Inferring AS-level Internet Topology from Router-Level Path Traces

Hyunseok Chang
Department of EECS
University of Michigan
Ann Arbor, MI 48109-2122
hschang@eecs.umich.edu

Sugih Jamin*
Department of EECS
University of Michigan
Ann Arbor, MI 48109-2122
jamin@eecs.umich.edu

Walter Willinger
AT&T Labs-Research
180 Park Ave.
Florham Park, NJ 07932-0971
walter@research.att.com

Abstract

A number of recent studies characterize AS-level topology of the Internet by exploiting connectivity information contained in BGP routing tables. In this paper, we present an alternative method for discovering AS connectivity by inferring individual AS connections from the Internet's router-level topology. This methodology has several advantages over using BGP routing tables. First, it allows us to obtain AS-level connectivity information at a finer granularity (e.g., multiple connections between a pair of ASs); second, we can discover ASs aggregated in BGP routing tables; and third, we can identify AS border routers, which may allow us to further characterize inter-AS connections. Border routers have multiple interfaces, each with an address in a potentially different AS. A major challenge of our approach is then to properly map border routers to their corresponding ASs. To this end, we present in this paper several mapping rules and heuristic for inferring the ASs of border routers. We report on results showing the effectiveness and validity of these rules and heuristic.

Keywords: Internet topology, AS border routers

1 Introduction

The immense commercial success of the World Wide Web has to a large part been responsible for the sustained growth that the Internet has experienced, not only in terms of traffic volume (total number of packets transmitted) but also in terms of the sizes of the individual subnetworks that make up the Internet. Maintaining a functioning Internet under such growth conditions requires judicious network engineering and involves constant hardware upgrades and a continuous expansion of the underlying Internet connectivity. Researchers have been interested in how such individually and locally made engineering efforts to optimize network performance are reflected in the current global Internet connectivity. A number of recent empirical studies [9, 5, 15, 17] have tried to characterize the spatial properties and the temporal development of the Internet topology.

For the purpose of this paper, Internet topology is defined on two different abstract levels: router-level and Autonomous System (AS)-level. (An AS is a network under a single administration domain. ASs connect to each other through border routers. Within an AS, the network could be further decomposed into subnetworks connected with internal routers.) Therefore, topological adjacency in the Internet context can be defined either between any two physical routers or between two ASs. However, given the immense scale of the Internet, its continuous evolution, and decentralized administration, discovering the Internet topology is not a trivial task.

Empirical studies attempting to characterize the Internet AS-level topology have exploited connectivity information contained in BGP routing tables. Since BGP routing tables hold a set of complete AS-level routing paths for each destination network, we can elicit AS-level connectivity information from them. In contrast, discovering *router-level* adjacency must be based on active path tracing, whereby a path to a certain destination is incrementally discovered by hop-by-hop probing from a probing host.

*This project is funded in part by NSF grant number ANI-0082287 and by ONR grant number N000140110617. Sugih Jamin is further supported by the NSF CAREER Award ANI-9734145, the Presidential Early Career Award for Scientists and Engineers (PECASE) 1999, and the Alfred P. Sloan Foundation Research Fellowship. Additional funding is provided by AT&T Research, and by equipment grants from Sun Microsystems Inc. and Compaq Corp.

In this paper, we present a method for discovering AS connectivity by inferring individual AS connections from the Internet's router-level topology. This methodology has several advantages over using BGP routing tables, where the latter has been the method-of-choice in practically all empirical studies to date. First, our methodology enables us to obtain AS-level connectivity information at a finer granularity (e.g., multiple connections between a pair of ASs) which, in turn, can provide physical-level insight into AS peering relationships. Second, using our approach, we are able to get around the intrinsic limited-view problem of BGP routing tables: to contain the growth of BGP routing tables, upstream ASs aggregate the advertisements of several downstream ASs with contiguous address ranges into one advertisement of a larger AS [2]. Also, with existing BGP route advertisement schemes, where not all possible AS paths are circulated on the Internet due to policy routing [8], it is unavoidable that BGP routing tables don't capture all existing AS links. Finally, being able to identify AS border routers, we may further characterize inter-AS connections. Tangmunarunkit et al. [18] have also attempted to infer AS connectivity from router-level topology. Our approach complements theirs (see Section 3.2 for a comparison of their method against ours).

Since border routers have, by definition, multiple interfaces, each with an address in a potentially different AS, a major challenge of our approach is to properly map border routers to their corresponding ASs. To this end, we present in this paper several mapping rules and heuristic for inferring the ASs of border routers and report on results showing the effectiveness and validity of these rules and heuristic. Finally, we compare the resulting AS-level topology with the one originating from BGP routing tables.

2 AS mapping of traceroute path

Given a router-level path trace generated by `traceroute` tool [11], a natural way to infer AS-level connectivity is to determine the AS of each router hop and to extract AS adjacency information from the resulting sequence of ASs associated with the router-level path. Our first step is therefore to construct an AS mapping table that would allow us to determine the AS to which a certain address block is allocated. We then generate AS paths from a set of traceroute paths by using this table, and compare them to the BGP-derived AS-level topology.

