results produced by this algorithm do not differ so much from the algorithm using the degree: less than 3% of the relationships change. And the new results show little tendency toward more correctness. The number of improvements and deteriorations are about the same as we observed via a rough manual check. For AS-3303, more peers are correctly recognized, whereas for AS-4589, the peers become customers incorrectly. The shift of the ASes in the ranking causes these changes. AS-4589 had lots of wrongly interpreted customers before, so they were all counted to AS-4589's address space which made the AS more important, and so previous peers become customers. The problem stems from the fact that we bootstrapped the algorithm with the one in Section II-A.1 so that old mistakes are carried over. And, of course, new mistakes may be generated when giving too much or too little influence to an AS due to the difficulty in assigning the used address space.

A.3 Important IP Addresses

The IP addresses associated to users were collected in a crawl of the Gnutella network in May and June of 2002. We obtained a total of about 50'000 distinct addresses. The websites, using random words in google.com, were gathered in January 2004. Also note that to compute the AS relations, we did this recursively as discussed in Section II-A, whereas the rankings are obtained non recursively.

We summarize the results of the important websites, Section II-A.3, and the IP addresses of the endusers, Section II-A.4, together, because the arising problems are very similar. Table VI shows how many links the website based and the enduser-based approach have in common.

Compared to the results of the degree-based approach, the main difference is that there are over 10'000 ASes to which no IP address out of our sample belongs to. Since we have no information about them, they are all weighted equally. These are mainly ASes not belonging to ISPs, but to companies which neither host websites nor sell Internet access to endusers. ASes that only provide transit for other ISPs, but have no endusers or websites within their address space are not considered as being important which is the same kind of problem when looking only at the address range.

Common Links	33512
Common Customer-Provider	28551
Common Peers	804
Common Siblings	216

TABLE VI

COMMON RELATIONSHIPS OF THE WEBSITE AND ENDUSER-BASED APPROACH.

To mention an example of what goes wrong, the peerings of AS-8210 (Teleonor backbone) disappear and instead AS-8210 becomes the provider. This happens because AS-8210 has no important website nor Gnutella users. There are only five smaller prefixes announced by this AS number. In most paths AS-8210 is located next to AS-2119. AS-2119 (Teleonor Internet access) announces a large address space where we find some important websites and Gnutella users. So, AS-2119 always becomes the top provider and according to the valley-free path rule, a peering can only be with the top provider. This means that AS-8210 no longer can be interpreted as a peer of AS-2119. However, in the original approach, AS-8210 has degree 205 and AS-2119 has degree 5.

AS number	AS name	Degree
701	UUNET	2382
1239	Sprintlink	1770
7018	AT&T	1627
209	Qwest	999
3356	Level 3	968

TABLE VII
TOP FIVE BY DEGREE.

AS number	AS name	Address Space
568	DISO-UNRA	72,625,925
3356	Level 3	44,663,587
701	UUNET	35,453,551
7132	SBC	30,510,764
7018	AT&T	28,520,584

TABLE VIII
TOP FIVE BY SIZE OF ADDRESS SPACE.

AS number	AS name	Users
1668	AOL	88631
3320	Deutsche Telekom AG	30330
7018	AT&T	25004
7132	SBC	24004
22909	Comcast Cable	14427

TABLE IX
TOP FIVE BY NUMBER OF (GNUTELLA) USERS.

B. Rank-Based

The rank-based algorithm sees more connections due to multiple vantage points. Interestingly, the total number of peerings of the degree-based algorithm is higher despite the fact that algorithm [6] has multiple vantage points and so theoretically could see (more) local peerings. Table XI shows the number of connections seen by both algorithms, the number of connections interpreted as peerings from both algorithms and the number of customer-provider relationships which they have in common (siblings are considered as peers for this comparison). While the number of common edges is quite high, the number of relationships interpreted as peerings by algorithm [5] versus algorithm [6] is rather low, only about one percent. Note that the data for this algorithm also comes from the routing tables [17].

AS number	AS name	Sites
2914	Verio	2001
13749	Everyones Internet	1457
11305	Interland Incorporated	896
3356	Level 3	719
701	UUNET	677

TABLE X
TOP FIVE BY NUMBER OF WEBSITES.

The largest difference of the results taken from the website [20] is the number of peerings in [5] that are interpreted as customer-provider relationships by the rank-based approach. This is the case for ASes with open peering policies directly connected to the route collector like AS-3303, AS-12956, AS-3257. ASes tend to be more liberal for peerings abroad, so that both parties can save transit cost, and they do not compete with each other, because their customers are from different geographical regions. This way, small ASes can have peerings with bigger ASes. Algorithm [6] is error-prone in theses cases due to the different ranks assigned to ASes not in the same stage (see Section II-B). On the other hand, [6] is better in peerings between bigger ASes like AS-701 (UUnet), AS-3356 (Level 3), AS-1290 (SprintLinkBackbone).

