On Inferring and Characterizing Internet Routing Policies

Feng Wang
University of Massachusetts - Amherst

On Inferring and Characterizing Internet Routing Policies

Feng Wang
fewang@ecs.umass.edu

Lixin Gao
lgao@ecs.umass.edu

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst
MA 01002, USA

ABSTRACT

Border Gateway Protocol allows Autonomous Systems (ASs) to apply diverse routing policies for selecting routes and for propagating reachability information to other ASs. Although a significant number of studies have been focused on the Internet topology, little is known about what routing policies network operators employ to configure their networks. In this paper, we infer and characterize routing policies employed in the Internet. We find that routes learned from customers are preferred over those from peers and providers, and those from peers are typically preferred over those from providers. We present an algorithm for inferring and characterizing export policies. We show that ASs announce their prefixes to a selected subset of providers. The main reasons behind the selective announcement are the traffic engineering strategy for controlling incoming traffic. The impact of these routing policies might be significant. For example, many Tier-1 ASs reach their (direct or indirect) customers via their peers instead of customers. Furthermore, the selective announcement routing policies imply that there are much less available paths in the Internet than shown in the AS connectivity graph. We hope that our findings will caution network operators in choosing the selective announcement routing policy for traffic engineering. Finally, we study export policies to peers and find that ASs tend to announce all of their prefixes to other peers. To the best of our knowledge, this is the first study on systematically understanding routing policies applied in the Internet.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

General Terms

Measurement, Performance

IMC'03, October 27–29, 2003, Miami Beach, Florida, USA.

Keywords

Routing Policies, BGP, Traffic Engineering

1. INTRODUCTION

The Internet connects thousands of Autonomous Systems (ASs) operated by many different administrative domains such as Internet Service Providers (ISPs), companies and universities. Routing between ASs is determined by the interdomain routing protocol, Border Gateway Protocol (BGP) [1]. A key feature of BGP is that it allows ASs to adopt diverse routing policies to control the selection of routes and propagate reachability information to other ASs. For example, a multihomed AS can control the in-bound traffic link by propagating prefixes to a subset of its providers only. Therefore, the prefix can be reached only through the subset of its providers. As a result, connectivity does not mean reachability in the Internet and the extent of the reachability is determined by both connectivity and routing policies. Although a significant number of studies have been focused on the Internet topology [2][3][4][5], little is known about what routing policies network operators employ to configure their networks.

Understanding routing policies applied in the Internet has several implications. First, it is important to have a global view of the routing policies applied. Clearly, each ISP has information about its own routing policies. However, many ASs are unwilling to reveal their routing policies to others. Furthermore, the routing information stored in Internet Routing Registry (IRR) [9] is either incomplete or out-of-date. Therefore, there is no global view of the typical routing polices configured in an AS. Second, the global view of routing policies might have implications on important properties of Internet. The connectivity in the Internet does not mean reachability since routing polices might lead to less available paths. Moreover, this can lead to implications on robustness of the Internet. Third, being able to infer routing polices of other ASs might allow an AS to perform traffic engineering effectively. To control traffic flow, network operators can change their routing policies to shift traffic load among multiple candidate routing paths. This task can be performed if candidate routing paths can be predicted by inferring routing policies of ASs involved.

In this paper, we first infer and characterize import routing policies. In particular, we infer the route preference setting among routes learned from providers, customers and peers. From a large collection of routing tables, we find that in most cases, route preference conforms to AS relationships. That is, routes learned from customers are typically pre-

ferred over those from providers or peers, and routes learned from peers are typically preferred over those from providers. In addition, we observe that about 98% of route preference assignments are simply based on next hop ASs.

Second, we present an algorithm for inferring export policies and characterize the export policies. We infer how an AS announces its routes to its (direct or indirect) providers. Our results show that a significant number of ASs announce their prefixes to a selected subset of providers. Furthermore, the selective announcement is prevalent and persistently present. We investigate the cause of the selective announcement. We find that prefix splitting and aggregation are not the main reasons. The majority of the cases are due to the traffic engineering practice for controlling incoming traffic. That is, an AS may announce its prefixes only to a subset of its direct providers, or to its direct providers with a community tag indicating that the prefixes should not be announced further. Although the selective announcement routing policies are not surprising, the impact of these routing policies might be significant. For example, many Tier-1 ASs reach their (direct or indirect) customers via their peers instead of customers. That is, selective announcement might lead to "curving" routes in which a peer route is used when there is a customer route from the AS connectivity graph. Furthermore, the selective announcement routing policies imply that there are much less available paths in the Internet than shown in the AS connectivity graph. We hope that our findings will caution network operators in choosing the selective announcement routing policy for traffic engineering. Finally, we study export policy to peers. We observe that most ASs tend to export all of their prefixes to their peers.

To the best of our knowledge, this is the first study on systematically understanding routing policies applied in the Internet. The rest of this paper is organized as follows. In Section 2, we describe the Internet architecture and routing policies. Section 3 presents our data source. In Section 4, we describe our methodology for inferring import policies and the characteristics of the import policies. Then, in Section 5, we present algorithms for inferring export policies and characterize the export policies. We conclude the paper with a summary in Section 6.

2. BACKGROUND

In this section, we first present an overview of the Internet architecture, and then describe Internet routing policies.

2.1 Internet Architecture

Routing within ASs is achieved by the Interior Gateway Protocols (IGP). Routing information between ASs is determined by BGP, which includes interior BGP (iBGP) and exterior BGP (eBGP). eBGP exchanges reachability information between ASs, while iBGP exchanges exterior reachability information within an AS.

ASs negotiate agreements to achieve two forms of AS relationships between various networks, namely provider-to-customer and peer-to-peer. A pair of ASs is said to have a provider-to-customer relationship if one offers Internet connectivity to the other ; a pair of ASs providing connectivity between their respective customers is said to have a peer-to-peer relationship.

We represent AS relationships by an annotated AS graph. An annotated AS graph is a graph $G = (V, E)$, where the node set V consists of ASs and the edge set E are classified into provider-to-customer, and peer-to-peer edges. Fig. 1 shows an example of an annotated AS graph. In this example, AS2 is the provider of AS4, and AS4 is a customer of AS2. AS3 peers with AS4.

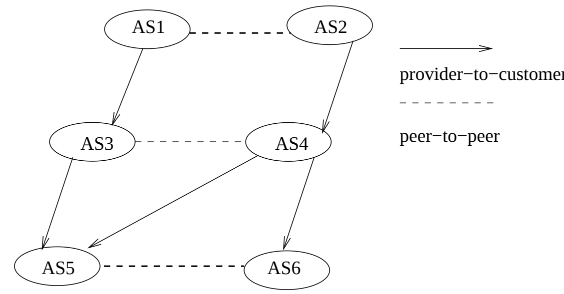


Figure 1: An example of an annotated AS graph. AS2 is the provider of AS4, and AS4 is a customer of AS2. AS3 peers with AS4.