2.1 Default AS mapping

The AS mapping table matches any existing address prefix (or route) to its origin AS. Since inferring AS-level topology depends crucially on this AS mapping table, maintaining an up-to-date table with respect to a collected router-level topology is of highest priority. The mapping table construction is based on two sources of information: BGP routing table and the Internet Routing Registry (IRR) [13].

The Oregon route server [14], which as of Mar. 2001, maintains BGP sessions with 37 other BGP routers on the Internet, makes the daily updated version of its BGP routing table available at NLANR [6]. Unlike a conventional BGP routing table, however, it retains, for each advertised route, not only the 'best' AS path but also all the other AS paths that it has learned from its peers. Therefore, it is essentially a collection of BGP routing tables of all its peers, which include most of the existing major ASs. We use this AS path information to trace the AS to which any existing address block is allocated. Once appropriate AS paths are selected, it is straightforward to extract from the BGP tables the prefix-AS tuples that constitute the AS mapping table. The set of prefix-AS tuples obtained from one week's worth of BGP table snapshots (from 10/15/00 to 10/21/00) were merged with more recent data overriding less recent one. Sometimes, a single prefix was mapped to more than one AS from the BGP routing tables of the same day because of hardware failure or misconfigurations of some ISPs. In such cases, we removed the prefix for the day.

It should be noted that a certain address prefix belonging to a legitimate AS could be hidden from other ASs' BGP routers, e.g., when route aggregation occurs [2]. To expand our route view beyond this aggregation boundary, we include in our AS mapping table the route origin information from IRR [13]. RIPE database of IRR is fairly up-to-date and its routing policy entries are actively used by most ISPs in Europe to generate route connection filters. Some European Network Exchange Points [12] require members to keep their AS information in IRR up-to-date. In our AS mapping table, all the RIPE entries and any other database entries which have been updated within the past 6 months are included. When the two sources of information were in conflict, the BGP-originating information is given priority over the IRR data (because the former is more likely to have up-to-date information).

Given the AS mapping table, the administration domain of a router interface can be determined in a straight-

Table 1: Default Mapping of Traceroute Paths

	# of paths	Percentage
Total	31,144	100%
Unknown AS	3,866	12.4%
Routing loop	1,450	4.7%
AS-valley	675	2.2%
Dubious total	5,362	17.2%

forward manner by finding the longest address prefix that matches the address. We call AS mapping based on the longest prefix matching *default AS mapping*. We evaluate default AS mapping using a set of traceroute paths from [3] against AS mapping table collected during the same period.

2.2 Analysis of default AS mapping

Table 1 summarizes three different kinds of irregularities that we found in our AS mapping result. First, some router interface addresses do not have a corresponding address prefix in our AS mapping table (Unknown AS). Second, some AS paths contain duplicate ASs in a routing loop, e.g., -A-B-A-C-, which should be avoided by loop-free path vector attribute of BGP protocol [16] (Routing loop). Finally, in some cases, a fairly "small" AS is seen between two reasonably "large" ASs (AS-valley).[1] Unless there is a policy routing error, such situations should not occur in current *valley-free* policy routing scheme [7]. In total, about 17% of the traceroute paths that we tested yield dubious AS mapping result or failed AS mapping. We found 4,161 (13.3%) AS links on the inferred AS map not found in the Oregon BGP routing table. We cannot consider these links erroneous; for example, they could simply be invisible to the Oregon BGP route server, either because of route aggregation or because of BGP export policies that constrain re-advertisement of peering routes [8].

Assuming that our AS mapping table is fairly accurate, we next investigate the causes of these incorrect AS paths. One possible culprit is the path trace mechanism of the traceroute tool. The discovery of a router-level path by the traceroute tool is based on retrieving the source address of ICMP error messages (Time-Exceeded or Port/ Protocol Unreachable, etc.) sent by intermediate routers when UDP probe packets trigger such errors on them. According to the ICMP implementation practice of current IP version 4 routers [1], when a UDP packet is sent to a certain router interface and triggers any type of error to be reported to the sender, the router will respond with an appropriate ICMP error message whose source address is set to the address of the *outgoing interface* transmitting the message. That is, what we see as the address of each router hop in the traceroute path belongs to the neighboring network through which a corresponding router has emitted its response.

Now let's consider what kind of scenario could happen with *AS border routers* that are implemented in such a way. The border routers of each AS interface with other ASs through either direct private physical connections, or at a public exchange point (IXP) owned by a third party [4]. Therefore, each border router could have an interface whose IP address is assigned either from its neighboring ASs' address space (private peering) or from the address space of the third party in question (public peering). Figure 1 shows five routers (R2 to R5 are border routers) connected as described. AS4 is a public exchange point which provides hardware switches (ATM or FDDI) and its own address space for its member ASs. If we assume that each router sends back ICMP messages through its top interface, then the AS path a prober in AS1 sees is "AS1-AS3-AS2-AS4", and not the correct "AS1-AS2-AS5".