C. Registry

There are a number of complications, in addition to the ones already discussed in Section II-C, associated with the interpretation of the registry entries. The data stems from the databases of RIPE [22] and RADB [23], inspected from October 2003 until February 2004.

Several ASes, as in Table XII, always accept `ANY` for simplicity and to be independent of macro name changes of the neighboring ASes. When generating filters out of this information, routes announced by mistake will not be detected. This `accept ANY` also leads to misinterpretation because all their peers and customers may be incorrectly interpreted as providers. When announcing `ANY` to a customer it is possible to detect the conflict. For peers, there is also a chance to spot a conflict upon examining

Links	Peers	CPs	Date
32634	423	29653	2003/12/29
28660	143	27634	2003/11/28
32094	401	29302	2003/11/13
31576	434	28918	2003/10/29
31649	430	28850	2003/10/15
29915	373	27800	2003/09/30
27665	131	26754	2003/09/02
30362	287	28433	2003/08/18
30260	260	28351	2003/08/04
29842	268	28026	2003/07/21

TABLE XI
COMMON LINKS OF ALGORITHMS [5] AND [6], AGAIN FROM LATEST TO EARLIEST.

the registry entry of the other AS. But for the cases where there is no policy entry for the other AS, we have no control mechanism.

```
import: from AS2818  accept ANY
import: from AS6728  accept ANY
import: from AS8406  accept ANY
import: from AS9191  accept ANY
import: from AS16353 accept ANY
import: from AS20915 accept ANY
import: from AS13005 accept ANY
import: from AS5587  accept ANY
import: from AS25108 accept ANY
import: from AS12621 accept ANY
```

TABLE XII
EXAMPLE OF AN ACCEPT-ANY FROM EVERYONE.

The statements in Table XIII lead to the decision of AS-701 and AS-7046 being peers, although AS-701 is the provider. With additional knowledge about the top ten providers of the world, for instance, such mistakes would not happen.

```
import:  from AS701    accept AS701
import:  from AS14677  accept AS14677
export:  to AS701      announce AS7046
export:  to AS14677    announce AS7046
```

TABLE XIII
ONE-SIDED ENTRY FOR AS-7046.

One could get much better results out of the registry by not just parsing, but being able to understand the context. An example:

```
AS8437
remarks:  ----------------------------
remarks:  Peerings NIX
remarks:  ----------------------------
```

or

```
remarks:  Peers:pref=80,Transit:pref=200
remarks:  AS112 Project:
import :  from AS112 action pref=80;
                         accept ANY
export :  to AS112 announce AS-BIT
remarks:   SurfNet:
import :  from AS1103 action pref=80;
                         accept ANY
export :  to AS1103 announce AS-BIT
```

A person would immediately understand the remark, but since everybody is free on how or whether at all to write remarks, it is infeasible to understand all possible remarks just by parsing.

The registry contains data about inactive ASes as well. Some ASes have changed their number, but the old entry has not been deleted.

A significant advantage of the registry approach is that we obtain information about all ASes who have an entry. When one merely looks at routing tables, only local peerings can be seen. That explains how we get a much higher percentage of peerings. There are over 30'000 peerings registered in the Registry that are not seen in the routing tables used to run the algorithms. Either they are legacy entries or it is a local peering that cannot be seen by the other algorithms. For example AS-1257 peers with AS-16245, which can be verified by the looking glass server of AS-1257.

	II-C vs. II-A.1	II-C vs. II-B
Common Links	30285	31257
Peerings	1736	671
Cust.-Prov.	26972	26996

TABLE XIV
COMPARISON OF THE REGISTRY TO ALGORITHM [5] AND ALGORITHM [6], DECEMBER 2003.

We present a comparison of the results from the registry and algorithms [5] and [6] in Table XIV. The authors of [8], where they also attempt to parse the registries, stated that 83% of the relationships inferred are identical with results produced by [5]. It should be noted that they also inferred siblings. It is indeed a good idea to infer siblings. But only looking at the entry of one AS announcing and accepting ANY from its neighboring AS is not very reliable, because many ASes enter this for their provider as well (e.g. (AS-25899, AS-701), (AS-25899, AS-2914), (AS-25899, AS-7911), (AS-28973, AS-701), (AS-29748, AS-701), (AS-30107, AS-4323)). For ASes mutually announcing ANY and accepting ANY the chance is higher that they act as siblings, but it does not necessarily mean that they are in a sibling relationship (e.g., AS-30916, AS-12713). On the other hand, there may be siblings out there which we do not detect, because their route announcements look like peers.