2.2 Routing Policies

Routing policies are a set of rules that are configured by network operators such that one AS can determine how to select the best routes, and whether to propagate its best routes to neighboring ASs. Routing policies include import policies and export policies.

2.2.1 Import Routing Policies

From each neighbor, a router receives a set of route announcements. In order to distinguish routes from different neighbors, we define a route received from a customer as *customer route*, and the AS path the route traversed as *customer path*; a route received from a provider as *provider route*, and the AS path the route traversed as *provider path*; a route received from a peer as *peer route*, and the AS path the route traversed as *peer path*. We will use those terms to help us inferring routing policies through this paper. Each route announcement contains a set of attributes, AS path, multi-exit discriminator, next hop, and community [11]. Those attributes are used for configuring routing policies.

After receiving a route announcement, a BGP router discards the route if its own AS number is present in the AS path to avoid a loop in AS path. The router then applies import policies to this route which include denying, or permitting a route, and assigning a local preference to indicate how favorable the route is. The preference is a value used to rank routes received from different neighboring ASs.

Local preference is typically assigned based on prefix or AS. For example, the following configuration:

```
router bgp 65503
neighbor 192.1.250.23 remote-as 65504
neighbor 192.1.250.23 route-map isp1 in
access-list 1 permit 0.0.0.0 255.255.255.255
route-map isp1 permit
  match ip address 1
  set local-preference 90
```

shows setting the local preference based on the next hop AS (AS65504). All routes received from AS65504 are assigned

with a local preference value of 90. The *permit* in the *access-list* and *route-map* statements means matched routes will be permitted to enter the BGP table or be propagated to the neighbor. Using *prefix-list*, local preference can be set according to the destination IP address, for example, the following configuration:

```
ip prefix-list 1 permit 10.1.1.1/24
route-map isp1 permit
  match ip address prefix-list 1
  set local-preference 80
```

sets local preference value to be 80 for the prefix 10.1.1.1/24.

BGP incorporates a sequential decision process which calculates the degree of preference for various routes to a given prefix. A BGP router selects the best route for each prefix from a set of routes according to the following criteria:

1. Routes with the highest local preference.
2. Routes with the shortest AS path.
3. Routes with the lowest origin type, where a route originally learned from IGP is preferable to a route learned from the BGP.
4. Routes with the smallest MED for routes with the same next hop AS.
5. Prefer routes learned from eBGP over those from iBGP.
6. Routes with the smallest IGP metric to the egress border router.
7. Routes with the smallest router ID.

Note that the first route selection picks the route with the highest local preference value. In this paper, we focus on inferring local preference setting in import policies.

2.2.2 Export Routing Policies

After selecting the best route, a BGP router will propagate only the best route to its neighboring ASs. Export policies allow the router to determine whether to advertise the best route to a neighbor. Export policies include permitting or denying a route, assigning MED to control the inbound traffic, tagging a BGP community to indicate what preference a neighboring AS should assign to it, and prepending AS paths or redistributing its prefixes to affect the inbound traffic. The configuration of export policies is similar to that of import policies. It can be based on the AS path or prefix.

The following rules are well known BGP export policies [15]:

- Exporting to provider: A customer can export to its providers its routes and the routes learned from its own customers, but cannot export routes learned from other providers or peers.
- Exporting to customer: A provider can export to its customers its routes, the routes learned from the other customers, its providers, and its peers.
- Exporting to peer: A peer can export to another peer its routes, the routes learned from its customers, but cannot export the routes learned from its providers and other peers.

We note that route selection of an AS depends on the routes coming from its neighboring ASs after export policies are applied , as well as on the import policies of the AS. In the following, we assume that an AS accepts all route announcements from its neighbors.

3. SOURCES OF DATA

We conduct our analysis using data sources from Oregon RouteView server [16] and Looking Glass servers [17]. Oregon RouteView provides a view of the global routing system from the perspectives of several different backbones and locations around the Internet. On Nov. 2002, it peered with 56 ASs which announce their default-free routes to it. Those ASs include nearly all Tier-1 ASs in the Internet, such as AS1239 (Sprint) and AS7018 (AT&T) [1].

Besides BGP tables from Oregon RouteView, we use BGP tables from 15 ASs' Looking Glass servers. These tables include 3 Tier-1 ASs, AS1, AS3549, and AS7018. Through those Looking Glass servers, we can retrieve fine-grained routing information, such as Local Preference, and BGP community. Combining BGP tables from RouteView and ASs' Looking Glass servers, we have 68 routing tables from different ASs. Table 1 shows the name, degree, and location of each AS. Among them, 42 ASs are managed by ISPs in North America. 33 ASs are managed by ISPs in Europe, 3 ASs in Australia, 2 ASs in Asia. The sizes of those ASs span a large range. For example, AT&T (AS7018) has a degree of 1330, and Lirex Net (AS8262) has a degree of 14. We believe that those data sources are sufficient for our study. All BGP tables are downloaded on Nov.11, 2002, and Nov.18, 2002.

Our study relies on AS relationships. There are several novel algorithms which can be used to infer AS relationships from a collection of BGP routing tables [12][8][13]. Here, we choose the one described in [12]. In Section 4.3, we show that the potential error introduced by inferred AS relationships is small.

4. INFERRING IMPORT POLICIES

One of the most important aspects of import policies is to set local preference. Local preference can be used to influence the selection of the best route among a set of routes, and control outgoing traffic. First, we infer route preference among routes from providers, customers and peers. Then we analyze the consistency of local preference setting with next hop ASs.

4.1 Route Preference Among Provider, Customer, and Peer Routes

BGP default routing policy which selects the route with the shortest AS path length is overridden by routing policies that set local preference. Network operators usually assign different local preference values to customer, provider, and peer routes.

We use BGP routing tables shown in Table 1 to discover route preference. After knowing AS relationships between an AS and its neighbors and deriving local preference from those tables, we associate each neighbor, or customer, provider and peer, with one or more local preference values. Hence, we compare local preference values among different routes. We define:

- *Typical Local Preference*: customer routes have higher local preference than peer routes and provider routes, and peer routes have higher local preference than provider routes.

[1]We classified each AS to its tier using the method described in [8].

Table 1: Characteristics of Oregon RouteView and 15 ASs' Looking Glass servers. 42 ASs are managed by ISPs in North America (NA). 33 ASs are managed by ISPs in Europe (Eu), 3 in Australia (Au), 2 in Asia (As).