In the following sections, we propose a method that attempts to correctly infer AS-level connectivity from router-level topology in the presence of AS border routers.

[1] Here we predict the size of an AS by its BGP-derived outdegree. We consider those of ASs with outdegree less than 4 as "small" ASs, and those with outdegree larger than 30 as "large" ASs.

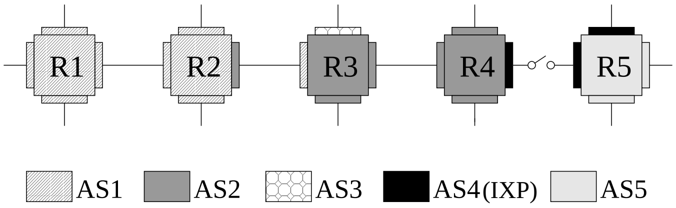


Figure 1: AS mapping of a traceroute path

Table 2: Default-Routed vs. Source-Routed Probing

	Default-routed	Source-routed
Total # of probes	319,511	597,303
Not responded	116,909 (36.6%)	461,304 (77.2%)
Responded with no alias	170,999 (53.5%)	104,534 (17.5%)
Responded with alias	31,603 (9.9%)	31,465 (5.3%)
Total # of alias pairs	39,117	

3 Discovery of AS border routers

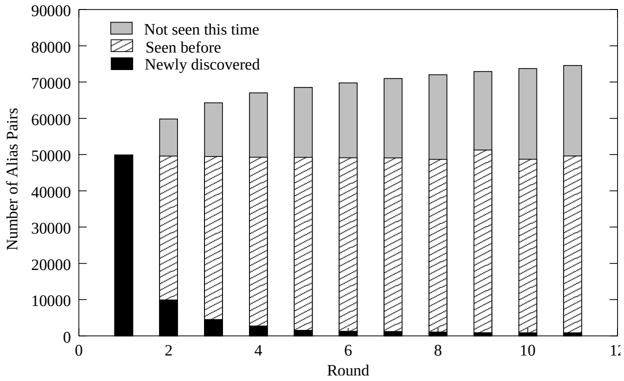
The first step toward correct AS mapping in the presence of AS border routers is to tell which routers are AS border routers. In order to distinguish an internal router from a border router, we need to know the addresses of all the routers' physical interfaces, or so-called *router aliases*. If a router has its interface addresses assigned from multiple administration domains, it is a border router.

3.1 Router alias probing

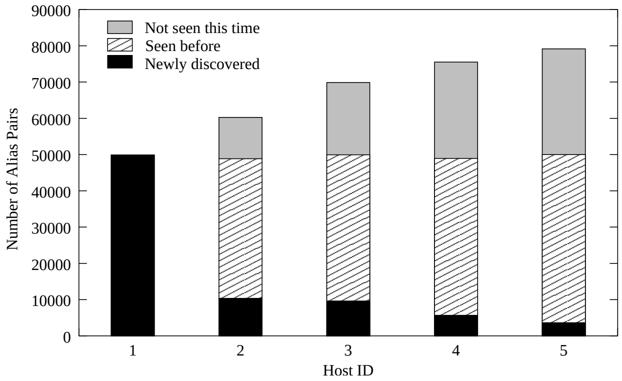
The discovery of router aliases has recently been attempted by the Mercator software [10]. Mercator is an Internet mapper program aimed at efficiently discovering router-level links by traceroute-style hop-limited probes. Alias resolution is necessary to construct an accurate router-level topology from the interface adjacency information that Mercator discovers. The alias resolution by Mercator is based on the ICMP source addressing mechanism that was described in Section 2. If the probed interface and the one responding with ICMP messages have different addresses, those two can be considered as two different aliases belonging to a single router.

We conducted our own alias probing experiment, starting in mid-October, 2000, and lasting for one month. Our router prober emitted UDP probe packets to 350,000 router interfaces found in the Internet mapping database [3] as our probing destinations and inspected the source address of the returned responses. For each destination IP address, the prober sent up to 3 UDP probe packets. It took approximately one day to probe all 350,000 IP addresses. Since the Internet topology continues to evolve, ideally we would like to send all the probes at the same time so as to capture a current snapshot of the Internet. Sending the probes over a longer period of time could mean that we are looking at different instances of the Internet topology at the start and the end of the probing period. Unfortunately, due to heightened security awareness on the part of network administrators, our probe packets were frequently mistaken for network intrusion attempts. Increasing the probing frequency simply raised the number and level of alarms. Below we elaborate on other problems we encountered during our probing.