C.1 Customers

Whereas this approach sees a lot of peerings not identified by the others, there are plenty of customer-provider relationships which are not cataloged in the Registry. The number of customers not seen exceeds 10’000 which is over 30% of all customer-provider relationships seen by the other algorithms. The number of missing edges is so high because important ASes are missing a policy entry in the registry, namely AS-1 (Genuity), AS-701 (UUNET) and AS-1239 (SPRINT), AS-7018 (AT&T), AS-209 (Qwest). They do not have an entry, neither in RADB nor in RIPE, so these ASes would add hundreds of customer-provider relationships to our approach. On the other hand, over 10’000 customer-provider relationships are found by the registry for which no other algorithm even sees a connection. If we only look at results where the registry entries of both ASes suggest the same relationship, the number of C/P relationships decreases by a factor of 10. When querying `sh ip bgp regex AS` A `AS` B on some route servers, there is no connection seen. These seem to be legacy entries or backup providers which are only used in case of failure of the connection to the main provider, which would imply that they are not normally seen in the global routing table. AS-1755, EBONE, was shutdown on the 2nd of July 2002, yet there are still 25 former customers who seem to have not updated their registry entry since then.

D. *Simplified Robban Tool*

The Simplified Robban Tool identifies only 0.5% peerings. Only looking at the last two ASes and dropping paths seen less than three times reduces the total number of connections. Compared to other approaches, we see that most of the missing connections are peerings. About 95% of the relationships inferred are identical to those of the other approaches. This implementation has its greatest weakness in finding peerings, which could possibly be improved by collecting data from several vantage points.

The original Robban Tool used its own collected data on [21]. For the simplified version, and for a better comparison to the other algorithms, we again used the data from the Route Views project [17].

E. *Summary*

In summary, it is evident that each of the above approaches to identifying AS relations has its drawbacks. If we consider the actual state of AS relations as a fixed characteristic of the Internet graph (at a given time), then some of the problems encountered are in fact fundamental, while for others there is hope to alleviate them by other, perhaps external means.

In particular, the valley-free approaches (Section II-A), as they stand now, have the serious methodology flaw that no single measure can capture the diversity of factors influencing business decisions. The potential for improvement lies in a combination of these and probably other importance measures, yet we may never find enough of them to correctly classify a big enough portion of the AS relationships. Furthermore, recall that already the algorithm of Sec. II-A.2 was composed of a combination of two importance measures due to bootstrapping.

Similarly, the rank-based approach suffers from exactly the deficiency that is also its strength, namely the multiple vantage points. As we have seen in Sec. III-B, “uneven peers” can have very different rankings in their respective views, so that the algorithm will never classify them as peers.

The registry approach admits its greatest weakness in the imprecision (or total lack) with which companies enter their information into the global database. In such a case, “external means” would be some sort of an enforcement for ASes to update their routing policy information into the IRRs. Seen from that angle, it is not an entirely fundamental defect. However, the approach is inherently centralized, and an ideal algorithm would be able to gather (approximately) the same information in a distributed manner, as is the case for the other algorithms.

IV. General Problems

Aside from the individual difficulties that trouble each approach, we want to mention reasons why problems arise for all algorithms. Finding a perfect abstraction from reality in our case is impossible and therefore some misinterpretations inevitably occur. Essentially, the discussion below details some further issues that any algorithm with a claim to fame needs to address.

A. *Parameter Dependence*

A general difficulty is that all algorithms, however simple or complicated, are dependent on a parameter to express a ratio. How to tune this ratio has to be found out by looking at examples and observing when they are interpreted correctly. This means you already need some information about some ASes to set your parameter, and tuning the parameters shows that the percentage of peerings can change considerably.

Looking again at Table IV, we can observe this effect in particular with the algorithms of Sections II-A.1 and II-B. (Recall that for the other valley-free paths approaches $R = \infty$.) This shows how much the percentage of peerings can vary by changing the parameters. Even with the registry approach, there is an implicit parameter asserting the confidence in the results given how many "pathological cases" were encountered while parsing. For example, if both ASes of a given link have matching entries in the database, then this is a high-confidence result, whereas one could entirely leave out those links with a problematic one-sided entry.