AS number	AS name	Degree	Location
AS6664	Oregon RouteView	peering with 56 ASs	NA(40) As (2) Eu (12) Au (2)
AS7018	AT&T	1330	NA
AS577	Bell Backbone	89	NA
AS3549	Global Crossing	558	NA
AS6539	Group Telecom Data Core	157	NA
AS1	GTE Internetworking	599	NA
AS12859	Business Internet Trends BV	109	Eu
AS2578	Demos, Moscow, Russia	34	Eu
AS513	European Organization for Nuclear Research	39	Eu
AS5511	France Telecom	168	Eu
AS12359	INTELIDEAS	31	Eu
AS6667	Jippii Group	26	Eu
AS8262	Lirex Net	14	Eu
AS559	Swiss Academic and Research Network	33	Eu
AS6762	Telecom Italia international high speed	120	Eu
AS7474	Optus Communications Pty Ltd	114	Au

- *Atypical Local Preference*: the local preference of peer routes or provider routes is not lower than that of customer routes, or the local preference of provider routes is not lower than that of peer routes.

Table 2 shows the percentage of prefixes which have typical local preference for each AS. Our result implies that the percentage of atypical local preference for each AS is very small. Those 15 ASs include 3 Tier-1 ASs (AS1, AS3549, and AS7018), 2 Tier-2 ASs (AS5511, and AS7474), and other 10 ASs.

In order to get a more complete view of local preference setting, we resort to the Internet Routing Registry (IRR) to infer import policies. IRR maintains ASs' routing information in several public databases. The motivation of IRR is to coordinate global routing policies, but the IRR database may not be complete and some part of it can be out-of-date. We downloaded public IRR database files mirrored at [10] on Nov. 25th, 2002. First, we check each AS's last update time and discard those ASs which are not updated during 2002.

The IRR database expresses routing information at various levels (e.g., individual prefix or AS, etc.). The following example shows how the import policy is expressed in Routing Policy Specification Language (RPSL).

```
aut-num: AS1
import: from AS2 action pref = 1; accept ANY
```

Policy actions in RPSL can assign a preference to a route. This example states that all routes are accepted from AS2

Table 2: Typical local preference assignment for 15 ASs. It shows the prevalence of typical local preference.

AS number	% of typical local preference	AS number	% of typical local preference
577	94.3	2578	99.9982
5511	96.5	513	100
3549	99.7	6762	100
6667	99.94	559	100
7474	99.955	12859	100
12359	99.98	8262	100
7018	99.99	6539	100
1	99.994		

with preference 1. In a real router configuration the preference can be done by setting a local preference [2].

However, some ASs shown in IRR do not appear in Oregon BGP table. Hence, we cannot infer their AS relationships. Therefore, we only consider those ASs which have more than 50 neighbors and most of their AS relationships can be inferred. Finally, we infer the typical local preference for 62 ASs from IRR, shown in Table 3. Those ASs include 5 Tier-1 ASs, and others are Tier-2 or Tier-3 ASs. Even though those ASs (15 ASs from BGP tables, 62 ASs from IRR) are a small fraction of ASs in the Internet, we believe that the chosen ASs are representative for studying import policies in the Internet. Therefore, we conclude that local preference value for a customer is typically higher than for a provider and peer, and that local preference for a peer is higher than that for a provider.

4.2 Consistency of Local Preference with Next Hop ASs

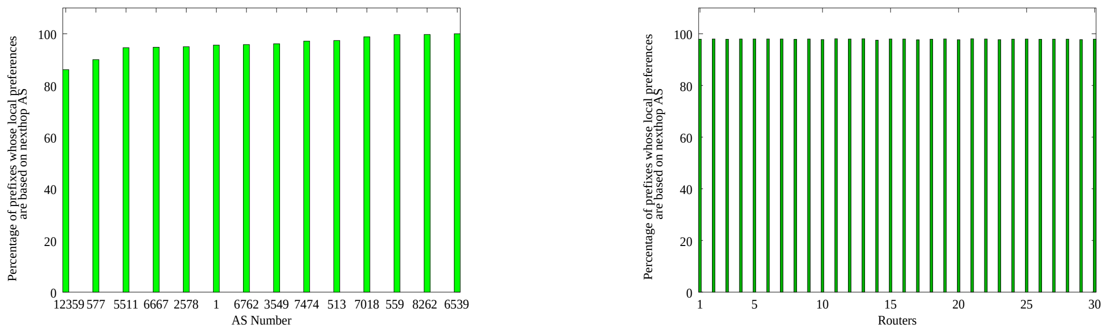
As mentioned above, operators may set local preference value on network prefix or next hop AS. It is easy for network operators to maintain local preference configuration based on next hop AS. This motivates us to study the consistency of local preference values with next hop ASs. We use 14 ASs in our dataset to study the consistency. Fig. 2(a) shows that most of the ASs assign a unique local preference value for each next hop AS.

All routing tables that we use are collected from only one or several routers at each AS. In order to understand if local preference values are consistent within an AS, we use AT&T routing tables (Jan. 4, 2002) combined from 30 backbone routers to study the consistency. Fig. 2(b) shows that most local preference values assigned in AT&T are based on its next hop ASs. ASs tend to assign local preference values based on next hop AS instead of on prefix.

4.3 Potential Error Introduced by Inferred AS Relationships

Since studying routing policies relies on AS relationships, a large number of ASs with incorrectly inferred AS relationships will affect our conclusion about import policies. We use BGP community to verify some inferred AS relationships. One of the most common usages of community values is to tag the routes received from specific neighbor ASs. In this case, an AS defines different community values

[2]Preference is opposite to local preference in that the smaller values are preferred over larger values.



(a) 14 ASs

(b) AT&T

Figure 2: Consistency of local preference with next hop ASs.

for its customers, peers, and transit providers. When border routers of the AS receive a route from its neighbors, they tag the route with a community indicating the relationship with those neighboring ASs. Details about this method are described in the Appendix.

Table 4 shows that the AS relationships between 9 ASs and their neighboring ASs are verified. As shown in the table, for those 9 ASs, most of their AS relationships are correctly inferred. Therefore, the potential error introduced by inferred AS relationship is so small that it will not affect our results.

5. INFERRING EXPORT POLICIES

How to announce prefixes to a customer, peer, or provider is an important component of export policies. For a provider, it has to announce all of its prefixes, or default routes to its customers depending on their agreements. However, a customer may advertise its prefixes to either all of its providers, or a subset of providers. In the latter case, customers can control their inbound traffic on a heavy traffic link by switching announcements of some prefixes away from the link. Peers also have control over their prefix announcements to other peers. Here, we focus on two problems in export policies:

1. Export to provider: strategies a customer uses to export prefixes to its provider.

2. Export to peer: strategies a peer uses to export prefixes to its peers.

5.1 Export to Provider

In this section, we first describe an algorithm to infer export policies which customers use to advertise their prefixes to direct or indirect providers. Then we characterize those export policies.

5.1.1 Algorithm for Inferring Export Policies to Providers

The direct way to infer a customer's export policies is to use the BGP table from its provider. After searching prefixes originated by the customer in the table, if those prefixes have customer routes, which we defined above, we know that the customer exports those prefixes to the provider. On the contrary, if those prefixes do not exist or do not have customer routes, it implies that the customer does not export them to the provider directly.