Due to firewall restrictions or router policies, not all of our probe packets made it to their destinations. To alleviate the consequences of dropped probe packets, we supplemented our probing experiment as follows. First, we repeated the probing process in multiple rounds with the same target interfaces until we saw diminishing returns of returning probes. In order to reduce the likelihood of triggering security alarms within the probed networks, the probing rounds were spaced one or two days a part, and the sequence of the probing target list was randomized each time a new round was initiated. Second, to map around policy routing restrictions, we used both source-routed packet probing and performed our probing from multiple geographically and topologically dispersed hosts. With source-routed probing, we used the source routing option of IPv4 to force our probe packets to traverse certain routers [10].



(a) Multiple Rounds From A Single Host



(b) A Single Round From Multiple Hosts

Figure 2: Alias Probing

Table 3: Alias Table Statistics

	All Routers	Border Routers
Total # of distinct alias pairs	93,372	
Total # of routers	36,813	10,547
Total # of interfaces	107,445	33,839
Max. # of interfaces per router	188	132
Min. # of interfaces per router	2	2
Avg. # of interfaces per router	2.92	3.21

This way, we hoped to increase the probability that our probes will get through to a target interface, which would otherwise be unreachable from our original probing host. Unfortunately, since most regular IP packets do not use source routing, our unconventional probe packets raised a large number of security alarms. The motivation for using multiple probing hosts was the same as that of using source-routed probes, but without relying on the source-routing option.

Table 2 compares the efficiency of two alias probing experiments: Default-routed probing vs. Source-routed probing. When probed, router interfaces either (1) did not respond (due to packet loss or unreachable routes) (2nd. row), (2) responded with the same probed address (3rd. row), or (3) responded with a different alias (4th. row). Although the source-routed probing enables one to control routing paths to a certain extent on a single host, we can see that it is not very efficient in terms of alias gathering. Apparently, due to security reasons, many routers refuse to respond to source-routed packets and even do not forward them.

Figure 2 shows the number of router aliases discovered by running multiple-round and multiple-host probings. The probing result from each probing session (per-round or per-host) consists of a set of alias pairs. Each bar in the Figure 2(a) represents the *cumulative* number of alias pairs we have discovered up to the number of rounds along the x-axis. Each bar consists of three components: "Newly discovered" is the number of alias pairs discovered in the current round but not in any previous rounds; "Seen before" is the number of alias pairs observed in both the current round and one or more previous rounds; "Not seen this time" denotes the number of alias pairs observed in at least one of the previous rounds but not in the current round. Similarly for Figure 2(b), except the x-axis counts the number of hosts, instead of rounds, used in the probing experiments. As expected, the number of newly discovered alias pairs decreases monotonically with the number of probing experiments. However, the multi-round alias probing was still able to find 800 or so new alias pairs even after the apparent saturation point at the 9th round. This could be attributed to the inevitable routing dynamics and the continuous growth of the Internet. In case of multi-host alias probing, we continue to discover a nonnegligible number of new alias pairs even after the fifth probing host.

Based on these probing results, we created an alias table as follows. We first merged the alias pairs by transitive closure into disjoint alias groups. Each group represents the aliases of a single router. Then the alias groups from the various probing hosts were again merged by transitive closure to obtain the final alias table. As listed in Table 3, the 107,445 interfaces discovered make up 93,372 alias pairs, which by transitive closure are attributed to 36,813

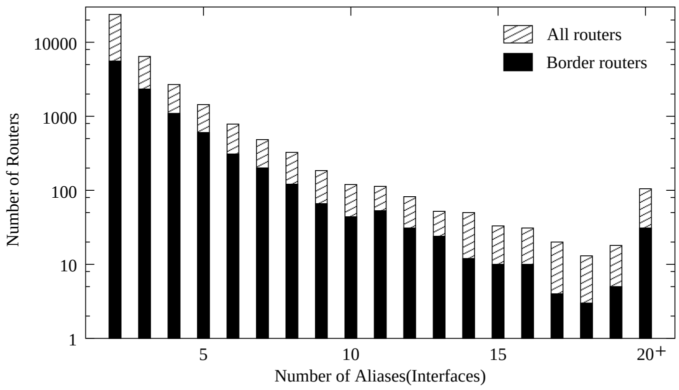


Figure 3: Router Alias Distribution

routers. Of these, 10,547 have their addresses in multiple administrative domains and are hence identified as border routers (also represented as the solid portion of the bars in Figure 3).

3.2 Mapping of AS border routers

In order to identify the inter-AS links connecting two different AS border routers, we need to correctly determine in which administration domain each of those border routers that were previously discovered resides. However, given the limited number of the interfaces that we were able to collect for each border router,[2] it is by no means simple to correctly pinpoint the AS of the router. In [18], they mapped each AS border router by picking the AS that is most frequently assigned to its interfaces. Here, we attempt to resolve the AS of a border router based on the following two observations:

1. If a border router R is known to have an outgoing interface mapped to an AS X, the router could belong to either X or any of the ASs which peer with X. For example, router R2 in Figure 1 could belong to AS1 or AS2.

2. If a border router R has an adjacent router which is known to be in AS X, then the router could belong to either X or any of X's peers. For example, if we know that router R2 in Figure 1 peers with router R1 which has been determined to be in AS1, R2 could be in AS1 or AS2, which peers with AS1.