B. Cycles

When AS B is classified as a customer of AS A and AS C is classified as customer of AS B and AS A is a customer of AS C again, then the algorithm "discovered" a cycle. It normally makes no sense to be the customer of an AS that is a customer of your customer. There are, however, exceptions where it makes sense from the business point of view. If a big provider wants to enter a new market, it becomes a customer of a smaller local AS in order to acquire connectivity to the new market. They do not want to set up a peering, because many other local ASes of similar size would then also like to have a peering with the new important AS. But the big AS has no interest to make all of its paying customers to peers. This is the point where every "natural" algorithm comes to fall, because the assumption of a customer-provider relationship is somehow based on the magnitude of an AS, or its position in the global graph. That a very important AS is the customer of a smaller and lower AS in the global hierarchy is not accounted for in these algorithms which thus produce a P/C relationship.

C. Exchange Points

Taking a closer look at the ASes within a cycle, the following incidents seem to occur the most. Exchange points on level 3 are likely to produce conflicts. An exchange point peers with some AS and announces all the routes it gets from it to everyone. The ASes are seen as peers of the exchange point, although they are paying for the service of this neutral exchange point. Being connected to a layer 3 exchange point means that you do not have full control which of your routes will be announced to whom. Sometimes the exchange point operates servers as well, and needs global connectivity for this reason. So, the exchange point becomes a customer of many of its peers. This leads to the conflict, because the ASes connected to the exchange point should be seen as peers. When the exchange point is on level 2, it is not seen at all and therefore does not produce conflicts at all. Today, most Exchange Points do not provide any layer 3 service, except IP-address allocation for the routers.

D. Common Misinterpretation

Comparing the different approaches by hand, we observe that a few of the ASes produce most of the incongruities. AS-3303 (Swisscom) has many peerings, which are all listed in the macro AS-SWCMPEERS. Especially the rank-based algorithm and the Robban Tool misinterpret about 50% of the peerings of AS-3303 as a P/C relationship. Swisscom peers directly with the route collector of the Route View Project, so these peerings can be seen, but all paths announced to the route collector have the form "AS-3303 $ASPEER$." For the Robban Tool, this implies that AS-3303 is the provider, if the peer does not announce its routes to the route collector. For the rank-based algorithm, if AS-3303 and $ASPEER$ are both not in the residual graph (see Section II-B), it is likely that $ASPEER$ will be interpreted as customer. The Robban Tool can be prevented from doing this misinterpretation by only looking at paths with at least three ASes. This, in turn, breaks if a customer of AS-3303 also peers with the route collector.

Many times, all algorithms infer a customer-provider relationship, but the policy readout of the registry deduces a peering. It may be that the policy is defined as in Table XIII and just one AS has entered its policy, therefore no conflict can be detected. The number of inferable relationships decreases sharply if we just look at the results of the registries where the entry for AS A and the entry for AS B imply the same relationship between the two ASes (i.e., only ASes who both have a registry entry with information about their import and export behavior are taken into account). But the fraction of the relationships inferred differently by the algorithms stays constant.

Mainly, the same ASes seem to always produce conflicts, regardless of whether the relationship is decided from one or two registry entries. The effective number of conflicts per AS if not double checked is higher. But seeing that this AS leads to conflicts in the double-checked case as well, we can conclude that the algorithms misinterpreted the importance of this AS. The main part of conflicts seems to be caused by ASes with open peering policies (e.g. AS-3303, AS-4589).

E. Peerings

Note that peerings cause the most problems as evidenced by the previous sections. When choosing a measure to express importance, one has the problem that not every AS uses the same benchmark to accept another AS as its peer. Here is a cut-out of Level 3 peering policy:
"Level 3 will continue to monitor the market and traffic conditions and accordingly change the peering policy to suit the market and customer needs."[3]

Therefore, we cannot rely on just one definition of importance to decide whether they peer or not. Many other considerations influence the peering decision process: Is there a business competition in the same market or is the peer at an exchange point abroad? Is it an IPv6 or IPv4 peering? Can we improve latency? Do we trust in the abilities of the network operators that they only announce what they should? Will we lose time by supporting the other network operator? Is the other AS expected to grow or to shrink? Do we have special agreements where the peer has to pay if the traffic asymmetry goes beyond a certain ratio?

Because the answers to all these questions are not available to any algorithm, it seems impossible to detect every peering. The majority of the connections seen with only some vantage points are customer-providers relationships, because we have no information about local peerings. Table XI clearly shows that the number of common peers is only about 1.03% of the common links. This is far away from the 5% peerings of algorithm [6] and up to 7.8% peerings plus up to 1.5% siblings of algorithm [5].

Another important aspect is the tier in which the peerings are located. The rank-based algorithm identifies exclusively peerings between ASes in higher tiers. Generally, it is more interesting to know the peerings of the higher levels, because more routes are affected by them. The common peerings of approaches II-A.1, II-B, and II-C are peerings on higher levels.