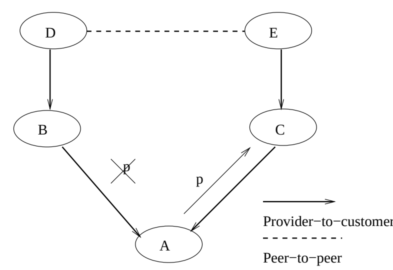


Figure 3: The selective announcement routing policies employed by its customers can be observed at provider D. Customer A announces prefix p to provider C but not to B. In the BGP table of provider D, prefix p is received from its peer E.

Therefore, we infer the export policies for customers from the viewpoint of a provider. As we described above, a customer can export prefixes to all of its providers or a subset of providers. For a given provider, if it receives a prefix originated by a customer via a peer path instead of a customer path, we call this prefix as a *selective announced prefix* (*SA* prefix) with respect to the provider. Here, we use our analysis result that customer's selective announcement policies give rise to *SA* prefixes. We will discuss the causes of SA prefixes in Section *5.1.5*. As a result, the selective announcement used by customers can be observed from the viewpoint of a provider. For example, in Fig. 3, customer A exports prefix p to a selected subset of providers, provider C. In D's BGP routing table, prefix p is received from D's peer, E. No customer route to p is received from customer B.

Note that from the point of view of a provider, the best routes to customers' prefixes, instead of all routes, are sufficient to infer the selective announcement policies. From Section 4.1, we know that a customer route is typically pre-

Table 3: Typical local preference assignment for 62 ASs (ASs are sorted according to their AS degree in non-decreasing order) which are selected from IRR. It shows the prevalence of typical local preference.

AS number	% of typical local preference	AS number	% of typical local preference
12635	100	5611	98
15498	100	8608	100
4004	99.86	12306	93.5
6863	99.90	5400	100
12322	99.92	3215	100
12779	100	3300	94.7
12626	99.94	1740	100
2518	100	8341	100
8650	91.66	293	83.2
20646	100	6705	80
5539	89	8434	100
5615	100	12390	98
12573	96	5607	95
1140	98	5427	99
6873	100	4000	100
12781	98	1901	97
8365	100	15290	100
852	100	3320	83
8527	100	13127	93
5551	100	9191	100
3313	97.8	5466	94
12731	97.8	5597	98
15435	98.9	6453	100
3216	100	12868	99
2118	88.6	5594	96
1103	88.9	13129	99.23
21392	100	6830	100
9013	96.9	1299	99.1
5571	98	3292	86
3344	90.4	4513	100
5503	98	3561	99.46

ferred over other routes. In a provider's BGP table, if a customer route to a prefix exists, the route is the best route as well. Otherwise, if a customer route does not exist, the best routes are peer routes or provider routes.

The first step of the algorithm for inferring export policies to provider is to find if an AS is a customer of a given provider. This can be solved by using *Depth First Search* (DFS) algorithm in a directed graph to find a customer path from the provider to the AS. If there is a customer path, the AS is a customer of the provider. Not all paths found by DFS can be customer paths, however, those paths should obey export rules described in Section 2.2. That is, from the direction of provider down to customer, each pair of ASs in the path should have provider-to-customer relationship. In an annotated AS graph $G = (V, E)$, we use modified *DFS* which satisfies path relationship constrains to find a customer path between a pair of ASs.

The next step is to investigate if the best routes to the customer's prefixes are peer or provider routes. If the best routes are peer or provider routes, those prefixes are not exported from the customer to the provider, or some intermediate customers who receive those prefixes do not export them.

Fig. 4 shows the algorithm in detail. Given an AS, we use this algorithm repeatedly for all of the AS's customers to infer those customers' export policies.

Table 4: The AS relationships between 9 ASs listed below and their neighbors are verified as shown in the Appendix.

AS number	# of neighbors	Percentage of AS relationships between AS and its neighbors verified
AS1	599	95.65%
AS577	89	98.9%
AS3549	558	96.28%
AS5511	168	99.4%
AS6539	157	96.45%
AS6667	26	97.46%
AS7018	1330	99.55%
AS12359	31	94.1%
AS12859	109	98.2%

Table 5: Percentage of *SA* prefixes for 16 ASs

AS number	% of *SA* prefixes	AS number	% of *SA* prefixes
AS1	32	AS7018	22
AS3549	23	AS701	27.8
AS6453	48.6	AS6461	4
AS1239	29.4	AS3561	5.2
AS2914	14	AS209	38
AS5511	18	AS577	17
AS6538	11	AS6667	13
AS12359	0	AS12859	0

5.1.2 Prevalence of SA Prefixes

Here, we present experimental results of inferring export prefixes to provider using the algorithm. We first use dataset described in Section 3 to construct the annotated AS graph which is used to find all direct or indirect customers of a given provider. We then use the routes from Oregon or ASs' BGP tables to derive the best routes to customers' prefixes. *SA* prefixes for 10 Tier-1 ASs can be inferred by using Oregon RouteView and 3 Tier-1 ASs' BGP tables.

Table 5 shows the percentage of customers' prefixes that are *SA* prefixes for 16 ASs. We find that Tier-1 ASs, such as AS1, AS3549, and AS7018, have a significant number of *SA* prefixes. Those Tier-1 ASs reach their (direct or indirect) customers via their peers instead of customers. For example, in Fig. 5, AS6280 is a customer of AS1. However, AS1 does not receive a prefix p originated by AS6280 from AS852. It receives p from its peer, AS3549. Note that *SA* prefixes for a provider may be due to the selective announcement policies of originating ASs or intermediate ASs. For example, in Fig. 5, the *SA* prefix for AS1 may be due to the selective announcement policies employed by AS6280 or AS852.

Next, we examine *SA* prefixes from the viewpoint of a set of customers. We consider those customers which all have 3 direct or indirect providers: AS1, AS3549, and AS7018. From those customers, we select 8 ASs which originate a significant number of prefixes as shown in Table 6. Table 6 shows that those 3 providers cannot access some of customers' prefixes directly via their customer paths.

Applying the selective announcement policies, a customer can balance its inbound traffic but its inbound and outbound traffic might be asymmetric. From the point of view of a provider, it may find that traffic between its customers has to forward to the rest of Internet via its peer links.