The first observation implies that if a given router has a set of known interfaces and they are default-mapped to $\{AS_1, \ldots, AS_N\}$ (N>1), the router must belong to: $\{AS_1, peer(AS_1)\} \bigcap \{AS_2, peer(AS_2)\} \bigcap \ldots \ldots \bigcap \{AS_N, peer(AS_N)\}$.[3]

1. Intersection rule: If the above intersection set reduces the candidate set to a single element set, the sole AS in the candidate set is then adopted as the administrative domain of the given router.

If the candidate set has not been reduced to a single element, we apply the second observation to further reduce the size of the candidate set. However, the second observation is applicable only if we have information about router-level topology. In our case, we used the traceroute path data of [3] that were collected during the same one month's period when we conducted our alias probing experiment. The resulting adjacency graph involves 80,457 router interfaces, of which about 89% can be default-mapped. For each of the remaining unmapped router interfaces, by the second observation above, if it has a set of neighboring interfaces already mapped to $\{AS'_1, \ldots, AS'_M\}$, its candidate ASs must belong to: $\{AS'_1, peer(AS'_1)\} \bigcap \{AS'_2, peer(AS'_2)\} \bigcap \ldots \bigcap \{AS'_M, peer(AS'_M)\}$. Hence:

Extended intersection rule: If two candidate sets (one from the original intersection rule and the other from the above intersection) have a single element in common, then the corresponding AS is adopted as an administration domain of a given border router.

[2] As shown in Table 4, the average number of interfaces discovered per border routers is about 3.

[3] One exception, however, is that if a certain border router has an interface assigned from any public exchange point (IXP), then it does not mean that the router's AS is peering with the IXP, which is simply a third party helping with connecting the AS to other ASs. Therefore, those interfaces belonging to any existing IXPs should not be considered in this scenario.

Table 4: AS mapping of border routers

	# resolved	Avg. # of interfaces
Intersection	1,289	3.82
Majority	781	7.71
Remaining	8,477	2.70
Total	10,547	3.21

Unfortunately, because of the limited alias information and the richness of AS peering relationships, the above two intersection rules reduced only a relatively small number of candidate sets to a single element set. In total, only 1,289 border routers out of 10,547 were resolved. For the remaining unmapped routers, the following rule is applied.

2. Majority rule: If a given router is known to have more than four interfaces and more than two thirds of them are mapped to one AS, that AS is chosen as the router's administrative domain.

The result of our AS mapping of border routers is summarized in Table 4. After applying the intersection and majority rules, we were able to reduce the number of unmapped nodes in the router-level adjacency graph from 8,942 (11%) to 5,617 (7%).

4 Construction of AS overlay map

In the previous section, we attempted to infer border routers' administrative domains. However, we were only able to resolve about 20% of the discovered border routers.

In this section, we construct an *AS overlay map* on top of the router-level adjacency graph. The AS overlay map, together with its underlying router-level adjacency graph, allows us to identify all inter-AS links. Note that the router-level adjacency graph was created from a set of traceroute paths originating from a single probing host. Therefore, it is essentially a *directed* graph, where each node corresponds to a router interface and each directed edge corresponds to a router-level link connecting one node to its valid next-hop node. As we will see below, retaining such node precedence in the graph allows us to predict a node's AS in a more informed manner.

Although we would expect the resulting directed graph to be a simple rooted tree, whose leaf nodes are a set of destination IP addresses, it turns out that this is not the case. Due to policy routing, it is frequently observed that there is more than one routing path to an *intermediate* backbone router, which means that each node in the graph can be preceded by more than one nodes. We even found instances where a set of connected nodes formed a closed loop or cycle. Figure 4 plots in log-log scale the number of parent and children nodes that each node in the directed graph has, against their frequency.[4]

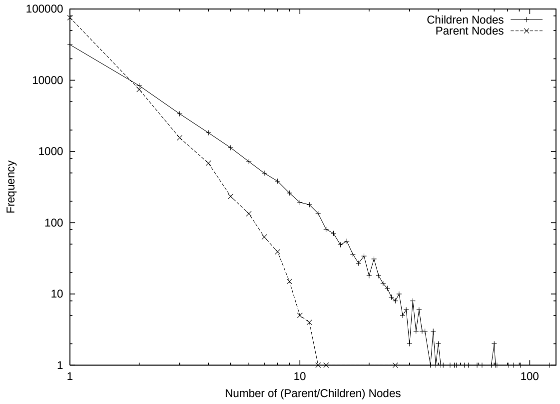


Figure 4: Frequency vs. Parent (Children) Nodes

[4]Max. number of parent nodes and children nodes are 26 and 122 respectively.