F. Siblings

The concept of sibling has no origin in the world of Internet Service Providers. An ISP probably does not know that sibling stands for mutual transit agreement. To infer siblings is at least as difficult as peerings. If two ASes only have a mutual backup agreement, how do we catch that moment when this backup functionality takes over? Perhaps we needed to collect BGP update traces at different dates and consider all of them to get a higher chance of inferring mutual backup agreements.

We also know about cases where two "siblings" just announce their peerings to each other, but not the routes of their transit providers. They could be classified as half-siblings to get a more fine-grained and thus more realistic model of their relationship.

V. Conclusion

Algorithms to infer relationships between AS seem to be reasonably accurate in identifying customer-provider relationships. Peerings are more difficult to find. Problems arise when encountering unusual or unexpected cases. Our model of the BGP-Internet graph is a tree with cross connections between nodes on a similar level to represent peerings. Misinterpretation is most likely to occur when higher-level nodes are connected to leaves, ASes have more peerings than customers, exchange points appear on level 3, or ASes have hybrid forms of relationships.

As they stand at the moment, all of the currently available algorithms have individual weaknesses (Section III-E) as well as general problems (Section IV) which hinder us from obtaining a highly accurate view of the AS Internet graph. In fact, we have developed an interactive website which allows the user to "play" with the different weighting mechanisms giving much leeway for tweaking the rank of your favorite AS.

To end on a positive note, introducing more fine-grained types could help to get a more precise image. In addition to just knowing the provider or peer role, it would complete the image of relationships if we could determine for which subsets and for which size of prefixes an ISP acts as provider or peer. The Registry is able to detect special cases, but the information has to be entered. The Registry actually provides views from many vantage points, namely one for every AS entered. No algorithm based on routing tables can have views from all vantage points, due to lacking access to the routing tables of all ASes. Therefore, an up-to-date and correct registry could help to interpret atypical cases and add many of the local peering connections to complete the graph of the Internet hierarchy. If the IRRs would be as accurate as wanted by Georgos Siganos [8] at Nanog, it would be possible to construct a graph far more complete and more precise than any algorithm could do. We join his (and others') demand to practitioners and the related authorities to maintain and use the IRRs.

References

[1] Y. Rekhter and T. Li, *A Border Gateway Protocol 4 (BGP-4)*, IETF, Mar 1995, RFC 1771.

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

[3] CAIDA, "Skitter," http://www.caida.org/tools/measurement/ skitter/as_adjacencies.xml.

[3] from http://www.level3.com/1511.html, before the March 30th, 2004, update.

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

[5] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 6, pp. 733–745, 2001.

[6] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[7] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *Proc. of the ACM SIGCOMM Conference*, 2002.

[8] G. Siganos and M. Faloutsos, "Analyzing BGP policies: Methodology and tool," in *Proc. of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2004.

[9] R. Govindan and A. Reddy, "An analysis of Internet inter-domain topology and route stability," in *Proc. of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 1997.

[10] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery.," in *Proc. of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2000.

[11] Y. Hyum, A. Broido, and K. Claffy, "Traceroute and BGP AS path incongruities," http://www.caida.org/outreach/papers/2003/ASP/, 2003.

[12] H. Chang, R. Govindan, S. Jamin, W. Willinger, and S. Shenker, "On inferring AS-level connectivity from BGP routing tables," in *Proc. of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[13] D. Magoni and J.-J. Pansiot, "Analysis of the autonomous system network topology," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 3, pp. 26–37, 2001.

[14] G. Di Battista, M. Patrignani, and M. Pizzonia, "Computing the types of the relationships between autonomous systems," in *Proc. of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[15] F. Wang and L. Gao, "On inferring and characterizing internet routing policies," in *Proc. of the ACM SIGCOMM and USENIX Internet Measurement Conference (IMC)*, 2003.

[16] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network topology generators: Degree-based vs. structural," in *Proc. of the ACM SIGCOMM Conference*, 2002.

[17] David Meyer, "Route views project," http://antc.uoregon.edu/route-views/.

[18] Merit Network Inc., "Internet routing registry," www.irr.net.

[19] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra, *Routing Policy Specification Language (RPSL)*, IETF, Jun 1999, RFC 2622.

[20] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," http://www.cs.berkeley.edu/˜sagarwal/research/BGP-hierarchy/.

[21] Pascal Gloor, "Netlantis," www.netlantis.org.

[22] Ripe NCC, "Routing information service raw data," data.ris.ripe.net.

[23] Merit Network Inc., "Routing assets database," www.radb.net.