Algorithm for inferring export policy
Input:
  Annotated AS graph G
  AS o which originates prefixes P
  routing table from the viewpoint of AS u
Output:
  Whether P contains SA prefixes from the point of view of AS u
Phase1: Initiation
1. Selected AS set $S = \{u\}$

Phase2: Determine if AS o is a customer of AS u
1. while there is a selected AS
2.   for each AS v that is a customer of the selected AS
3.   if v is AS o
4. o is a customer of AS u
5.   go to Phase3
6.   else add v into S
7. AS o is not a customer of u
8. return

Phase3: Determine if P contains SA prefixes
1. for each next hop AS w of the best route to p_i, $p_i \in P$
2.   if u is not a provider of w
3. p_i is a SA prefix
4.   else p_i is not a SA prefix
5. if any prefix in P is a SA prefix
6.   then P contains SA prefixes
7. else P does not contain SA prefixes

Figure 4: Algorithm for inferring exporting policy.

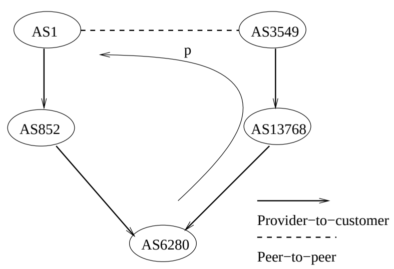


Figure 5: For AS1, a prefix p originated by its customer AS6280 is received from AS1's peer, AS3549. Prefix p is a SA prefix.

This strategy may affect traffic engineering practice of the provider.

5.1.3 Verification of SA Prefixes

Because SA prefixes depend on AS relationships, the goal of our verification is to confirm the validity of all AS relationships which are used to infer SA prefixes. When we infer customers' export policies, or selective announcement policy, using the algorithm described before, we first investigate if the AS which originates the SA prefix is a customer of a given provider, and then the AS relationship between the provider and the next hop AS of the best route to the prefix is examined. We verify SA prefixes by following steps:

Step 1: Verify AS relationships between a given provider and its neighboring ASs. As described in Section 4.3 , the AS relationships between 9 ASs in Table 4 and their neighboring ASs are verified by using BGP community as shown in the Appendix. Such small error in inferring AS relationships can be neglected. In our verification, we focus on 3 Tier-1 ASs, AS1, AS3549, and AS7018, since a large number of SA prefixes are observed from those 3 ASs. The result from Table 4 also implies that the peer relationship between those 3 providers and their next hop ASs in the best route is verified.

Table 6: Percentage of prefixes from each customer inferred as SA prefixes for AS1, AS3549 and AS7018

Customer	# of prefixes	# of SA prefixes for AS1, AS3549 and AS7018
AS376	344	205 (60%)
AS6280	33	32 (97%)
AS10910	51	17 (33%)
AS11647	28	24 (86%)
AS14743	22	15 (68%)
AS15087	65	11 (17%)
AS19024	30	13 (43%)
AS19916	25	24 (96%)

Step 2: Verify the customer relationship between an AS which originates a SA prefix and a given provider. Because an AS can be a direct or indirect customer of a provider, we first verify the direct customers. We can verify direct customers since we already verify AS relationships between those 3 providers and their neighboring ASs. For indirect customer, we need to verify all AS relationships between each pair of ASs in the customer path which is used in the algorithm described in Fig. 4 to infer a SA prefix. Our method is to investigate the existence of the customer path in the Internet. Even though the customer path is derived from our algorithm, it is possible that some other prefixes are really announced through it. We call a customer path *active* if other prefixes traverse the same path. By searching all paths in BGP routing tables, we determine whether a customer path for a SA prefix is active. Given a customer path, for example, `AS1 AS12 AS14 AS15`, if the customer relationship between `AS1` and `AS12` is verified at the first step, and the path is active, the relationships between `AS12` and `AS14, AS14` and `AS15`, can be verified as provider-to-customer. Otherwise, if `AS12` is a peer, or a provider of `AS14, AS12` cannot announce path `AS12 AS14` to its provider `AS1` according to the export rule described in Section 2.2.

We verify all SA prefixes for AS1, AS3549, and AS7108 according to the steps described above. Table 7 shows that most of SA prefixes for those 3 ASs are verified.

Besides verifying AS relationships, for each AS, if its import policies are already inferred, we only consider those prefixes that have typical local preference. That is, prefixes from peers have lower local preference than those from customers, or prefixes from providers have lower local preference than those from peers. We note that the percentage of anomaly local preference is small in Table 2, and will not affect our results.

5.1.4 Persistence of SA Prefixes

Having identified the prevalence of SA prefixes, we now turn our attention to SA prefix persistence. Network operators may change prefix exporting pattern at different time. To find out how persistent SA prefixes are over a period of

Table 7: Large number of *SA* prefixes for AS1, AS3549, and AS7018 are verified.

Provider	# of *SA* prefixes	% of *SA* prefixes verified
AS1	9120	97.6%
AS3549	3431	95%
AS7018	4374	97%

measurement time, BGP tables for March 2002 from Oregon RouteView are used. We are also interested in how *SA* prefixes change within one day, so we use data from Oregon RouteView on March 15, 2002. Here, we only present the result of AS1 because it has a large number of *SA* prefixes.

Fig. 6(a) shows the number of *SA* prefixes during March, 2002. Fig. 6(b) shows the number of *SA* prefixes on that day (March 15, 2002). From Fig. 6, we find that *SA* prefixes are consistently present in AS1.

As we mentioned above, network operators might change their routing policies to control incoming traffic. To find out how export policies affect the existence of *SA* prefixes, we define the times each prefix appears during the measuring time as *uptime*. *SA prefix uptime* is defined as the times a *SA* prefix appears. The maximum uptime of a prefix during our measurement is 31 days or 24 hours depending on which view of the data we are examining. We study how many prefixes shift from *SA* prefix to non-*SA* prefix during the whole period. For example, some prefixes have 31 *uptimes* but less than 31 *SA prefix uptimes* during the whole month. Those prefixes shift from *SA* prefixes to non-*SA* prefixes. From Fig. 7, we observe that about one sixth of *SA* prefixes are not stable during one month, but most of them are stable during one day period. Changes in routing policies can affect existence of *SA* prefixes.

5.1.5 Causes of SA Prefixes

A provider may have two different connectivities to its customers: a direct customer path, and a "curving" peer path. This leads us to analyze two ways which customers use to connect providers: multihomed and single-homed. Fig. 8(a) shows that customer v is multihomed to two providers. Path $u_0u_2u_1v$ is the best path, and path u_0u_3v is a customer path from u_0 to v. The best path and the customer path are disjoint paths. Fig. 8(b) shows that customer v is single-homed. Path $u_0u_2u_1v$ is the best path, and path $u_0u_3u_1v$ is a customer path from u_0 to v. They share some paths.

For AS1, AS3549, and AS7018, we examine the connectivities between those 3 ASs and their customers. From Table 8, we find that among those customers whose announced prefixes are *SA* prefixes, about 75% of them are multihomed, and others are single-homed. In the multihomed case, origin ASs or intermediate ASs may apply selective announcement policies to prefixes. In the single-homed case, only intermediate ASs which are multihomed can apply selective announcement policies to their prefixes or their customers' prefixes. Intuitively, it is more likely for multi-homed ASs to generate *SA* prefixes than for single-homed ASs. This is confirmed in Table 8.