The AS overlay map can be constructed by first mapping each router-level node to its AS, and then coalescing a cluster of nodes mapped to the same AS into a single AS-level node. Therefore, we first have to properly assign all the remaining unmapped nodes (7%) in the adjacency map to their ASs. For this purpose, we introduce a "hole-filling" heuristic, where a *hole* is defined to be an unmapped connected component in the adjacency map surrounded by mapped nodes. The size of a hole is the number of nodes that make up the hole. Table 5 shows that about 97% of holes are of size one or two. The hole of size 122 is an outlier caused by an address prefix not found in the AS mapping table.

4.1 Hole-filling heuristic

For some unmapped node belonging to a given hole, it has a set of parent nodes and children nodes, which are mapped to two different AS sets $\{AS_1, \dots, AS_p\}$ and $\{AS'_1, \dots, AS'_c\}$ respectively. Some of the parent or children nodes could also be unmapped unless the size of the hole is one. We consider two rules to fill holes in the adjacency map: inheritance rule and propagation rule. The inheritance rule states that a given node *inherits* its parent node's AS. The propagation rule allows a child node's AS to be *propagated* up to a parent node. Basically, our hole-filling heuristic attempts to intuitively interpolate the administrative domain of the holes in our router-adjacency graph.

We apply the inheritance rule to a given unmapped node N if all its parent nodes are mapped to a single AS ($p = 1$) and its children nodes are mapped to more than one ASs ($c > 1$). This is based on our expectation that the parent AS in this case would provide a common route for a number of different children ASs, through the node N. For the same reasons, the propagation rule is applied if the parent nodes are mapped to more than one ASs ($p > 1$) and all the children nodes are mapped to a single AS ($c = 1$).

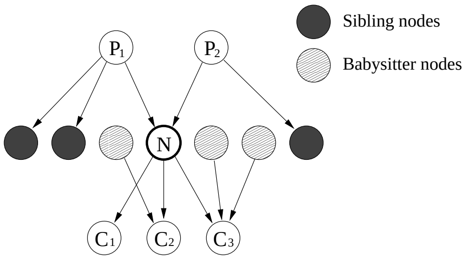


Figure 5: Sibling Nodes vs. Babysitter Nodes

In the case that $p = c = 1$ and the parent AS is different than the children AS, we look at the sibling nodes and the babysitter nodes of N. The sibling nodes of N are the nodes that share the same parent node with N, and the babysitter nodes are the ones that share the same child node with N. Then, we pick an AS, which is responsible for a majority of the sibling and babysitter nodes, and compare it with the parent AS and the child AS. If the picked AS can be matched with either of them, we apply either inheritance rule or propagation rule so that N is mapped to the selected AS.

If $p = 0$ or $c = 0$, the AS that a majority of the parent (children) nodes belong to, is inherited by (propagated to) N, unless N is a terminal node.

Finally, if N is a terminal node, we simply stay with the default AS mapping, having no better alternative.

Note that our heuristic can be applied in multiple rounds to map holes of size larger than one. Of course, the mapping accuracy will decrease as the number of rounds increases. However, as shown in Table 6,[5] the number of nodes that are mapped in multiple rounds decays exponentially fast with the number of rounds. Eighty five nodes remain unmapped even if we run the heuristic for more than four rounds.

4.2 Performance of hole-filling heuristic

As mentioned earlier, most network service providers are very sensitive to any kind of probing activities on their networks. It is practically impossible to get actual configuration information on all the AS border routers (e.g., how

[5]The 3,731 nodes in the initial configuration are the non-terminal nodes among the 5,617 originally unmapped nodes.

Table 5: Hole Size Distribution

Size	1	2	3	4	5–22	122
Frequency	2,825	229	60	17	9	1

Table 6: The Profile of Hole-filling Heuristic

	# of mapped nodes		# of unmapped nodes
	Inheritance rule	Propagation rule	after current round
Initial Conf.	-	-	3,731
Round #1	2,465	936	330
Round #2	151	80	99
Round #3	7	5	87
Round #4	2	0	85

many interfaces they have and what AS they actually belong to) to verify our heuristic. Instead, we perform the following verification.

First, we assume that every node which does not belong to any known border router is *correctly* mapped by the longest prefix matching. We then randomly generate holes of size one within the correctly mapped regions and check whether or not our hole-filling heuristic can correctly map these newly created holes. We also compare our heuristic to several other methods for inferring a router's AS.

- *I&P*: The hole-filling heuristic
- *I-only*: Always inherits the AS that a majority of parent nodes belong to.
- *P-only*: Always propagates the AS that a majority of children nodes belong to.
- *Majority*: Picks an AS that a majority of neighboring nodes belongs to.
- *Random*: Randomly picks an AS that any of neighboring nodes belongs to.

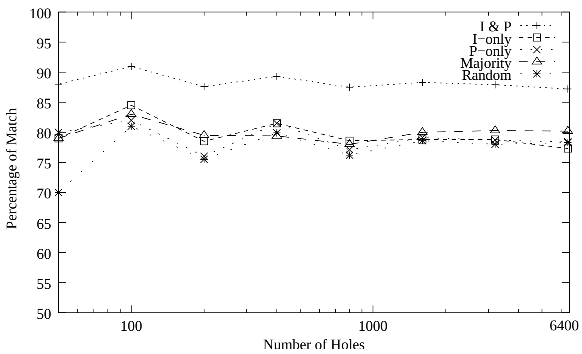


Figure 6: Performance of Hole-filling Heuristic

We tested our heuristic with a varying number of holes (from 50 to 6,400). The testing range is based on the number of initially unmapped non-terminal nodes in the graph before we applied our heuristic (i.e., 3,731). For each case, we tested twice and averaged the two results. As seen in Figure 6, the hole-filling heuristic clearly does better than the other methods, irrespective of test case or number of holes.

Table 7: Number of Inter-AS Links

AS map	# of ASs	# of links
Complete BGP map (BGP)	9,488	20,971
Reduced BGP map (BGP-)	5,640	13,995
Overlay map (Overlay)	5,854	8,580

Table 8: Outdegree Distribution of Overlay-only ASs

Outdegree	1	2	3	4	5	8	Total
Frequency	159	18	6	1	1	1	186

4.3 Coverage of AS overlay map

We want to determine what portion of the existing Internet connectivity is covered by our inferred AS overlay map. For this purpose, we first constructed an AS-level topology derived from one month's worth of BGP routing tables (from 10/14/00 to 11/15/00) [6]; we call this the *complete BGP map*. We then reduced this topology by discarding the ASs (together with their inter AS-links) that were not found in the AS overlay map; we call the resulting topology the *reduced BGP map*. While the complete BGP map allows us to determine what portion of existing ASs our AS overlay map has discovered, the reduced BGP map characterizes the connectivities of the AS overlay map against the BGP view that spans the same set of ASs.

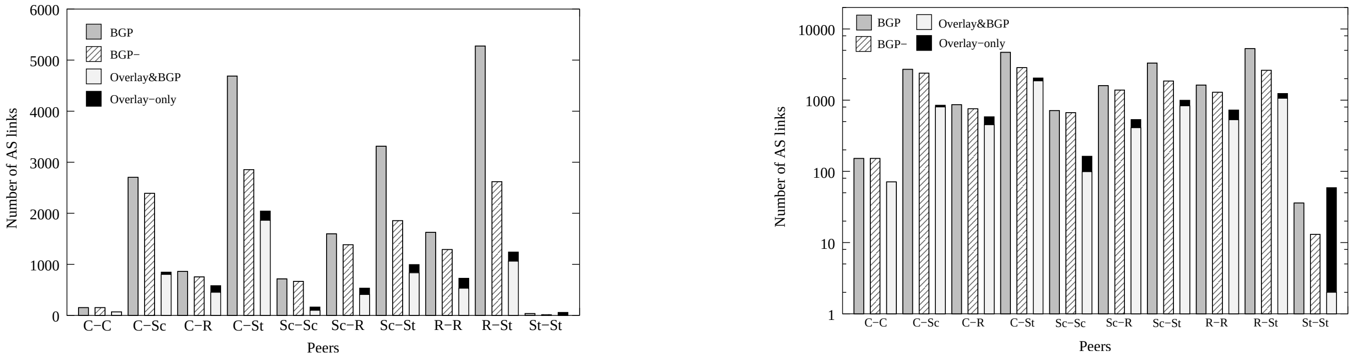
Table 7 summarizes the AS connectivities of (1) the complete BGP map, (2) the reduced BGP map, and (3) our AS overlay map. According to the table, our overlay map covers about 62% of existing ASs (comparing the number of ASs in BGP and Overlay) and 61% of AS-level connectivity of those ASs (comparing the number of links in BGP- and Overlay). Note that there are 186 ASs that are only found in the overlay map but not in the complete BGP map (Overlay-only AS). According to Table 8, a majority of those ASs has an outdegree of one in the AS overlay map, suggesting that they are most likely customer stub ASs whose presence is hidden by their provider ASs in most BGP routing tables. The newly found AS with outdegree 8 is in fact connected to only three provider ASs whose outdegrees are 148, 146 and 48 (two of these turned out to belong to the same company as the AS itself). That makes the route aggregation more probable.

Next, we look at the number of links that connect the ASs; we categorize the ASs into four different hierarchy levels: *Core* (C), *Subcore* (Sc), *Regional* (R), *Stub* (St). Core ASs are those ASs that constitute a maximum clique of BGP-derived AS map; Subcore ASs are those ASs which connect to more than two of the core-ASs that have been discovered. The regional ASs are the remaining transit ASs, i.e., those ASs that show up in the middle of one or more AS paths in the BGP routing tables; and stub ASs are those that do not provide transit service.

In Figure 7, "Overlay&BGP" refers to the AS links in the overlay map that are also found in the complete BGP map, "Overlay-only" refers to AS links that are not found in the complete BGP map. The same result is shown in two different scales in the number of AS links (y-axis) for clarification. The comparison of the reduced BGP map and the overlay map shows that the peering links among backbone networks (core ASs or subcore ASs) is less visible in the overlay map. This feature can be explained in terms of the relatively high degree of connectivity among the backbone networks, a property that cannot be fully observed from a single site within the Internet.