From the point of view of a provider, we define a prefix as a *SA* prefix if the provider receives the prefix originated by its customer from a peer instead of a customer. However,

Table 8: Distribution of multihomed and single-homed ASs whose prefixes are *SA* prefixes for AS1, AS3549, and AS7018

Provider	# of ASs whose prefixes are *SA* prefixes	
	multihomed	single-homed
AS1	611 (75%)	201 (25%)
AS3549	1664 (75%)	549 (25%)
AS7018	2063 (77%)	608 (23%)

Table 9: The number of prefixes contributes to prefix splitting and prefix aggregating.

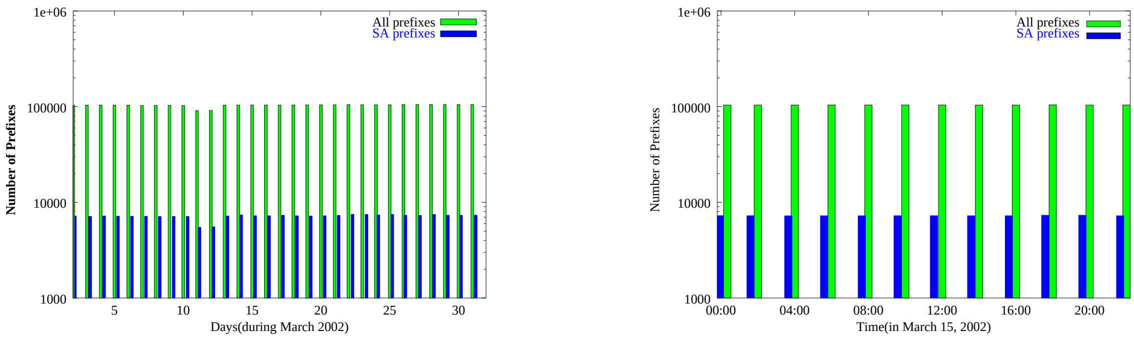
Provider	# of *SA* prefixes	# of prefix splitting	# of prefix aggregating
AS1	9120	127	218
AS3549	3431	63	104
AS7018	4374	71	179

other cases can cause the provider not to receive the prefix from its customer path. Here, we study 3 cases which may produce *SA* prefixes.

Case 1: Prefix splitting. Network operators can split one prefix into some more specific prefixes [7]. For example, a prefix "12.0.0.0/19" can be split into a more specific prefix "12.10.1.0/24". Then they announce the specific prefix to a provider through a peer link, and announce the original prefix to the provider through a customer path. They take advantage of this approach to balance load and tolerate link failures. If the customer path through which the more specific prefix is announced is broken, the prefix can be accessed through the other path. This configuration can produce *SA* prefixes.

For all prefixes in AS1, AS3549, and AS7018, we analyze the number of prefixes which are split prefixes. If we find one prefix that can be aggregated by another prefix, and both prefixes belong to the same source AS but have different routes (i.e. one is a customer route, the other is a peer route), those two prefixes are split. Comparing with the number of *SA* prefixes, we find that the number of prefixes splitting, shown in Table 9, is so small that prefix splitting is not the main cause of *SA* prefixes.

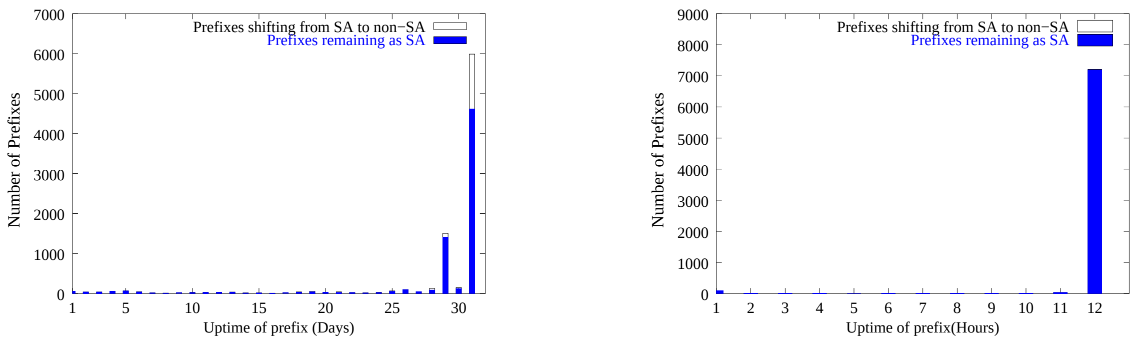
Case 2: Prefix aggregating. A provider can allocate a part of its IP address space to its customers. When the provider receives prefix announcements from customers, those allocated prefixes can be aggregated by the provider so that they will not be announced to other ASs. For example, a customer is allocated with a prefix "12.10.1.0/20" from its provider's IP space "12.10.0.0/19". Even though the provider receives an announcement for "12.10.1.0/20", it announces "12.10.0.0/19" only. We call this prefix aggregating. For all prefixes of AS1, AS3549, and AS7018, we analyze the number of prefixes which are aggregated. For simplicity, we estimate this case by finding how many *SA* prefixes can be aggregated by other prefixes without considering if providers can aggregate those prefixes or not. Our estimation can be regarded as upper bound of this case. Table 9 shows the extent to which prefix aggregating exists in



(a) *SA* prefixes for AS1 during March 1-30, 2002

(b) *SA* prefixes for AS1 on March 15, 2002

Figure 6: Persistence of *SA* prefixes for AS1.



(a) March, 2002

(b) March 15, 2002

Figure 7: The number of prefixes which shift from *SA* prefix to non-*SA* prefix or remain as *SA* prefix for AS1 during the measuring time.

those 3 ASs. We find that prefix aggregating cannot be the main cause of *SA* prefixes.

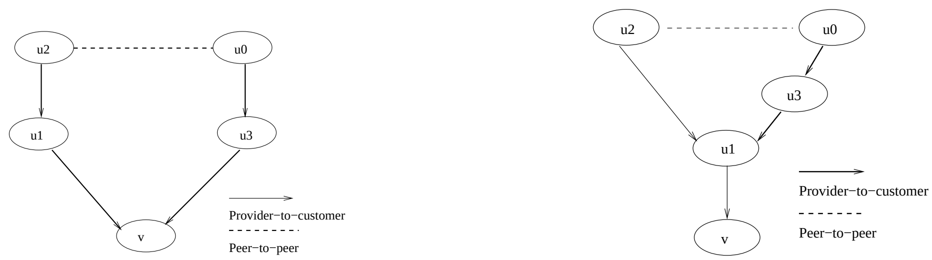
Case 3: Selective announcing. Here, we focus on how origin ASs export prefixes to their direct providers. We only show the result for AS1 since we observe about 32% of *SA* prefixes in its routing table.