Table 9: Four-level AS hierarchy

Hierarhcy	Core	Subcore	Regional	Stub	Total
# of ASs	18	590	1,426	7,454	9,488
Percentage	0.19%	6.22%	15.0%	78.6%	100%



(a) Linear scale in y-axis

(b) Log scale in y-axis

Figure 7: Distribution of Inter-AS Links

4.4 Outdegree Distributions

Figure 8 plots on the y-axis the number of connections (outdegree) each AS has to its neighboring ASs; on the x-axis is the rank of the AS by its number of connections, sorted in decreasing number of connections. Aside from the BGP and Overlay maps, we also show on the same figure the physical connectivity of each AS, i.e., the total number of router-level links connecting an AS to its neighboring ASs in the overlay map. Both the AS outdegree and the physical connectivity of the overlay map closely follow the well-known outdegree frequency power-law in [5].

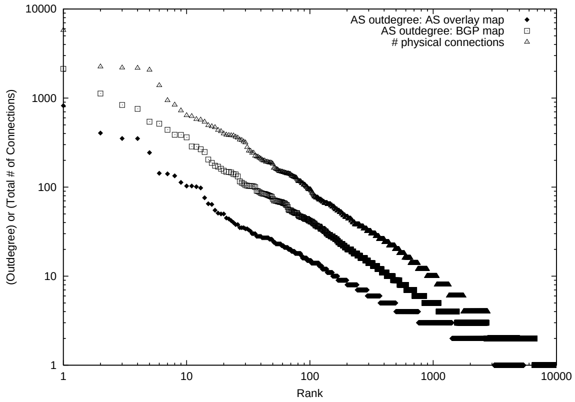


Figure 8: Rank vs. AS outdegree & Physical Connectivity

5 Conclusions

In this paper, we present an alternative method for discovering AS-level Internet topology. Not only is our method capable of compensating for the intrinsically limited view provided by BGP routing tables, but it also allows us to gain important insight into actual Internet connectivity at the AS level; e.g., how densely ASs are paried with each other. Our method requires a complete, or at least an up-to-date, router-level topology. Due not only to the sheer number of probes that must reach all existing networks in the topology discovery process, but also to the heightened security awareness of network administrators who tend to view legitimate network discovery efforts as malicious intrusion attempts, we found that topology discovery efforts consisting of simple path-tracing to greedily map network links to be not practical.

6 Acknowledgements

We thank Ramesh Govindan for offering us nonanoymized Internet map data and valuable feedback during our study. We also thank Andrew Adams, Bengt Ahlgren, Pavel Curtis, Ramesh Govindan, Ed Knightly, and Jörg Liebeherr for allowing us to run our prober on their sites.

References

[1] F. Baker. RFC 1812: Requrements for IP version 4 routers, June 1995.

[2] E. Chen and J. Stewart. RFC 2519: A framework for inter-domain route aggregation, Feb 1999.

[3] B. Cheswick. Internet mapping project. http://www.cs.bell-labs.com/who/ches/map.

[4] B. Chinoy and T. Salo. Internet exchanges: Policy-driven evolution. *Harvard Workshop on Co-ordination of the Internet*, Sept 1996.

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *Proceedings of ACM SIGCOMM '99*, August 1999.

[6] National Laboratory for Applied Network Research. Raw routing table information. http://moat.nlanr.net/Routing/rawdata.

[7] L. Gao. On inferring autonomous system relationships on the Internet. In *Proc. IEEE Global Internet Symposium*, Nov 2000.

[8] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Proc. ACM SIGMETRICS*, June 2000.

[9] R. Govindan and A. Reddy. An analysis of Internet inter-domain topology and route stability. In *Proceedings of IEEE Infocom*, 1997.

[10] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *Proceedings of IEEE Infocom*, April 2000.

[11] Lawrence Berkeley National Laboratory. Traceroute utility. ftp://ftp.ee.lbl.gov/traceroute.tar.Z.

[12] The London Internet Exchange (LINX). Memorandum of understanding. http://www.linx.net/joining/mou.thtml.

[13] The Merit Network. Internet routing registry database. ftp://ftp.radb.net/routing.arbiter/radb/dbase/.

[14] University of Oregon Route Views Project. http://www.antc.uoregon.edu/route-views/.

[15] P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, and D. Estrin. On characterizing network topologies and analyzing their impact on protocol design. Technical Report 00-731, Computer Science Dept., Univ. of Southern California, 2000.

[16] Y. Rekhter and T. Li. RFC 1771: A border gateway protocol 4 (BGP-4), Mar 1995.

[17] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topologies, power laws, and hierarchy. Technical Report 01-746, Computer Science Dept., Univ. of Southern California, 2001.

[18] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The impact of routing policy on Internet paths. In *Proceedings of IEEE Infocom*, April 2001.