Note that some origin ASs are single-homed ASs. They must announce their prefixes to direct providers. For example, in Fig. 8(b), AS v announces its prefixes to its direct provider u_1. AS u_1 is the last common AS of the best path $u_0u_2u_1v$ and the customer path $u_0u_3u_1v$ except the source and destination ASs. Therefore, it may be due to export policies of AS u_1 or ASs before it, such as AS u_3. In the single-homed case, we investigate how the last common AS exports prefixes of itself and prefixes learned from its customers to its direct providers.

For each *SA* prefix in AS1, we search all paths in BGP table to investigate how the customer (the originating AS or the last common AS) connects with the direct provider in the path. If the provider is left to the customer, the customer exports the prefix associated with the path to the provider. If between the provider and the customer, there is a upstream provider of the provider, then the customer does not export this prefix to the provider. For example, in Fig. 8(a), if we find the path u_3v, v exports its prefixes to u_3. If we find the path $u_3u_0u_2u_1v$, v does not export its prefixes to u_3.

This method depends on the number of peers in Oregon RouteView. Thus some of *SA* prefixes cannot be identified. In AS1, about 90% of *SA* prefixes can be identified by using this method. Among those identified prefixes, we find that about 21% of customers announce their prefixes to the direct provider, and about 79% of customers do not export prefixes to the provider. That is, in order to control their incoming traffic, some customers announce their prefixes or their customers' prefixes to only a subset of direct providers. Multihomed ASs can use BGP conditional advertisement, in which some prefixes are advertised to one of the providers only if information from the other provider is missing [18]. This feature can be used to provide administrative control over traffic flow.

BGP community tagged with routes can explain the case



(a) Multihomed AS

(b) Single-homed AS

Figure 8: Examples of multihomed and single-homed ASs.

that customers can export *SA* prefixes to direct providers. One of the important usages of BGP community attributes is for traffic engineering. The well-known NO_EXPORT and NO_ADVERTISE communities are employed to control prefix announcement [19]. Besides these values, there are some communities that are often used to indicate that receivers do not announce routes to a specified AS. A large number of ASs support this kind of community, and a lot of communities related to this utilization can be found in BGP tables [20]. Therefore, it is possible that direct providers or their upstream providers do not announce the customers' routes because of those communities.

From these 3 cases, we find that the major cause for *SA* prefixes is due to selective announcing. This implies that selective announcement policies employed by customers can affect the paths through which the top providers receive their prefixes. Providers may receive its customers' prefixes from their peers. However, if those peers do not announce these prefixes, providers cannot access customers. Afek et al. [21] studied *policy atoms* which are groups of prefixes with a common AS path at any Internet backbone router. [21] shows that most policy atoms are created by origin ASs' routing policies. Our work can answer the questions as to what kind of routing policies create policy atoms in [21].

Policies for exporting to providers are the major cause for *SA* prefixes. However, some special cases can also give rise to *SA* prefixes. For instance, an AS has several networks at different areas, but this AS does not have backbone to connect its networks. The AS will connect to the nearest providers for each network. For example, AOL (AS1668) belongs to this case. We will study other cases which may affect our result in the future work.

5.2 Export to Peer

It is more complicated to infer what strategy peers use to export their prefixes to other peers, since a pair of peers can define flexible export policies to balance their inbound traffic. Here we focus on how a peer exports its prefixes to other peers.

From Oregon's BGP table in Nov. 18, 2002, we investigate how peers of AS1, AS3549, and AS7018 export their own prefixes to those 3 ASs. Table 10 shows the final results for those 3 ASs. For AS1, there are only 6 peers which do not export all of their prefixes directly to it. Among them, 4 peers export most of their prefixes directly to AS1. This may be due to load balancing among peer links. For AS3549, all of its peers export their prefixes to AS3549. For AS7018, only 4 peers do not export their prefixes to it. From Table 10, we observe that most peers export their own prefixes directly to those 3 ASs. That is, peers tend to announce their prefixes to other peers directly.

Table 10: Percentage of peers which announce their prefixes to AS1, AS3549, and AS7018, respectively

AS number	# of peers	% of peers announcing their prefixes
AS1	43	86%
AS3549	41	100%
AS7018	35	89%

6. CONCLUSION

In this paper, we demonstrate how to infer the routing policies and characterize the routing policies. We first infer the import policy. We find that for most ASs routing preference conforms to AS relationships. Routes learned from customers are typically preferred over those from providers and peers, and routes received from peers are typically preferred over those from providers. Moreover, route preference assignment is based on next hop ASs. Second, we present an algorithm for inferring export policies, and characterize export policies. Customers can export their prefixes to a selected subset of providers. For 3 Tier-1 ASs, we find a large number of prefixes are exported to a selected subset of providers. We find that prefix splitting and aggregation of prefix are not the main reason for the selective announcement. From our study, customers announce their prefixes to a subset of providers because of load balancing. Furthermore, most peers tend to export their prefixes to other peers directly.

Acknowledgments

The authors would like to thank Jennifer Rexford for many helpful discussions and comments on the paper, and for providing the AT&T routing data. We would also like to thank anonymous reviewers for many valuable comments. This work was supported in part by NSF grant ANI-0208116, ANI-0085848, and the Alfred P. Sloan fellowship.

7. REFERENCES

[1] Rekhter, Y., and T., Li. *Border Gateway Protocol 4 (BGP-4)*, RFC 1771, T.J. Watson Research Center, IBM Corp., Cisco Systems, March 1995.

[2] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker and Walter Willinger. *Network Topology Generators: Degree based vs. Structural*, ACM SIGCOMM 2002.

[3] Ramesh Govindan, and Hongsuda Tangmunarunkit. *Heuristics for Internet Map Discovery*, IEEE INFOCOM 2000.

[4] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. *On Power-Law Relationships of the Internet Topology*. In ACM SIGCOMM, Cambridge, MA, September 1999.

[5] Paul Barford, Azer Bestavros, John Byers, and Mark Crovella. *On the Marginal Utility of Network Topology Measurements*. In Proc. of the SIGCOMM Internet Measurement Workshop (IMW'01), November 2001.

[6] Ratul Mahajan, David Wetherall and Tom Anderson *Understanding BGP Misconfiguration*, ACM SIGCOMM, Aug 2002.

[7] Philip Smith. *BGP for Internet Service Providers*, NANOG 25, Toronto, Canada.

[8] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz. *Characterizing the Internet hierarchy from multiple vantage points*, in Proc. IEEE INFOCOM, June 2002.

[9] Meirt Network. *Internet routing registry*, http://www.irr.net.

[10] Merit Network. *Internet routing registry mirror site*, ftp://ftp.radb.net/routing.arbiter/radb/dbase/.

[11] B.Halabi. *Internet Routing Architectures (2nd Edition)*, Cisco Press, August 2000.

[12] Lixin Gao. *On inferring autonomous system relationships in the Internet*, ACM IEEE Transation on Networing, Dec, 2001.

[13] G. Di Battista, M. Patrignani, and M. Pizzonia, *Computing the Types of the Relationships between Autonomous Systems*, Technical Report RT-DIA-73-2002, Dipartimento di Informatica e Automazione, University Roma Tre, Rome, 2002.

[14] E. Chen, T. Bates. *An Application of the BGP Community Attribute in Multi-home Routing*. MCI, cisco Systems, August 1996.

[15] G. Huston. *Interconnection, peering, and settlements*. In Proc. INET, June 1999.

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

[17] http://www.traceroute.org

[18] *Configuring and Verifying the BGP Conditional Advertisement Feature*. http://www.cisco.com/warp/public/459/cond_adv.html.

[19] P. Traina, R. Chandrasekeran, and T. Li. *BGP communities Attribute*. Internet Engineering Task Force, RFC1997, August 1996.

[20] Bruno Quoitin, and Olivier Bonaventure. *A survey of the utilization of the BGP community attribute*, draft-quoitin-bgp-comm-survey-00.txt. February 2002.

[21] Yehuda Afek, Omer Ben-Shalom and Anat Bremler-Bar. On the Structure and Application of BGP Policy Atoms. IMW 2002.

APPENDIX

[20] shows that BGP community is widely used, and describes the most common usages of the BGP community attributes in the global Internet. Besides the well-known community values, one of the most common utilizations of community values is to tag the routes received from specific neighbor ASs. In this case, an AS defines community values for its customers, peers, and transit providers. When border routers of the AS receive a route from its neighbors, they tag the route with a community indicating the relationship with those neighboring ASs. Other routers within the AS can make their routing decisions on the community. For example, in RIPE whois database, AS12859 defines the community values, shown in Table 11, to tag routes received from its customers, peers, and providers.

Table 11: An example of tagging communities published by AS12859

12859:1000	Route received from AMS-IX peer
12859:1010	Route received from OpenPeering (AMS-IX)
12859:1020	Route received from private peers at Telecity 2
12859:2000	Route received from transit link at SARA
12859:2010	Route received from transit link in Ede
12859:2020	Route received from transit link at Telecity 2
12859:4000	Route received from customer

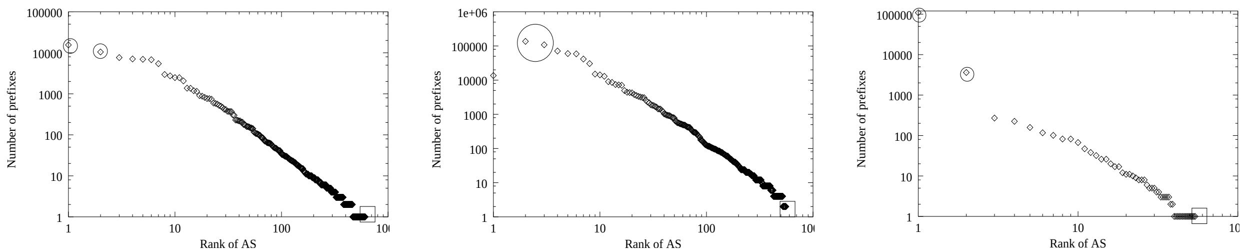
Note that one type of relationships can be indicated by a unique community value, such as AS12859 using "12859:4000" to indicate all of its customers. It can also be indicated by different community values. For example, AS12859 uses 3 different community values to indicate its peers, and other 3 values to indicate its providers. Therefore, different ranges of community values, which do not overlap, can indicate each type of relationships. We call two community values as the "same" if they belong to the same range of values. For example, "12859:1010" and "12859:1020" are the "same" values because they belong to the range of community values for peers which is between "12859:1000" and "12859:2000".

Given an AS, we can verify AS relationships between the AS and its neighboring ASs by following steps:

Step 1: querying community associated with next hop ASs. We first select some prefixes announced by each next-hop AS. Then we send "show ip bgp IP address" Cisco IOS command to query communities tagged with those prefixes. For example, in AS12859, the query result shows as follow:

```
> show ip bgp 80.96.180.0
BGP routing table entry for 80.96.180.0/24
Paths: (1 available, best #1)
 8220 12878 5606 15471
  193.148.15.101 from 213.136.31.5
  Origin IGP, metric 5, localpref 210, internal, best
   Community: 12859:1000
```

Step 2: inferring the semantics of community values. It is easy to infer the semantics of community values when ASs publish their rules, such as registering them in IRR database. AS12859, for example, publishes the semantics of community value in IRR, and AS6667 publishes the semantics of community values on its web page. Here we



(a) AS1 (b) AS3549 (c) AS8736

Figure 9: Number of prefixes announced by the next-hop ASs of AS1, AS3549, and AS8736.

focus on how to infer the semantics if ASs do not publish them. The main idea is that because all ASs with the same relationship will be tagged with the same type of community, we just identify each kind of relationships for only one pair of ASs. Then all ASs that are tagged with the same type of community will have the same relationship.

Based on the number of prefixes from next hop ASs, We can infer AS relationships for some special ASs. Suppose an AS has providers, it will receive full BGP tables, or partial list of prefixes from them. It will also receive prefixes announced by customers. The big gap between the number of prefixes received from a provider and a customer indicates that one AS that announces the most prefixes is a provider. The AS that announces the lest prefixes is a customer. If the AS does not have providers, then the top one which announces most prefixes must be a peer. If the AS has both providers and peers, we can distinguish one provider by virtue of the largest number of prefixes it announces. All providers should have the same type of community value, and peers have different one from that of providers. A peer is the one that announces a large number of prefixes, and with a different community from providers.

For example, Fig. 9 shows the number of prefixes AS1 (GTE), AS3549 (Global Crossing), and AS8736 (Grapes Network Services) received from their next hop ASs. We sort those next hop ASs in non-increasing order according to the number of prefixes they announce. Because AS1 and AS3549 do not have providers, we conclude that the first several ASs (indicating in a circle in Fig. 9), which are larger ASs, such as AS701 (UUNET), are peers. And we also can conclude that the last several next hop ASs, indicating in a square in Fig. 9, which announce very small number of prefixes, such as 1 or 2 prefixes, should be customers. In AS8736, one next-hop AS announces more than 100k prefixes to it, and then we conclude that that AS is a provider. Our conclusion for the semantics of AS3549's community is confirmed by this AS.

Step 3: mapping community to AS relationship. Based on the semantics of community values, we derive AS relationships between ASs and their next-hop ASs. For the example above, prefix "80.96.180.0/24" is tagged "12859:1000" community in AS12859. According to Table 11, the relationship between AS8220 and AS12859 is peer-to-peer relationship.